MEASURING, MONITORING, AND ASSESSING SOFTWARE PROCESS USING PAMPA 2.0 KNOWLEDGE-BASED SYSTEM

A Dissertation

by

JIN HWAN JUNG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2004

Major Subject: Computer Science

MEASURING, MONITORING, AND ASSESSING SOFTWARE PROCESS USING PAMPA 2.0 KNOWLEDGE-BASED SYSTEM

A Dissertation

by

JIN HWAN JUNG

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Dick B. Simmons (Chair of Committee)	
William M. Lively (Member)	Frank Shipman (Member)
Thomas L.Rodgers (Member)	Valerie E. Taylor (Head of Department)

May 2004

Major Subject: Computer Science

ABSTRACT

Measuring, Monitoring, and Assessing Software Process using PAMPA 2.0 Knowledge-Based System. (May 2004)

Jin Hwan Jung, B.A., Naval Academy, Korea;

B.A., Yonsei University, Seoul, Korea; M.S., Yonsei University, Seoul, Korea

Chair of Advisory Committee: Dr. Dick B. Simmons

My research is about monitoring the software development process to assess Capability maturity level. Capability Maturity Model (CMM) was developed to improve the software process based on subjective assessment by teams of experts. We propose an objective CMM assessment, which replaces expensive and time-consuming human effort by a knowledge-based system. Compared to Subjective CMM assessment, Objective CMM assessment can be less expensive, takes less time, and is easy to estimate the software development environment maturity. The accuracy of Objective CMM assessment can be the same as Subjective CMM assessment if enough activities are represented as objective activities. For example, if subjective activities total 80 % and objective activities total 20 %, then the accuracy of Objective CMM assessment is not reliable. It would be reliable if the objective activity is increased up to 80% from 20%.

This dissertation presents how to change from Subjective CMM assessment to Objective CMM assessment, and we will prove that Objective CMM Assessment is effective.

TABLE OF CONTENTS

Page

ABSTRACT……………………………………………………………..……iii

TABLE OF CONTENTS……………………………………………..……iv

LIST OF TABLES…………………………..……………….……….vii

LIST OF FIGURES………………………………………………………viii

1.INTRODUCTION.......……………………………………………..……1

A. Metrics………………………………………………………………...1
B. PAMPA 2.0……………………………………..………………...2
C. CASE Tools…………………………………..……..………4
D. PAMPA 2.0 Architecture Overview…...……………..………...15
E. Combine PAMPA 2.0 with CASE Tools……………………..…16

2.OBJECTIVE CMM ASSESSMENT………...…………………………19

A. CMM Assessment Procedure…………….………………........19
B. What is CMM?...……………………………………………….....20
C. Why Do We Need to Measure Software Project Attributes Related to CMM KPAs? ..…………………………….…..…....22
D. Modified CMM Assessment……………………….............23
1. Subjective assessment using metrics…………..….………..24
2. Simplified assessment method..……………………….…....25
3. CMM assessment using expert system..…………………......26
E. ISO 9000 and Software Life Cycle Standards……….............27
1. ISO 9000..………………………………………………......27
2. Rational Unified Process……………………………….........27
3. ISO/IEC 12207……………………………………………...28
4. IEEE/EIA 12207..……………………………………………...28
F. Create an Intelligent Agent to Assess CMM Objectively…........29
G. Measuring CMM Level………………………………………..30

Page

1 Questionnaire……………………………………………..............31
2. KPA goal……………………………………………………......32
3. KPA status…………………………………………..………33
4. CMM level……………………………………………………...33

3.KPA ASSESSMENT…...…………………………………………..34

A. PAMPA Assessment Environment…..………………………..34
B. Mapping the Questionnaire to KPA Goals and Practices……..35
C. Objective Assessment Limitation……………………………..39

4.IMPLEMENTATION…………………………………………..…45

A. Introduction…..……………………………………………..…45
B. PAMPA Object & Attribute Display…………………………45
1. Project list…..……………………………………………..….45
2. Project…..………………………………………………..…..47
3. Life Cycle Phases…..……………………………………......48
4. Plan…..…………………………………………………..…..49
5. Process…..…………………………………………………...50
6. Activity…..……………………………………………….…51
7. Artifact…..……………………………………………..……52
8. Rework…..……………………………………………..……53
9. Supplier…..………………………………………………......54
10. ReusableSourceFile…..……………………………….……55
11. Feature (Supplier related)…..………………………………56
12. Organization…..……………………………………..……..57
13. Individual…..…………………………………………………58
14. WorkBreakDownStructure…..…………………………..59
15. Activity (Related to WorkBreakdownStructure)……......60
16. Software Product…..………………………………………61
17. Feature…..……………………………………………………62
18. Artifact (Related to Feature)………………………….…....63
19. Customer…..…………………………………………….......64

Page

20. Problem…..……………………………………………..…...65
C. Assess Process Maturity…..…………………………………...66
1. Facts extracted from Jinhwan's project…..……………..…...66
2. Facts extracted from CPSC606 project…..……………..……67
3. Process assessment and monitoring for Jinhwan's Project…..……………………………………...68
4. Process assessment and monitoring for CPSC606 Project. ……………………………………...69

5. EVALUATION AND CONCLUSION………………………………...70

6. FUTURE WORK…....……………………………………………..…73

A. Objective COCOMO II Scale Factor Measurement..…………73
B. Requirement Measurement…..…………………………….....75

REFERENCES…..……………….……………………………..………77

APPENDIX A KPA ASSESSMENT……..……………………..…..….81

APPENDIX B PAMPA II OBJECT CLASSES…..……………………..…404

APPENDIX C MATURITY QUESTIONNAIRE………………………..441

VITA…………………………………………………………………......452

LIST OF TABLES

Table Page

1 CASE Tools..……………………………………………………………………..4

2 CASE Tools for PAMPA 2.0..……………………………………………….......14

3 Combining PAMPA 2.0 with CASE Tools.……………………………………17

4 PAMPA 2.0 Knowledge Base and Key Practices..……………………….…..30

5 Mapping the Questionnaire to KPA Goals and Practices..…………………..…..35

6 Subjective & Objective Activity Status..………………………………….…40

7 Assessment Results..……………………………………………………….…70

LIST OF FIGURES

Figure Page

1 PAMPA 2.0 Knowledge Base Framework……………………………………..…3

2 Integration of Rational Tools & MS Project..…………………………………...15

3 PAMPA 2.0 Architecture..………………………………………………………16

4 CMM Assessment Procedure..…………………………………………………..20

5 Capability Maturity Models..……………………………………………………22

6 CMM & Risk……………………………………………………………23

7 Relationships of CMM Questionnaire, Goal, KPA, and CMM.………….........31

8 Project List..……………………………………………………………………...46

9 Project..………………………..………………………………………………47

10 Life Cycle Phases..……………………………………………………………..48

11 Plan..……………………………………………………………………………49

12 Process..………………………………………………………………………...50

13 Activity..………………………………………………………………………..51

14 Artifact..………………………………………………………………………...52

15 Rework..…………………………………………………………………..….53

16 Supplier..………………………………………………………………………..54

17 ReusableSourceFile..…………………….………………………………..…55

18 Feature (Supplier related)…………………..………………………………..…56

19 Organization..…………………………………………………………………..57

20 Individual..……………………………………………………….…….……58

Figure Page

21 WorkBreakDownStructure..59

22 Activity (Related to WorkBreakdownStructure)......................................60

23 Software Product...61

24 Feature..62

25 Artifact (Related to Feature)..63

26 Customer..64

27 Problem...65

28 Facts from Jinhwan's Project...66

29 Facts from CPSC606 Project..67

30 Assessment Result from Jinhwan's Project..68

31 Assessment Result from CPSC606 Project..69

32 COCOMO II Scale Factor Prediction...74

33 Requirement Measurement..76

1. INTRODUCTION

A large software development project includes many people such as managers, developers, customers, etc. In order to succeed on a large complicated project, you need to work cooperatively with everyone involved. For a project manager it is important to monitor activities to improve a process. Developers should understand their assigned tasks, and implement them following a planned time schedule.

This dissertation introduces a solution for project monitoring and assessment that saves effort and time by using PAMPA 2.0 (Project Attribute Monitoring and Prediction Associate), which includes the concept of metrics, expert system, knowledge base, and CASE (Computer Aided Software Engineering) tools.

A. Metrics

Metrics describe attribute values of a software development project. Perlis et al. illustrated many types of metrics such as application specific measures (compilers, interactive systems, virtual memory and paging systems, protection, reliability, testing techniques, statistical and numerical software), cost estimation, human factors, maintenance and enhancement, productivity measurement, performance evaluation, software life cycle, software monitoring (timing, sampling, event monitoring, special hardware, software systems, network monitoring), product, plan, organization, supplier,

---

The journal model is *IEEE Transactions on Engineering Management.*

and customer [9, 16, 17, 18, 19]. Simmons et al. showed the possibility that metrics can drive an expert system to objectively assist managers in directing software projects [2, 3, 4].

B. PAMPA 2.0

The PAMPA visualization toolkit was created to help managers control projects and improve processes [1]. The PAMPA 2.0 tool knowledge base was recently created to describe plans based on a spiral life cycle [5]. The expanded tool is used with a Software Project Planning Associate (SPPA) that can track work breakdown packages' compliance to plans [6]. Attributes gathered in PAMPA 2.0 can be used for process improvement, project control, and software product quality control.

The knowledge-based framework is described by the UML (Universal Modeling Language) diagram in Figure 1 [5]. Object classes that make up projects are displayed in the boxes. Knowledge base object attributes and relationships among the objects reflect the status of a project. Time images are periodically saved as **ProjectVersion**s and **SoftwareProduct Version**s. Snapshots of all aspects of a **Project** can be replayed in a manner similar to how airline flight recorders replay flight data to determine what happened during a flight before a plane crashes.

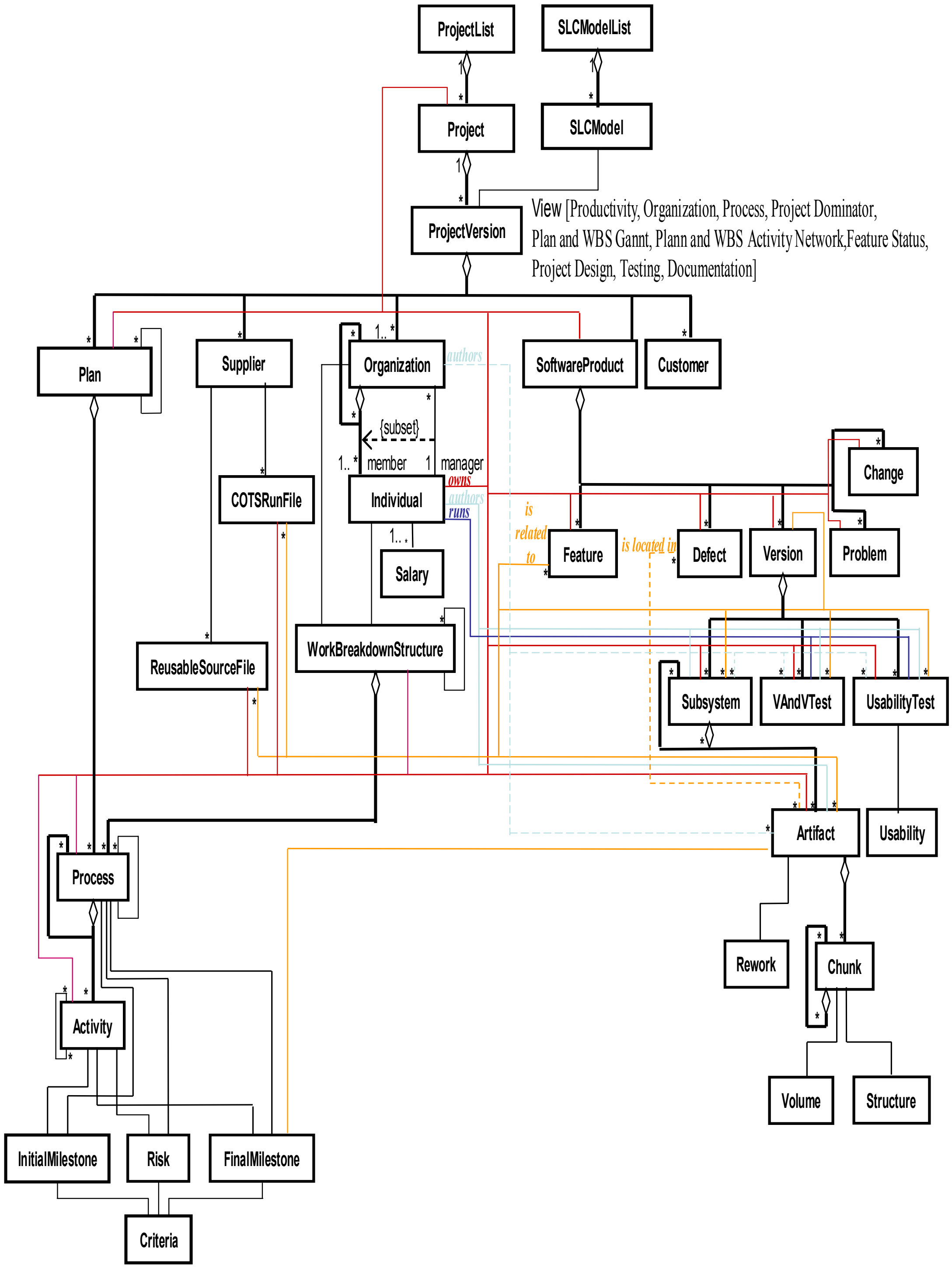


Figure 1. PAMPA 2.0 Knowledge Base Framework

C. CASE Tools.

In a complex software development environment, it is not easy to manage a software development project. To improve software development, CASE tools are designed for several software development purposes [Table 1]. Software projects use CASE tools for configuration management, project planning, requirements management, financial management, defect and change tracking, and software test management [21].

Table 1. CASE Tools

Tool Name	Vendor	Description
+1CM	+1 Software Engineering	Configuration Management Tool
AIM	Intergraph	Asset and Information Management (AIM) allows Windows® and UNIX users fast and intuitive access to the entire information base of the enterprise.
Aldon/CMS	Aldon Computer Group	Configuration Management Tool
AllChange	Intasoft Ltd.	Configuration Management Tool
Andromede	Jean-Francois Comber	Configuration Management Tool
CA-Endevor	Computer Associates	Configuration Management Tool
CCC/Harvest	Platinum Technology	Configuration Management Tool
Changeman	Optima Software	Configuration Management Tool
ClearCase	Rational Software	Configuration Management Tool
CM Windows	ISDE	Configuration Management Tool
CMS	Electronic Warfare Associates, Inc.	The CMS tool provides a mechanism for source code and documentation revision control, base lining, and software builds. RCS and SCCS are utilized as the backend revision control systems.
CMWin	Expertware, Inc.	Configuration Management Tool
CMZ	CodeME s.a.r.l	Configuration management tool
Code Co-op	Reliable Software	Configuration Management Tool

Table 1 continued.

Tool Name	Vendor	Description
Configuration Management Version Control (CMVS)	IBM Corp.	Configuration Management Tool
Continuus/CM	Continuus Software	Configuration Management Tool
CONTROL-CS	Network Concepts, Inc.	Configuration Management Tool
Corporate RCS	Thompson Automation	Configuration Management Tool
CVS	Cyclic Software	Configuration Management tool
DRCS	Software Services and Solutions, Inc.	Configuration Management tool
E!Vista	Metaphase Technology	Java-based Configuration Management tool
Neuma	Neuma Technology Corp	Configuration Management tool
Perforce	Perforce Software	Configuration Management Tool
PVCS Version Manager	Intersolv	Software version control system
Razor	Tower Concepts, Inc.	CM and problem tracking tool
RDM	Interleaf Corporation	Document management system
Source Integrity	MKS	Software version control system.
StarTeam	Starbase	Software configuration management and defect tracking tool
Team Connection	International Business Machines	Configuration Management Tool
TeamSite	Interwoven, Inc.	Configuration Management tool
TRUEchange	TRUEsoftware	Configuration Management Tool
Visual Enabler	Softlab	Configuration Management tool
Visual SourceSafe	Microsoft	Software version control system for managing software and Web site development
Web Integrity	MKS	Web object management system that manages all types of static and dynamic content and components, including text, html, graphics, and Java.
+1CR	+1 Software Engineering	Problem tracking tool
Bugcollector Pro	Nesbitt	Defect tracking tool
Census	MetaQuest	Defect tracking tool

Table 1 Continued.

Tool Name	Vendor	Description
Code Integrity	MKS	C source code static analysis tool that helps organizations manage their code development process. Code Integrity helps across all phases of the application development cycle; planning, development, quality assurance, and support.
Defect WorkFlow	SoftQuest Systems	Defect tracking and change management tool
DevTrack	TechExcel	Defect and project tracking tool. Features Internet integration, universal ODBC support, multi-user scalable client/server architecture, email support, and presentation-quality reports/graphics. Sports extensive customization and all-field searching.
GTbug	Globetrotter	Defect tracking tool
PR-Tracker	Softwise	Defect tracking tool
Project Management Tool Suite	Electronic Warfare Associates, Inc.	Series of Project Management Tools for UNIX and PC based systems. Includes: Project Action Item, System Trouble Report, System Change Request, System Test Description, Design and Code Walkthrough, CM, Requirements Traceability, Electronic timesheet.
PVCS Tracker	Intersolv	Software Problem report tracking
TeamTrack	TeamShare	Problem tracking and change management system that facilitates customer feedback
TestTrack	Seapine Software	Software testing utility
TRACK	Soffront	Defect tracking, Help Desk and Assetmanagement software
Track Integrity	MKS	Defect tracking tool
Track Record	UnderWare Inc.	Software development tool for tracking bugs, features, releases and other details associated with software projects. Interfaces to Visual SourceSafe, MKS Source Integrity and Intersolv PVCS.
Track Record	Numega	Defect tracking tool
Visual Intercept	Elsinore Technologies, Inc.	Project-oriented, three-tiered, enterprise-ready incident management system specifically designed for Microsoft Visual Tool users
@RISK	Palisade Corporation	Risk analysis and simulation add-in for Microsoft Excel and project, and Lotus 1-2-3. Replace values in your spreadsheet with @RISK distributions to represent uncertainty, then simulate your model using Monte Carlo simulation methods.
@RISK for Project	Palisade Corporation	Add-on risk analysis tool to analyze Microsoft Project schedules using Monte Carlo simulations of tasks and resources.
ARMS	RightWare, Inc.	Risk Management tool. Includes Risk planning with Microsoft Project, Risk Identification with eRisk tool, and Risk analysis

Table 1 Continued.

Tool Name	Vendor	Description
Cascade	Mantix Systems	Supports the project world of the future as a critical information system for senior executives; program, project, financial, and functional managers; as well as team members
Crystal Ball	Decisioneering	Forecasting and Risk Analysis Add-in for Microsoft Excel
DATA	TreeAge Software	Systematic methodology and framework for understanding a problem, identifying available options, evaluating options in the context of associated uncertainties
eRisk	RightWare, Inc.	Web-based companion tool for ARMS 2000, used for viewing and identifying risks over your company's intra or internet.
RAMAS Risk Calc	Applied Biomathematics	Performs a what-if analysis using classical interval analysis and its generalization, fuzzy arithmetic. Variability and uncertainty, fuzzy and probabilistic arithmetic: organic soil contaminants, remediation planning, QA for probabilistic risk analyses.
REMIS	Price Systems	Risk Evaluation & Management Information Systems
Risk Driver	Decision Products Inc.	Combined database tool and analysis tool. The WBS must be loaded along with the cash flow and time duration of each activity. Risks are loaded as performance, cash flow, and duration, and then are associated to the impacted WBS element(s).
Risk Master	Sphygmic Software Ltd.	Risk analysis tool for project schedule and cost that applies Monte Carlo simulation. Accepts project data from standard project management tools. SQL-driven report writer allows data retrieval
Risk Radar	Software Program Manager’s Network	Risk management tool in Excel
Risk+	C/S Solutions Inc.	Risk analysis add-on for Microsoft Project to quantify the cost and schedule uncertainty associated with project plans using Monte Carlo-based simulation techniques.
RiskTrak	Risk Services & Technology	Database tool to mange risk items. Track, prioritize, and organize risk information. Provides reports. Imports from, and exports to, any fully ODBC-compliant database.
RISKview	Palisade Corporation	Distribution viewer and analysis tool. Allows modification of parameter values for canned distributions. Distributions can be used in @RISK. Distribution Artist hand drawing of distributions
STAR	Mainstay Software	System Trades and Risk
TRIMS	U.S. Navy	Technical Risk Identification and Mitigation System (TRIMS) a Risk Management Tool. Based on Willoughy templates (DoD4245.7M - Transition from Development to Production).

Table 1 Continued.

Tool Name	Vendor	Description
wInsight	C/S Solutions Inc.	Integrates cost performance measurement data with schedule status. Optional interface to MS Project, Open Plan, or SureTrak. Cost risk analysis on performance measurement data via Risk+.
ACE IT	Air Force and Army Joint Program	Automated Cost Estimating Integrated Tools. An estimating system containing tools to assist in conducting cost analysis activities such as cost estimates, what-if studies, cost proposal evaluations, risk, and uncertainty analysis
AMCOS	U.S. Army Cost and Economic Analysis Center (USACEAC)	Army Military Civilian Cost System. Manpower estimation tool. Used for estimating: costs of manning new weapon systems over its life cycle, cost trade-offs of alternative weapon systems, cost of adding new positions to the force structure.
CASA	U.S. Army Logistics Center	Cost Analysis and Strategy Assessment. Derived from Honeywell's Total Resource and Cost Evaluation (TRACE) Logistics Support and Life Cycle Cost Models. User for LCC estimates, trade-off analysis, repair level analysis, production rate, and quantity
COCOMO	University of Southern California	COnstuctive COst MOdel for effort, cost and schedule estimation of incremental software development. COCOMO model published in Software Engineering Economics by Dr. Barry Boehm. The Intermediate COCOMO model, USC COCOMO, is available free
COCOMO II	University of Southern California	An update of COCOMO 1981 to address software development practices in the 1990s and 2000s. Tailorable mix of models. The Application Composition Model uses Object Point count for the early prototyping phase to resolve high-risks issues
CoCoPro	ICONIX Software Engineering	Implements Boehm's Constructive Cost Model (CoCoMo) technique for estimating costs of software projects. It supports the intermediate CoCoMo model, and allows automatic calibration of the model to a cost history database.
COOLSoft	Wright Williams & Kelly	A hybrid model using intermediate and detailed COCOMO. Allows for the reuse of existing code, development of new code, the purchase and integration of third-party code, and hardware integration. The output is effort, calendar schedule, support costs.
Cost Xpert	Marotz, Inc.	Cost Estimation
Cost$Benefit Analysis Tool	Legacy Systems Research	Cost-benefit analysis software suitable for environmental and exploration investment activities at any stage of a project/decision for go/no go, choosing alternatives, proposal evaluation, acquisition strategy, long-range plans.
Costar V5	Softstar Systems	Supports all COCOMO models. Version 5 includes COCOMO II models. V4 is the version available on RTIS LAN under Project Management Apps.

Table 1 Continued.

Tool Name	Vendor	Description
ENRV	U.S. Air Force Cost Analysis Agency	Revised Intermediate COCOMO and Enhanced REVIC Advisor. Differences are: equation coefficients are revised based on calibration using DOD projects, provides a single weighted "average" distribution for effort and schedule, and the ability for user to vary
Foresight	Price Systems	Cost Estimation
GECOMO	Marconi	Cost estimation tool
KnowledgePLAN	Artemis Management Systems	Software cost estimation tool that uses a knowledge base of 3,000 projects.
Monte Carlo	Primavera Systems Inc.	Utilizes project schedules and simulates project performance to assess likelihood of finishing on time and within budget
OPEN PLAN	Welcom Software Technology	Resource allocation
PES	Price Systems	PRICE Estimation Suite (PES). Parametric cost-estimating tool. Development and production costs and schedules for systems of electronic, electromechanical, and structural assemblies. Has links to RDD-100
PRICE H	Price Systems	Parametric cost-estimating tool. Development and production costs and schedules for systems of electronic, electromechanical, and structural assemblies. Has links to RDD-100 tool.
PRICE HL	Price Systems	Parametric cost-estimating tool. Operation and support costs for maintaining deployed hardware systems throughout their lifecycle. Has links to RDD-100 tool.
PRICE M	Price Systems	Parametric cost-estimating tool. Development and production costs and schedules for low-level electronics: IC, ASICs, MCMs, SEMs, Packaging, Printed Circuit Cards, etc. Has links to RDD-100 tool.
PRICE S	Price Systems	Parametric cost-estimating tool. Software development, maintenance, and support costs and schedules for total software cost of ownership. Has links to RDD-100 tool.
ProjectView	Artemis Management Systems	Enterprise business solution that brings project planning, cost control, resource tracking, and project analysis to the heart of the organization. Using a unique role-based approach to software design and implementation.
REVIC	U.S. Air Force Cost Analysis Agency	Revised Intermediate COCOMO and Enhanced REVIC Advisor. Differences are: equation coefficients are revised based on calibration using DOD projects, provides a single weighted "average" distribution for effort and schedule
SEAT		Software Estimation and Analysis Tool. Integrates Functional Point Analysis (FPA) for LOC estimation and COCOMO for effort and schedule estimation.

Table 1 Continued.

Tool Name	Vendor	Description
SEER-H	G A SEER Technologies	System Evaluation and Estimation of Resources (SEER). Hardware Estimation Model to aid in estimation of the development and production cost, scheduling, and risks associated with hardware acquisition.
SEER-HLC	G A SEER Technologies	Life cycle cost tool. Used during any program phase, from Concept Study and through a program's entire development, investment, and operational life. Outputs from other SEER models can provide SEER-HLC inputs.
SEER-SEM	G A SEER Technologies	Tools for estimating software development and maintenance cost, schedule risk, and reliability. Parameters handle spiral, prototype, evolutionary, or object-oriented development. Provides an independent assessment of the effective SEI rating for the par
SEER-SSM	G A SEER Technologies	System Evaluation and Estimation of Resources (SEER). Tools for estimating software size in terms of LOC or functional size.
SLIM	Quantitative Software Management	Putman's Software Life Cycle Model
SmartCost	Knowledge Based Systems, Inc.	Capture best-practice cost estimation knowledge with SmartCost, and make it available when and where it is needed. Integrate multiple knowledge sources to generate total-cost-of-ownership predictions
Views	Artemis Management Systems	Enterprise business solution that brings project planning, cost control, resource tracking, and project analysis to the heart of the organization. Using a unique role-based approach to software design and implementation.
ActionPlan	Netmosphere Inc.	Web-based project management tool
AIO WIN	Knowledge Based Systems, Inc.	Function Modeling tool that offers Activity-Based Costing (ABC) support and captures process time and resource costs for activities, resources, and products.
AutoPLAN	Digital Tools, Inc.	Project scheduling tool
CA-SuperProject	Computer Associates	Project scheduling tool
CAT II	Robbins Gioia Inc.	Gives you a clear picture of project status every step of the way. Strategic goals can be achieved by empowering your team with the ability to view all aspects of your program, perform What If analysis, and understand the full impact of trade-offs.
COBRA	Welcom Software Technology	Cost Management

Table 1 Continued.

Tool Name	Vendor	Description
Expedition	Primavera Systems Inc.	Expedition helps ensure all materials necessary for a project are designed, specified, ordered, and delivered on time to avoid delays
FastTrack Schedule	AEC Software	Project Scheduling Tool
JustProgress	Mainstay	Web-based Gantt chart applets used to plan, track, and display project status.
MicroPlanner	Micro Planning International Ltd.	Integrated suite of project management software, operating across the major hardware platforms - MacOS and Microsoft Windows; based on the technique of Critical Path Analysis
Milestones Etc.	KIDASA Software Inc.	Project scheduling tool
Office Timesheet 98	Tenrox	Time and expense tracking, project reporting
Primavera Project Planner	Primavera Systems Inc.	Multi-user project scheduling tool
ProChain	ProChain Solutions	Project scheduling tool that implements the Critical Chain approach
Project 98	Microsoft	Project cost and schedule tracking tools compatible with ODBC-compliant databases. Visual Basic?for Applications built in.
Project Scheduler	Scitor Corp.	Project scheduling tool
PROVISA	Lanner Group Ltd.	Powerful and flexible computer-based Finite Capacity Scheduling system which gives you the ability to schedule your complete plant or business, taking into account all resource constraints including machines, labor, raw materials, tools, fixtures, etc.
SureTrak	Primavera Systems Inc.	Resource planning and control tool for small-to medium-sized projects.
Time Line	Time Line Solutions	Project scheduling tool
TrackView	Artemis Management Systems	Enterprise business solution that brings project planning, cost control, resource tracking, and project analysis to the heart of the organization. Using a unique role-based approach to software design and implementation.
TRAKKER	Dekker	Integrates schedule, process flow, resources, costing, earned value, technical performance, and revenue projections into one database
AnalystStudio	Rational Software	Tool Suite. Includes RequisitePro, Rose, SoDA, and ClearCase
Caliber-RM	Technology Builders, Inc (TBI)	Requirements traceability tool

Table 1 Continued.

Tool Name	Vendor	Description
CORE	Vitech Corporation	Full life-cycle systems engineering CASE tool. It supports the systems engineering paradigm from the earliest days of concept development and proposal development, requirements management, behavior modeling, system design, and verification process.
CRADLE/REQ	3SL (Structured Software Systems)	Requirements Management tool capable of storing, within its database, graphs, spreadsheets, tables, diagrams, and any other information as part of a requirement.
DOORS	Telelogic (was QSS)	Requirements traceability tool
DOORS/ERS	Telelogic (was QSS)	Enterprise Requirements traceability tool suite
DOORSrequireIT	Telelogic (was QSS)	Requirements trace tool that is integrated with Microsoft Word. Data can be merged with DOORS databases
GMARC	Computer Systems Architects (CSA)	Generic Modeling Approach to Requirements Capture (GMARC). Toolset will also generate quality metrics for a specification enabling formal proof that use of the GMARC has improved the requirement set.
icCONCEPT	Integrated Chipware	Requirements traceability tool. Replaces RTM
IRqA	TCP Sistemas e Ingenieria	Integral Requisite Analyzer. A requirements management tool, but also a requirements analysis environment, that includes facilities to support problem domain modeling and automatic domain analysis.
ITraceSE	ITrace Systems	Requirements traceability tool
Life*CYCLE	Computer Resources International	Requirements traceability tool. (No longer available)
RDT	IGATECH Systems Pty Limited	Requirements traceability tool
RequisitePro	Rational Software	Requirements traceability tool. Also part of AnalystStudio
RIMS	Sygenex Incorporated	Requirements and Information Mamagement System (RIMS).
RTM	Integrated Chipware	Requirements traceability software. See icCONCEPT product.
RTS	Electronic Warfare Associates, Inc.	Requirements Traceability Systems (RTS). Complete foundation for tracking the requirements of a software/hardware project through the accompanying documentation and source code. This includes tracking the development and testing status of requirements

Table 1 Continued.

Tool Name	Vendor	Description
SLATE	SDRC SSG	System Level Automation Tool for Engineers (SLATE) is used to capture, organize, build, and document system-level designs from raw concepts through structural partitioning. Interfaces to Office 97, Project and CASE tools.
Systems Engineer	Blue Spruce	Requirements trace tool
Tofs	Tofs AB	Tool For Systems. Assists you in realizing and managing not only software, but also the manual (human) and hardware (electronic, hydraulic, mechanic, etc) parts of a system, which complete the system's missions together with the software.
Tracer	RBD, Inc.	Requirements traceability tool
Vital Link	Compliance Automation Inc.	Requirements traceability tool
XTie-RT	Teledyne Brown Engineering	Requirements traceability tool

Table 2 shows the selected CASE tools, produced by Rational and Microsoft, for the PAMPA 2.0 working as an attribute-gathering subsystem [26]. In this dissertation only these tools are used because of the convenience that these tools provide in planning, designing, configuration management, requirements management, defect tracking, and testing in the software development project. It also has the advantage of integrated tool use because most of the tools have come from the same company.

Table 2. CASE Tools for PAMPA 2.0

CASE Tools	
Name	**Purpose**
ClearCase	Configuration Management. Gather turmoil metrics.
RequisitePro	Requirements management. Gather requirement related metrics.
Rational Rose	A graphical component modeling and development tool using UML. Gather design specification metrics and reuse metrics.
ClearQuest	Defect and change tracking. Gather reliability and rework metrics.
Microsoft Project	Project planning (planning, organizing, and tracking a project's tasks, and identifying and scheduling resources to accomplish those tasks). Gather schedule, task, activity network, and cost information metrics.
Rational Robot, Test Manager	Gather Regression Testing, V&V Testing, Usability Testing, Functional testing metric.

Figure 2 shows the integrated procedure of CASE tools. RequisitePro manages software project development requirements and documents. Requirements are mapped into both Rational Rose models and MS Project plan. The design model in the Rose can be converted into program file format such as Java, which will be version-controlled in the ClearCase. Software development-related documents and requirements from RequisitePro are also version-controlled on ClearCase. When a new version of file is created in the ClearCase it is called an **Artifact**, which could be a source code or a document. The procedure to make an **Artifact** is called **Activity**, which is managed by ClearQuest. This **Activity** will be compared to the RequsitePro requirements and MS Project plan. And a new version of **Artifact**, which is a source code, is tested on Rational Robot for functional test, usability test, and regression test. This procedure follows the software development Life Cycle, which includes planning, designing, building, and testing.

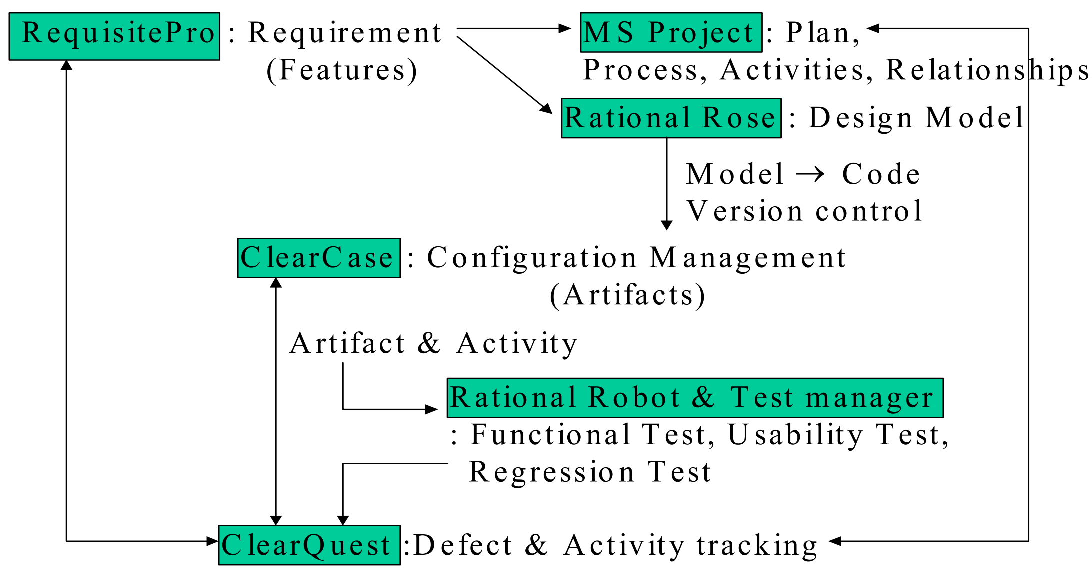


Figure 2. Integration of Rational Tools & MS Project

D. PAMPA 2.0 Architecture Overview

Figure 3 depicts the architecture of the PAMPA 2.0. The PAMPA 2.0 uses a three-tier architecture. A thin client such as an Internet browser or a handheld device represents the presentation tier. The middle tier comprises the PAMPA 2.0 Application components. These components are hosted in an Apache tomcat engine/web server. The presentation tier communicates with the application components through http requests to the web server. The application components act as a middleware that integrates various tools like JESS, Rational tools and Microsoft Project. A PAMPA 2.0 component gathers attributes from Rational tools, and MS Project and store them into the PAMPA 2.0 SQL database. Rules and initial facts from **Criteria** and facts retrieved from PAMPA Objects are sent to the Java Expert System Shell (JESS), to make inferences. These inferences in text format are then pushed to an Internet browser. These application components can be viewed as a web-based gateway to the tools and the knowledge base. The application components use JDBC (JAVA Database Connectivity), which interact with the third tier. The third tier containes the PAMPA 2.0 knowledge base that is stored on a Microsoft SQL Server 2000.

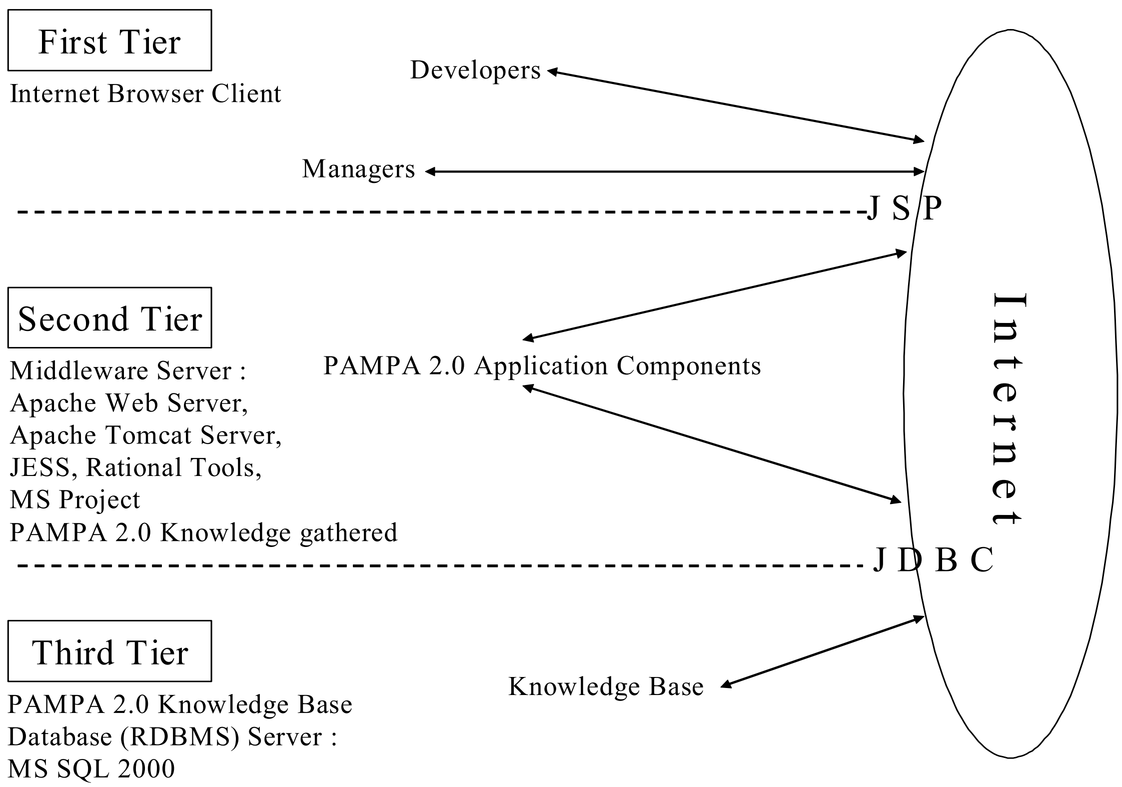


Figure 3. PAMPA 2.0 Architecture

E. Combine PAMPA 2.0 with CASE Tools.

Knowledge about a software project is gathered from CASE tools and stored into the PAMPA 2.0 knowledge base as objects, each of which has relationships to other objects and attributes. Table 3 shows the CASE tool sources for the PAMPA 2.0 knowledge base.

Table 3. Combining PAMPA 2.0 with CASE Tools

PAMPA 2.0			Source
Object	**Attribute**	**Relationships**	
ProjectList	***Name***$_{Set,}$ ***Description***$_{Set}$	A **ProjectList** contains **Project**s.	MS Project
Project	***Name***$_{Set,}$ ***Description***$_{Set,}$ ***Overhead***$_{Set,}$ ***Cost***$_{Calculate,}$ ***EffortToDate***$_{Calculate,}$ ***HeadCount***$_{Calculate,}$	**Project** is contained in a **ProjectList.**	
ProjectVersion	***Name***$_{Set,}$ ***Description***$_{Set,}$ **Cost**, **Time**	**ProjectVersion** is contained in a **Project.**	
Plan	***Name***$_{Set,}$ ***Description***$_{Set}$	**Plan** is part of a **ProjectVersion.**	
Process	**InitialMilestone**, **Risk**, **FinalMilestone**, ***Name***$_{Set,,}$ ***Description***$_{Set}$	**Process**es are contained in a **Plan**, **WorkBreakdownStructure.**	
Activity	**InitialMilestone**, **Risk**, **FinalMilestone**, ***Name***$_{Set,,}$ ***Description***$_{Set}$	**Activit(y)**ies are contained in a **Process**, related to **Activit(y)**ies.	ClearQuest
InitialMilestone	**Criteria**, ***PlannedStartDate***$_{Set,}$ ***AcualStartDate***$_{Set}$	**InitialMilestone** is an attribute of **Process**, **Activity**.	MS Project ClearQuest
Risk	**Criteria**, ***EstimatedRisk***$_{Set,}$ ***Description***$_{Set}$	**Risk** is an attribute of **Process**, **Activity**.	Elicited from Expert
FinalMilestone	**Criteria**, ***PlannedEndDate***$_{Set,}$ ***ActualEndDate***$_{Set}$	**FinalMilestone** is an attribute of **Process**, **Activity**.	MS Project ClearQuest
Criteria	***Knowledge***$_{Set}$(Describing when criteria 1 is met) ……..………….. ***Knowledge***$_{Set}$ (Describing when criteria n is met)	**Criteria** is an attribute of **InitialMilestone**, **Risk**, **FinalMilestone.**	Elicited from Expert
Supplier	***Name***$_{Set,}$ ***Description***$_{Set}$	**Supplier**s are contained in a **ProjectVersion.**	Project Object
ReusableSourceFile	***Name***$_{Set,}$ ***Description***$_{Set}$	**ReusableSourceFile**s are provided by a **Supplier.**	ClearCase, Rational Rose
COTSRunFile	***Name***$_{Set,}$ ***Description***$_{Set}$	**COTSRunFile**s are provided by a **Supplier**, related to **Feature**s.	Project Object
Organization	***AverageIndividualProductivity***$_{Calculate,}$ ***DefectRate***$_{Calculate,}$ ***Efficiency***$_{Calculate,}$ ***Productivity***$_{Calculate,}$ ***Speedup***$_{Calculate}$	**Organization**s contain **Organization**s, perform **WorkBreakdownStructure.**	MS Project
Individual	***Experience***$_{Set,}$ ***OverheadFactor***$_{Set}$(≥ 1), ***Title***$_{Set,}$ ***DefectRate***$_{Calculate}$, ***Productivity***$_{Calculate}$	**Individual** authors **Subsystem**s, **Deliverable**s, **VandVTest**s, **UsabilityTest**s, performs **WorkBreakdownStructure.**	
Salary	***Amount***$_{Set,}$ ***EffectiveDate***$_{Set}$	**Salar(y)**ies are related to an **Individual**.	
WorkBreakdown Structure	***Name***$_{Set,}$ ***Description***$_{Set}$	**WorkBreakdownStructure** is associated with an **Organization**, **Individual**, **WorkBreakdownStructure**s.	MS Project ClearQuest
SoftwareProduct	***Name***$_{Set,}$ ***Description***$_{Set,}$ **Reliability**, **Usability**, **Volume**	A **SoftwareProduct** is contained in a **ProjectVersion.**	Project Object

Table 3 continued.

PAMPA 2.0			Source
Object	**Attribute**	**Relationships**	
Feature	***Name***$_{Set,}$ ***Description***$_{Set,}$	**Feature**s are contained in a **SoftwareProduct.**	RequisitePro
Defect	***Name***$_{Set,}$ ***Description***$_{Set,}$ ***Identification***$_{Set}$(number)	**SoftwareProduct** contains **Defect**s.	ClearQuest
Version	**PreviousVersionIdentification**, **SourceDir**, **VersionIdentification,** ***Complete***$_{Set}$(Yes=1, No=blank or 0), **Defect**s, **VersionCreated**(date)	**Version**s are contained in a **SoftwareProduct**, contains **Subsystem**s, **VandVTest**s, **UsabilityTest**s, owned by an **Individual**, related to **Feature**s.	ClearCase
Subsystem	***Name***$_{Set}$, **Type**(RequementsFile, DesignFile, DocumentFile, SourceFile), ***Complete***$_{Set}$(Yes=1, No=blank or 0)	**Subsystem**s are contained in a **Version**, contains **Subsystem**s, **Deliverable**s.	ClearCase
Artifcat	**Rework**, ***Name***$_{Set}$, ***Language***$_{Set}$(Natural Language, Program Language(Ada, C++, Fortran, Java, etc.))	**Artifact** contains **Chunk**s, has attribute **Rework**, is authored by an **Individual**, **Organization.**	ClearQuest, ClearCase
Chunk	**Structure**, **Volume**, **Name**	**Chunk**s are contained in **Artifact**, contain **Chunk**s.	Elicited from Metrics
Volume	**BytesSourceCode**, **FunctionPoint**s, **Operand**s, **Operator**s, **SLOC**	**Volume** contains attributes of a Chunk.	
Structure	**EssentialComplexity**, **InheritanceDepth**	**Structure** contains attributes of a **Chunk**.	
Rework	**Add**s, **Change**s, **Delete**s, ***Turmoil***$_{Calculate}$	**Rework** contains attributes of a **Deliverable**.	ClearCase
VandVTest	**Status**(Failed/Passed), **Failure**(YES/NO), **Date**, **InputFile**s, **CoverageVector**(% by source)	**VandVTest** is contained in a **Version**, authored by an **Individual**, **Organization.**	Rational Robot
UsabilityTest	**Usability**, **InputFile**s, **Duration**, **Status**(Failed/Passed, i.e. Usability Test is ready or not ready to be run.)	**UsabilityTest**s are contained in a **Version**, authored by an **Individual**, **Organization.**	Elicited from Testers
Usability Attributes	**Response**s, **HelpRequest**s, **Efficiency**, ***Confidence***$_{Set,}$ ***Difficulty***$_{Set,}$	**Usability** contains attributes of a **UsabilityTest**.	RequisitePro ,Test Manager
Customer	***Name***$_{Set,}$ ***Description***$_{Set,}$ **Performance**, ***ExperienceLevel***$_{Set,}$ ***Satisfaction***$_{Set}$	**Customer**s are related to **ProjectVersion**.	ClearQuest
SLCModelList	***Name***$_{Set,}$ ***Description***$_{Set,,}$ Pointers to SCL models		
SLCModel	Waterfall, V SLC, Incremental, Prototype, Spiral		

2. OBJECTIVE CMM ASSESSMENT

A. CMM Assessment Procedure

The Department of Defense Software Engineering Institute (SEI) has developed a five-level CMM to evaluate and assess the processes used to create software [12]. Guidelines for using the CMM to improve the software process are based on subjective assessment by teams of experts [15]. Subjective assessment requires expensive human-based teams who can be replaced by expert systems. Knowledge can be acquired from software development experts to create a knowledge base. Metrics gathered from a development environment can drive an expert system to objectively assist managers in directing software projects [2, 3, 4].

This chapter describes how the PAMPA 2.0 tool can be extended based on the CMM to objectively assess the maturity of a software organization. The five-level CMM is based on 18 KPAs (Key Process Areas). Each KPA is composed of 2 to 4 processes. These processes are broken down into numerous sub-processes and activities. Knowledge in the form of rules, objects, attributes, relationships, and properties are being acquired to assess each of 18 KPAs. Once this has been done, the CMM level can be determined [Figure 4].

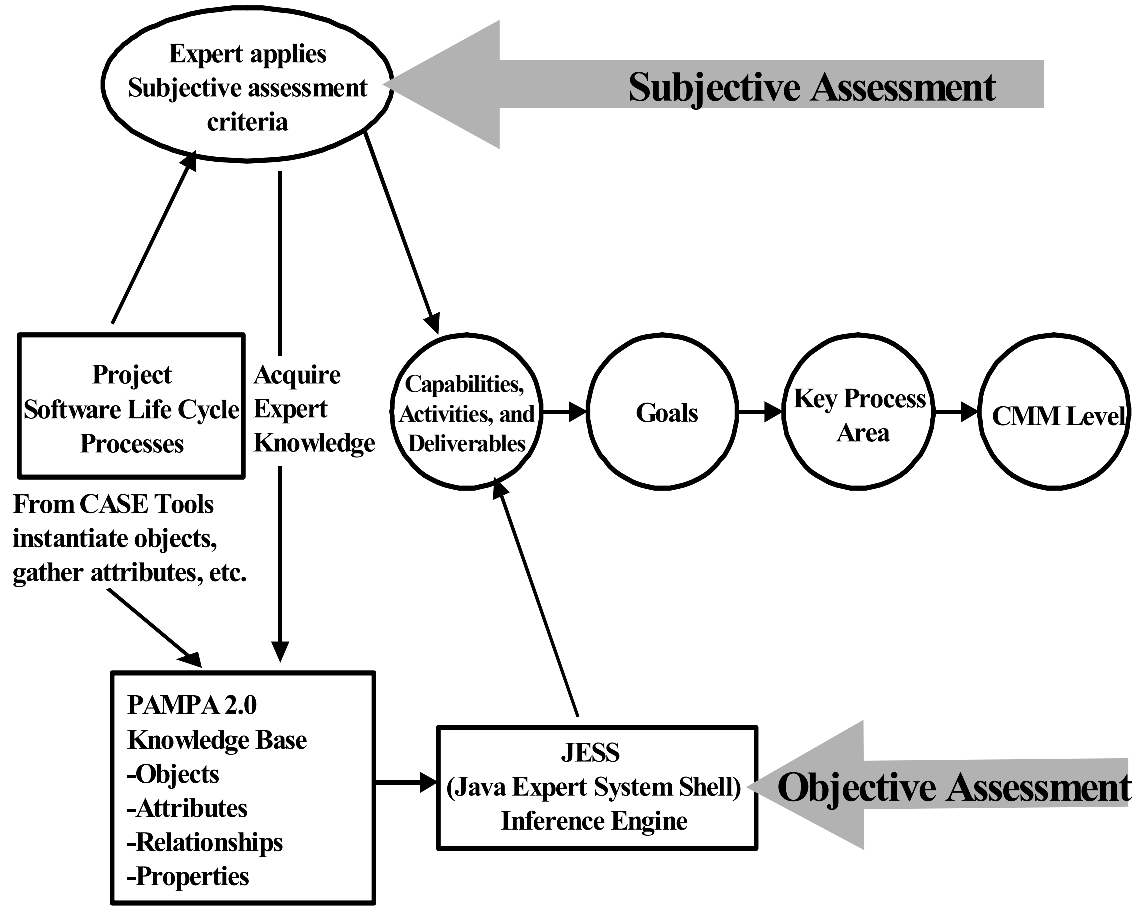


Figure 4. CMM Assessment Procedure

B. What is CMM?

The Department of Defense Software Engineering Institute (SEI) presents several types of CMMs [10]. Each addresses a different domain. The Software Capability Maturity Model (SW-CMM) helps software development organizations increase predictability in the development of software-intensive systems and software-related products. The People Capability Maturity Model (P-CMM) is a framework that helps organizations successfully address their critical people issues. Based on the best current practices in fields such as human resources, knowledge management, and organizational development, the P-CMM guides organizations in improving their processes for managing and developing their workforces. The Software Acquisition Capability Maturity Model (SA-CMM) is a capability maturity model for organizations that acquire or procure software-intensive systems. The Integrated Product Development Capability Maturity Model (IPD-CMM) is a framework to guide organizations in IPD design, development, appraisal, and improvement. The Systems Engineering Capability Maturity Model (SE-CMM) describes the essential elements of an organization's systems engineering process that must exist to ensure good systems engineering. The Capability Maturity Model Integration (CMMI) provides guidance for improving your organization's processes and your ability to manage the development, acquisition, and maintenance of products and services. The Federal Aviation Administration integrated Capability Maturity Model (FAA-iCMM) integrated the SW-CMM, the SE-CMM, and the SA-CMM, for use as a model for process improvement within the FAA [11]. The relationships among these CMMs are closely related, and most of them are derived from SW-CMM [Figure 5].

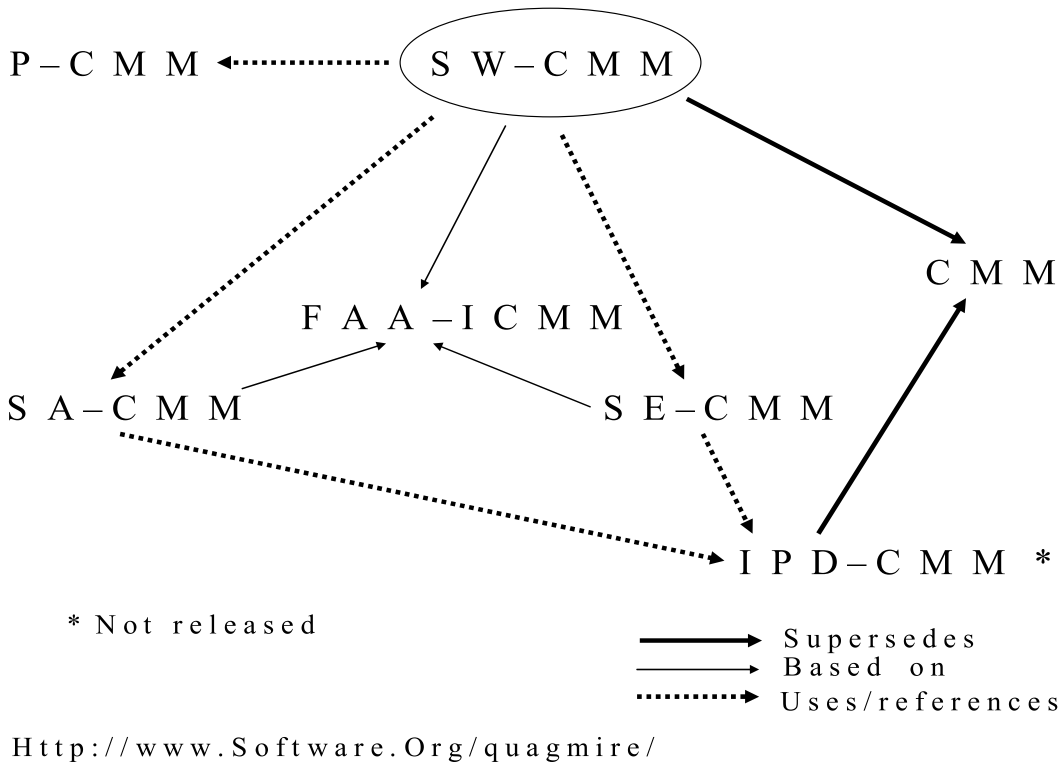


Figure 5. Capability Maturity Models

C. Why Do We Need to Measure Software Project Attributes Related to CMM KPAs?

By using the Capability Maturity Model (CMM), we can evaluate and assess the processes used to create software [12]. A CMM consists of 5 levels. From level 1 to level 5, those are initial, repeatable, defined, managed, and optimizing. The level of the CMM is directly related to the process maturity of a software development organization. Bradford proved that increasing one process maturity level could reduce development effort by 4% to 11% [13]. Also, software development risk is related to the level of the CMM. Risk is reduced as the software process improves [14]. If the risk is high, the CMM level is close to level 1. If the risk is low, the CMM level is close to level 5

[Figure 6]. In order to reduce the risk and to make more projects successful, the process should be improved, which is reflected on the CMM. People can measure the software process and try to improve the process maturity. Guidelines for using the CMM to improve the software process are based on subjective assessment by teams of experts [15].

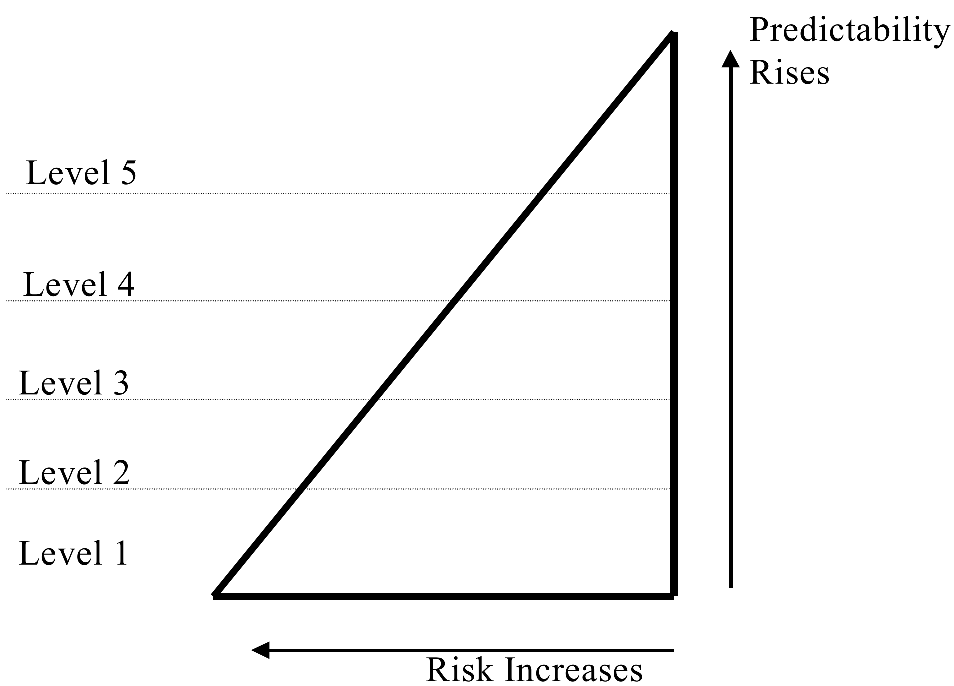


Figure 6. CMM & Risk

D. Modified CMM Assessment

The required effort and time to assess the CMM is not small. Below are several modified CMM assessments that try to save effort and time:

1. Subjective assessment using metrics

The MetricCenter Workstation [27] is a software development tool that can help CMM assessment. The MetricCenter Workstation collects the metrics needed to indicate performance of each process. It alerts any violation and shows it as specific data. The data can be shown in detailed charts for review. MetricCenter provides summary reports such as Measurement Process Report, Project Definition Report, Project Status Report, Project Data Source Report, Metric Detail Report, and Metric Audit Report, which provide the status of the project.

The metrics gathered are used as status indicator and predictor of future status. But it has the limitation that this tool uses metrics only to indicate performance of each process, not to assess the process. Compared to PAMPA 2.0 it does not have the concept of an expert system, and an expert's help is required for CMM assessment.

Gary Natwick developed *Integrated Metrics* for monitoring and alerting violation of project progress, resources, quality, and stability to achieve a high level of CMM [24]. Metrics were identified from the SW-CMM and CMMI key practices using a Goal-Question-Metric (GQM) approach [23]. An example of creating a metric using GQM is as follows:

- Goal: Project Management, i.e., plan, estimate, monitor, and control project quality.
- Sub-Goal: Improve customer satisfaction by reducing defects.
- Question: Where are defects introduced and removed?
- Metric: Defects detected in peer reviews and testing.

Integrated Metrics also has the limitation that data collection for metrics to indicate performance of each CMM and CMMI key practice is difficult.

2. Simplified assessment method

The Modular Mini-Assessment Method described by Wiegers and Sturzenberger [29] is a modified CMM assessment, which is flexible with multiple options available for most assessment steps. The time required for the assessment varied by the options chosen. For example, the questionnaire used in this method has 3 options: 1) Practices, sub-practices, some institutionalization factors. 2) All CMM key practices. 3) Institutionalization factors only. The assessment steps consist of opening a meeting, CMM orientation, questionnaire administration, participant discussion, findings generation, and findings presentation. The advantage of this method is that it is possible that many different combinations of assessment options are chosen to create custom approaches. But it cannot yield an official CMM maturity level rating because this modified CMM assessment does not comply with the CMM-based appraisal framework [20]. It is concerned more with identifying appropriated improvement opportunities.

The Self-Assessment Method by Frey-Pucko, et al is a low-budget assessment developed to identify possibilities for improving a complex development process, and it is focused on questionnaire and rating improvement. It modifies small-scale appraisals, which allows assessments to be performed more frequently. To make it simple and accurate, the original appraisal process is simplified from multiple activities to seven major activities (appraisal planning, familiarization meeting, selection of respondents, interviewing, data analysis, presentation of findings, progress measurement), applied to multiple choice answers instead of "Yes/No" type answers, and respondent weight is assigned based on respondent's understanding. But it still requires a certain budget even though it is reduced to a small-scale assessment [30].

3. CMM assessment using expert system

Karami and Garratt suggested an expert system that assists CMM assessment [25]. Experts' knowledge is stored into the system, and the system asks questions such as whether there is a required training program for all newly-appointed development managers designed to familiarize them with software project management. And the respondent should answer correctly. Based on these answers, the system shows the maturity level and finds out what you should do to improve the process level; for example, communicate the changes that affect the software projects with other groups such as SCM, SQA.

But to match the answers to the questions can be time-consuming and sometimes the answers are not correct. It would be better if the question was answered automatically. It is possible that rules and facts, which can be generated from experts' knowledge and gathered from software development object and attributes, automatically provide correct answers.

E. ISO 9000 and Software Life Cycle Standards

1. ISO 9000

ISO made a model similar to the CMM. Ghosh compared ISO 9000 with the CMM [31]. ISO 9000 focuses on establishing a basic framework for quality systems, which provides standards for capability determination of a software engineering organization. The CMM is more detailed and provides a maturity framework for software engineering processes, which aims at process assessment and process improvement.

2. Rational Unified Process

RUP (Rational Unified Process) is a Software Engineering Process model from Rational. It consists of Software engineering practices and has a life cycle consisting of four sequential phases: Inception, Elaboration, Construction, and Transition. It identifies roles, activities, artifacts, and workflows, and uses an iterative procedure to support many key practices of CMM. Roles define the behavior and responsibilities of individuals in the Project. An activity is something that a role does that provides a meaningful result in the context of the project. The artifacts produced through the RUP are mostly documents and measurements which are related to requirements, analysis and design, implementation, test, deployment, configuration and change management, project management, environment, and business modeling. A workflow is a sequence of activities. Rational shows the way to achieve Capability Maturity Model levels 2 and 3 with RUP [28]. RUP can be applied to the PAMPA 2.0 as a Software Life Cycle and

Artifacts and **Activit(y)**ies in RUP will be gathered as attributes in the PAMPA 2.0 knowledge base.

3. ISO/IEC 12207

ISO 12207 describes a high-level standard addressing all processes of the software life cycle. It has five primary processes, eight supporting processes, and seven organizational life cycle processes, which describe acquiring, developing, supplying, operating, and maintaining software. Compared to RUP, which is focused on software development, ISO 12207 defines activities that should take place but does not prescribe how they should be accomplished. It is more focused on acquisition and supply of software than development of software [33].

4. IEEE/EIA 12207

ISO/IEC 12207 was published as IEEE/EIA 12207, adding the implementation of process and data for defense, commercial, and international acquisitions. Ferguson and Sheard compared the CMM with IEEE/EIA 12207 [32]. The IEEE/EIA 12207 covers a full software product life cycle with no levels, whereas the CMM focuses on Software development with five levels, that include the KPAs, goals, and common features.

F. Create an Intelligent Agent to Assess CMM Objectively

The five-level CMM is based on 18 KPAs. Each KPA is organized into five sections called common features. They are Ability to Perform, Commitment to Perform, Activities Performed, Verifying Implementation, and Measurement and Analysis. The common features contain the key practices that, when collectively addressed, accomplish the goals of the KPA [6]. Each KPA has 2 to 4 goals, which are broken down into Capabilities, **Activit(y)**ies, and Deliverables. Key Practices and PAMPA 2.0 objects and attributes are common in Capabilities, **Activit(y)**ies, and Deliverables. This enables the PAMPA 2.0 knowledge base to assess key practices [Table 4].

The PAMPA 2.0 knowledge base is used to predict whether the KPA Goals have been satisfied. Knowledge in the form of rules, objects, attributes, relationships, and properties are acquired from experts to assess each of 18 KPAs. For example, KPA 1, one of the 18 KPAs, is Requirements Management. One question, "Are the people in the project who are charged with managing the allocated requirements trained in the procedures for managing allocated requirements?" is selected to assess the Goal in KPA 1. This question is about training, which is in the category of Capabilities. The PAMPA 2.0 objects such as **Organization, Individual**, and the attributes about training experience are applied to the Inference Engine to assess the Goal achievement.

Table 4. PAMPA 2.0 Knowledge Base and Key Practices

PAMPA 2.0 Knowledge Base		**Key Practices**	
Object	**Attribute**	**Common Features**	**Main Contents**
Individual, Organization	Total cost, Number of People, Individual cost (direct labor expense, overhead expense, travel expense, computer use cost), Experience, Training experience, Resource, Responsibility, WorkBreakDownStructure	Ability to Perform	Resources, organizational structures, and training
		Commitment to Perform	Establishing organizational policies and leadership
C A P A B I L I T I E S			
Plan, Supplier, Software Product, Customer, Organization	All the attributes in PAMPA 2.0 knowledge base	Measurement and Analysis	Basic measurement practices that are necessary to determine status related to the process
D E L I V E R A B L E S			
Plan, Process, Activity	Volume planned, Reliability planned, Usability planned, Salary Average planned, Number of People Planned, Time planned, Time (Initial Milestone, Final Milestone), Risk, Kind of process, Activities (performing the work, tracking, taking corrective action, reviewing, auditing)	Activities Performed	Establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary
		Verifying Implementation	Reviews and audits by management and software quality assurance
A C T I V I T I E S			

G. Measuring CMM Level

A subjective CMM assessment procedure includes administering the maturity questionnaire and examining the questionnaire results. After that, the expert should examine process and practice documents, conduct on-site interviews, and consolidate information. But objective assessment only involves assessing the questionnaire based on the rules and facts already stored in the knowledge base system. Questionnaires cover key practices of the CMM. Figure 7 shows the relationships of CMM Questionnaire, Goal, KPA, and CMM.

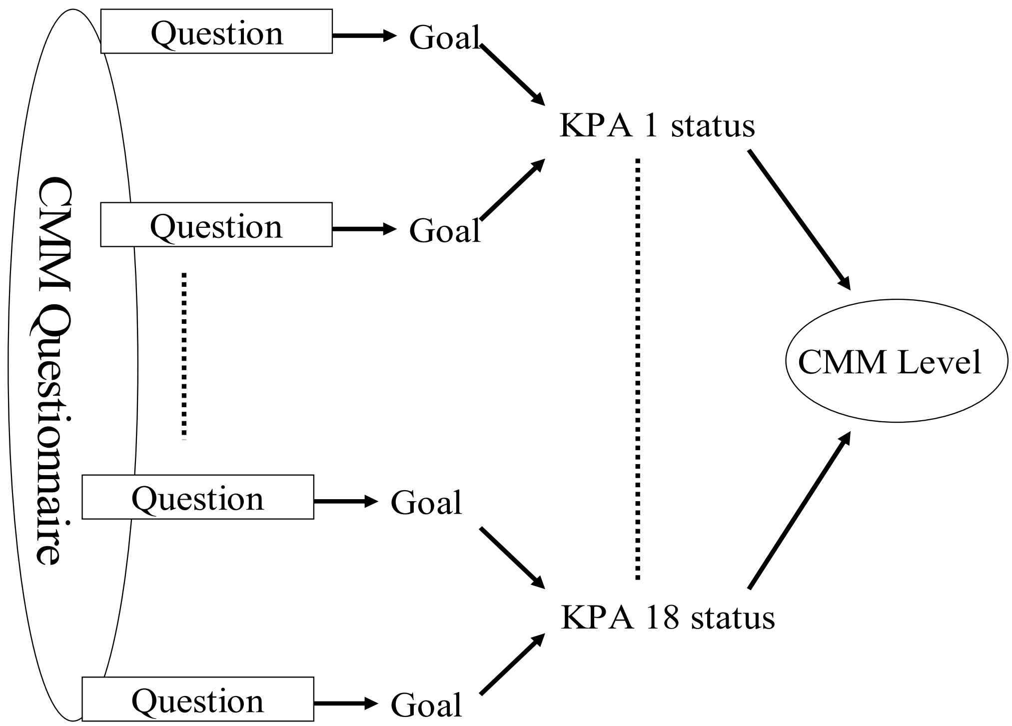


Figure 7. Relationships of CMM Questionnaire, Goal, KPA, and CMM

The questions on the CMM Questionnaire are related to the KPA Goals. Questions are answered by PAMPA 2.0 based on the facts and rules stored in the knowledge base. The KPA Status reflects the achievement of KPA Goals. We can find the CMM level from 1 to 5 based on the KPA Status.

1. Questionnaire

The grading of a subjective assessment is “Yes”, “No”, “Does not apply”, or “Do not know” choices used in the SEI’s maturity questionnaire [34]. It would be more accurate if the grading was based on percentage. The Inference Engine processes each question, and the result is represented as *Score* ranging from 0 % to 100 %. It is possible to change *Score* to five rating levels such as: almost always (over 90%), frequently (60 ~ 90%), about half (40 ~ 60%), occasionally (10 ~ 40%), and rarely if ever (less than 10%). Some questions cannot be mapped to the PAMPA 2.0 object and attribute. But once a question is mapped to the PAMPA 2.0 object and attribute, there is no answer such as "Does not apply" or "Do not know." "Does not apply" means the respondent has the required knowledge about the project or organization and the KPA but feels that the KPA does not apply to his or her circumstances. "Do not know" means the respondent is uncertain about how to respond.

2. KPA goal

The SEI's maturity questionnaire has 6~7 questions in each KPA to analyze the achievement of its KPA Goals [34]. Those questions in each KPA are related to commitment, ability, measurement, verification, and the KPA Goals. Commitment, Ability, Measurement, and Verification key practices help to achieve the KPA Goal. *The Capability Maturity Model* shows how each of the Commitment, Ability, Measurement, and Verification key practices maps to its associated Goals [15]. If the average *Score* from questions of Commitment, Ability, Measurement, and Verification key practices are above 50%, we know it provides significant support for KPA Goal achievement and satisfies the prerequisite for measuring KPA Goal achievement.

3. KPA status

The *Score* of each KPA Goal-related question is *Goal Question Score* and the sum of the *Goal Question Score* divided by the number of goals in each KPA represents the status of its KPA goal achievement. $KPAWeight_i = (\sum_{j=1}^{n} GoalQuestionScore_j)\frac{1}{n}$

4. CMM level

CMM levels are decomposed into several KPAs. CMM level 2 consists of KPA 1 ~ 6, CMM level 3 consists of KPA 7 ~ 14, CMM level 4 consists of KPA 15 ~ 16, and CMM level 5 consists of KPA 17 ~ 18. The higher level of the CMM can be achieved after satisfying all the lower levels of the CMM. For example, if your organization is at CMM level 3, most of the goals in KPA 1 ~ 14 have been achieved. The sum of the *KPAWeight* divided by the number of KPAs in each CMM level measures the achievement of each CMM level. It can be represented as $CMMlevel_2 = (\sum_{j=1}^{6} KPAWeight_j)\frac{1}{6}$,

$CMMlevel_3 = (\sum_{j=7}^{14} KPAWeight_j)\frac{1}{8}$, $CMMlevel_4 = (\sum_{j=15}^{16} KPAWeight_j)\frac{1}{2}$,

$CMMlevel_5 = (\sum_{j=17}^{18} KPAWeight_j)\frac{1}{2}$. There would be *CMMlevel* results for CMM levels 2, 3, 4, and 5. These results will be stored as fact in the PAMPA 2.0 knowledge base. If there are facts that show that CMM level 2 and CMM level 3 are satisfactory, and CMM level 4 and CMM level 5 are not satisfactory, then the project is in CMM level 3.

3. KPA ASSESSMENT

A. KPA Assessment Environment

In this research, I assessed the process maturity of *Jinhwan's Project* and *CPSC606 Project*. The significant difference in these projects is that *Jinhwan's Project* chose Software Life Cycle as Rational Unified Process (RUP), whereas *CPSC606 Project* chose simple Software Life Cycle. Both projects were implemented by support of Rational tools and MS Project. PAMPA 2.0 gathers attributes such as requirements management attributes from RequisitePro, configuration management attributes from ClearCase, **Activity** and defect tracking attributes from ClearQuest, testing attributes from Test Robot, and project planning related attributes from MS Project.

In this implementation the **Activit(y)**ies and **Artifact**s are predefined as related **Activit(y)**ies and documents in Requirements Management (RM), Software Project Planning (SPP), Software Project Tracking and Oversight (SPTO), Software Subcontract Management (SSM), Software Quality Assurance (SQA), Software Configuration Management (SCM), Organization Process Focus (OPF), Organization Process Definition (OPD), Training Program (TP), Integrated Software Management (ISM), Software Product Engineering (SPE), Intergroup Coordination (IC), Peer Reviews (PR), Quantitative Process Management (QPM), Software Quality Management (SQM), Defect Prevention (DP), Technology Change Management (TCM), and Process Change Management (PCM).

Appendix A shows how PAMPA 2.0 implemented objective CMM assessment to these projects. It describes the data available related to the CMM analysis, the heuristics used to make an assessment based on that data, an argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation, and the limitations of the data and heuristics.

B. Mapping the Questionnaire to KPA Goals and Practices

We use *The CMM Maturity Questionnaire* [34, Appendix C] and elicit the expert's knowledge from *The Capability Maturity Model* [15], which explains key practices in detail, to assess CMM objectively.

Table 5 shows the mapping relationship between the Questionnaire and the KPA goals, commitments, abilities, activities, measurements, and verification from reference [15]. In *CMM Matuirity Questionnaire*, KPA 1.1 is the first question under KPA 1, Requirement Management. Commitment, Ability, Activity, Measurement, and Verification are five sections in Common Features which contain the Key Practices, and mapping the Key Practices to Goals in Table 5 is adapted from *The Capability Maturity Model* [15]. For example, question KPA 1.1 maps to Goal 1, Ability 1,2,3, Activity 1, and Verification 1,2 of KPA 1.

Table 5. Mapping the Questionnaire to KPA Goals and Practices [15, Appendix C]

Questionnaire	Goal	Commitment	Ability	Activity	Measurement	Verification
KPA 1.1	1		1,2,3	1		1,2
KPA 1.2	2		3	2,3		1,2

Table 5 continued.

Questionnaire	Goal	Commitment	Ability	Activity	Measurement	Verification
KPA 1.3		1				
KPA 1.4			4			
KPA 1.5					1	
KPA 1.6						3
KPA 2.1	1	1	1,4	9,10,11, 12,15		1,2,3
KPA 2.2	2	1	1,2,4	2,5,6,7,8, 13,14		1,3
KPA 2.3	3	1	1,4	1,3,4		1,3
KPA 2.4		2				
KPA 2.5			3			
KPA 2.6						
KPA 2.7						2
KPA 3.1	1	1	1,3,4,5	1,5,6,7,8, 9,10,11,12, 13		2,3
KPA 3.2	2	1	1,3,4,5	2,5,6,7,8, 9,11		2,3
KPA 3.3	3	1	1,3,4,5	3,4		2,3
KPA 3.4		2				
KPA 3.5			2			
KPA 3.6					1	
KPA 3.7						1
KPA 4.1	1	2	1			3
KPA 4.2	2	2	1,3	3,4,6		1,3
KPA 4.3				8		
KPA 4.4	4	2	1,3	3,5,7,9,10, 11,12,13		1,3
KPA 4.5		1				
KPA 4.6			2			
KPA 4.7					1	
KPA 4.8						2
KPA 5.1	1		1,3	1,2		2,3
KPA 5.2	2		1,3,4	2,3,4,5		2,3
KPA 5.3	3		1,3,4	6,7,8		2,3
KPA 5.4	4		1,3,4	7		2,3
KPA 5.5		1				
KPA 5.6			2			
KPA 5.7					1	
KPA 5.8						1

Table 5 continued.

Questionnaire	Goal	Commitment	Ability	Activity	Measurement	Verification
KPA 6.1	1		2,3,4	1,2		2,4
KPA 6.2	2		1,2,3,4	2,3,4,7		4
KPA 6.3	3		1,2,3,4	5,6		4
KPA 6.4	4		2,3,4	8,9,10		1,2,4
KPA 6.5		1				
KPA 6.6			5			
KPA 6.7					1	
KPA 6.8						3
KPA 7.1	1	3	2,3,4	3,4,5,6,7		
KPA 7.2	2	3	2,3,4	1		
KPA 7.3		1				
KPA 7.4		2				
KPA 7.5			1			
KPA 7.6					1	
KPA 7.7						1
KPA 8.1	1		1	1,2,3,4		
KPA 8.2	2		1	5,6		
KPA 8.3		1				
KPA 8.4			2			
KPA 8.5					1	
KPA 8.6						1
KPA 9.1	1		1,3,4	1,2,3	1	3
KPA 9.2	2		1,3,4	3,4	1	2,3
KPA 9.3	3		1,3,4	5,6	1	2,3
KPA 9.4		1				
KPA 9.5			2			
KPA 9.6					2	
KPA 9.7						1
KPA 10.1	1		1	1,2,3		2
KPA 10.2	2		1,3	3,4,5,6,7, 8,9,10,11		1,2
KPA 10.3		1				
KPA 10.4			2			
KPA 10.5					1	
KPA 10.6						3
KPA 11.1	1		2,3,4	1,2,3,4, 5,6,7,8,9	2	1,2
KPA 11.2	2		2,3,4	10	2	1,2
KPA 11.3		1				

Table 5 continued.

Questionnaire	Goal	Commitment	Ability	Activity	Measurement	Verification
KPA 11.4			1			
KPA 11.5					1	
KPA 11.6						3
KPA 12.1	1		1,3,4,5	1		3
KPA 12.2	2		1,3,4,5	3,4,5		3
KPA 12.3	3		1,3,4,5	2,6,7		1,3
KPA 12.4		1				
KPA 12.5			2			
KPA 12.6					1	
KPA 12.7						2
KPA 13.1	1		1	1		
KPA 13.2	2		1,3	2,3		
KPA 13.3		1				
KPA 13.4			2			
KPA 13.5					1	
KPA 13.6						1
KPA 14.1	1	2	1,3,4,5	1,2,3		3
KPA 14.2	2		1,3,4,5	2,4,5,6		1,3
KPA 14.3	3	2	1,3,4,5	7		1,3
KPA 14.4		1				
KPA 14.5			2			
KPA 14.6					1	
KPA 14.7						2
KPA 15.1	1		1,2	1,2		2,3
KPA 15.2	2		1,2	3,5		2,3
KPA 15.3	3		1,2	2,4		2,3
KPA 15.4		1				
KPA 15.5			3			
KPA 15.6					1	
KPA 15.7						1
KPA 16.1	1	2	1,2,3	1,2		2
KPA 16.2	2	2	3	3,5		
KPA 16.3	3	2	1,2,3	4,6,7,8		1,2
KPA 16.4		1				
KPA 16.5			4			
KPA 16.6					1	
KPA 16.7						3
KPA 17.1	1	3	1,2,5	1		2
KPA 17.2	2	3	1,2,3,5	2,4,5,6		2

Table 5 continued.

Questionnaire	Goal	Commitment	Ability	Activity	Measurement	Verification
KPA 17.3		1				
KPA 17.4		2				
KPA 17.5			4			
KPA 17.6					1	
KPA 17.7						1
KPA 18.1				3		
KPA 18.2				6		
KPA 18.3	3	2	1,2,3,4	4,5,7,8,9		1,2
KPA 18.4		1				
KPA 18.5				3		
KPA 18.6					1	
KPA 18.7						7

C. Objective Assessment Limitation

Although we are trying to objectively assess all the key practices of the CMM, there are limitations in converting from subjective data to objective data. The reliability of objective CMM assessment depends on how much subjective data can be converted to objective data. Table 6 shows the status of subjective and objective data for each KPA. In this table, objective rate is represented as percentage based on the number of activities, which are difficult to figure out as objective activity from the total number of activities. For example, Questionnaire KPA 8.3 has objective rate 50% because there is one activity, which is difficult to be objective activity, out of 2 activities in Questionnaire KPA 8.3.

Table 6. Subjective & Objective Activity Status

Questionnaire	Subjective Activities difficult to be Objective Activity	Objective Rate (%)
KPA 1.1		100
KPA 1.2		100
KPA 1.3		100
KPA 1.4		100
KPA 1.5		100
KPA 1.6		100
KPA 2.1		100
KPA 2.2		100
KPA 2.3		100
KPA 2.4		100
KPA 2.5		100
KPA 2.6		100
KPA 2.7		100
KPA 3.1		100
KPA 3.2		100
KPA 3.3		100
KPA 3.4		100
KPA 3.5		100
KPA 3.6		100
KPA 3.7		100
KPA 4.1		100
KPA 4.2		100
KPA 4.3		100
KPA 4.4		100
KPA 4.5		100
KPA 4.6		100
KPA 4.7		100
KPA 4.8		100
KPA 5.1		100
KPA 5.2		100
KPA 5.3		100
KPA 5.4		100
KPA 5.5		100
KPA 5.6		100
KPA 5.7		100
KPA 5.8		100
KPA 6.1		100
KPA 6.2		100

Table 6 continued.

Questionnaire	Subjective Activities difficult to be Objective Activity	Objective Rate (%)
KPA 6.3	Changes to baselines are controlled according to a documented procedure.	50
KPA 6.4		100
KPA 6.5		100
KPA 6.6		100
KPA 6.7		100
KPA 6.8		100
KPA 7.1	Activity 3. The organization's and projects' activities for developing and improving their software processes are coordinated at the organization level. Activity 4. The use of the organization's software process database is coordinated at the organizational level.	60
KPA 7.2		100
KPA 7.3	Commitment 1. The software processes used by the projects are assessed periodically to determine their strengths and weaknesses. The software processes used by the projects are appropriately tailored from the organization's standard software process.	50
KPA 7.4		100
KPA 7.5		100
KPA 7.6		100
KPA 7.7		100
KPA 8.1	Activity 1. The organizations standard software process is developed and maintained according to a documented procedure. Activity 2. The organization's standard software process is documented according to established organization standards. Activity 4. Guidelines and criteria for the projects' tailoring of the organization's standard software process are developed and maintained.	25
KPA 8.2		100
KPA 8.3	Commitment 1. A standard software process is defined for the organization. A project's defined software process is a tailored version of the organization's standard software process.	50
KPA 8.4		100
KPA 8.5		100
KPA 8.6		100
KPA 9.1		100
KPA 9.2		100
KPA 9.3		100
KPA 9.4		100
KPA 9.5		100
KPA 9.6		100

Table 6 continued.

Questionnaire	Subjective Activities difficult to be Objective Activity	Objective Rate (%)
KPA 9.7		100
KPA 10.1	Activity 1. The project's defined software process is developed by tailoring the organization's standard software process according to a documented procedure.	66
KPA 10.2	Activity 11. Reviews of the software project are periodically performed to determine the actions needed to bring the software project's performance and results in line with the current and projected needs of the business, customer, and end users, as appropriate.	90
KPA 10.3	Commitment 1. Each project documents the project's defined software process by tailoring the organization's standard software process.	75
KPA 10.4		100
KPA 10.5		100
KPA 10.6		100
KPA 11.1	Activity 1. Configuration management models appropriate to the software project are selected and used. Activity 3. The software detailed designing is developed based on the software architecture. Activity 5. The adequacy of testing is determined based on the test coverage to be achieved.	90
KPA 11.2	Activity 10. As understanding of the software improves, changes to the software work products, plans, process descriptions, and activities are proposed, analyzed, and incorporated as appropriate.	75
KPA 11.3		100
KPA 11.4		100
KPA 11.5		100
KPA 11.6		100
KPA 12.1		100
KPA 12.2	Activity 4. Critical dependencies are tracked on a regular basis, and corrective actions are taken when appropriate.	90
KPA 12.3		100
KPA 12.4		100
KPA 12.5		100
KPA 12.6		100
KPA 12.7		100
KPA 13.1		100
KPA 13.2	Activity 2. The successful completion of peer reviews, including the rework to address the items identified in the peer reviews, is used as a completion criterion for the associated task.	90
KPA 13.3		100

Table 6 continued.

Questionnaire	Subjective Activities difficult to be Objective Activity	Objective Rate (%)
KPA 13.4		100
KPA 13.5		100
KPA 13.6		100
KPA 14.1	Activity 3. The strategy for the data collection and the quantitative analyses to be performed are determined based upon the project's defined software process.	70
KPA 14.2		100
KPA 14.3		100
KPA 14.4		100
KPA 14.5		100
KPA 14.6		100
KPA 14.7		100
KPA 15.1		100
KPA 15.2		100
KPA 15.3		100
KPA 15.4		100
KPA 15.5		100
KPA 15.6		100
KPA 15.7		100
KPA 16.1		100
KPA 16.2	For Activity 6, 7, we can check the existence of the software process, but we cannot get the objective data about how those processes are incorporated according to a documented procedure. Activity 6: Revisions to the organization's standard software process resulting from defect prevention actions are incorporated according to a documented procedure. Activity 7: Revisions to the project's defined software process resulting from defect prevention actions are incorporated according to a documented procedure.	50
KPA 16.3		100
KPA 16.4		100
KPA 16.5		100
KPA 16.6		100
KPA 16.7		100
KPA 17.1	Activity 1. Defines the long-term technical strategy and identifies the procedures to be followed in performing the organization's technology change management activities.	60
KPA 17.2		100
KPA 17.3		100
KPA 17.4		100

Table 6 continued.

Questionnaire	Subjective Activities difficult to be Objective Activity	Objective Rate (%)
KPA 17.5		100
KPA 17.6		100
KPA 17.7		100
KPA 18.1	Activity 1. The software process improvement plan is based on the organization's business and strategic operating plans.	80
KPA 18.2		100
KPA 18.3	Activity 4. Appropriated administrative procedures are included to encourage participation in and facilitate the software process improvement activities. Activity 5. Proposals include the findings and recommendations of software process assessments, examples of software process improvement proposals, feedback on previously submitted software process improvement proposals. Activity 8. Appropriated process changes are incorporated into the organization's standard software process. Appropriated process changes are incorporated into the projects' defined software processes.	70
KPA 18.4		100
KPA 18.5		100
KPA 18.6		100
KPA 18.7		100

4. IMPLEMENTATION

A. Introduction

PAMPA 2.0 is a web-based system which allows users to access on the web. The figures from this chapter are the snapshots of PAMPA 2.0 implementation.

B. PAMPA Object & Attribute Display

1. Project list

You can choose either *Jinhwan's Project* or *CPSC606 Project*. Both projects show the project objects, attributes, CMM assessment result, and project status. *Jinhwan's Project* is implemented by Jinhwan, and *CPSC606 Project* is implemented by spring 2003 CPSC 606 students. By clicking one of the project names, you can access the project [Figure 8].

Figure 8. Project List

2. Project

After you click *Jinhwan's Project* in the **ProjectList**, you can access *Jinhwan's Project* **Project** attributes such as project name, actual start date, and planned end date [Figure 9]. From here you can choose to access object and attributes of **Plan, Supplier, Organization, Software product**, and **Customer**. Also you can select Facts to see the facts, which are generated from attributes and CMM Assessment to see the result of the CMM assessment and project status.

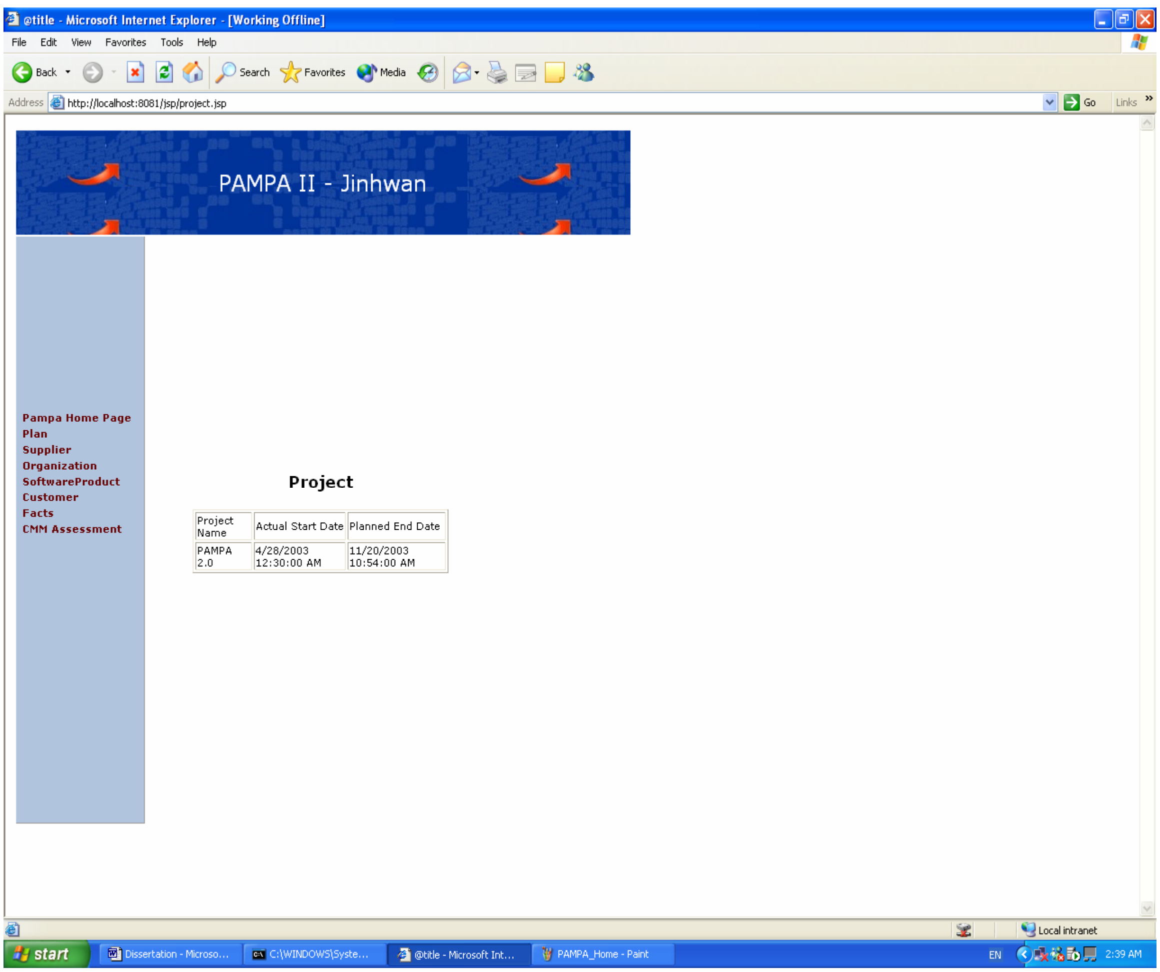
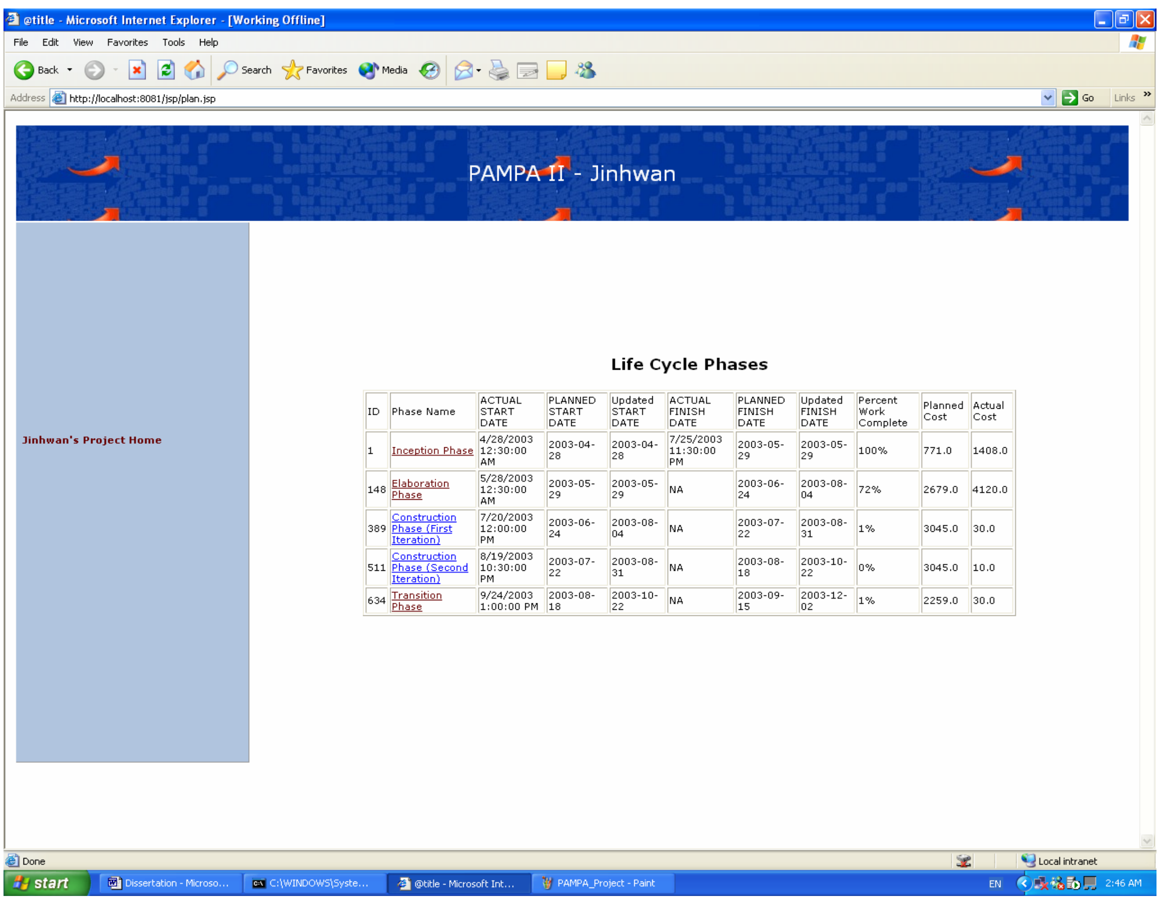


Figure 9. Project

3. Life Cycle Phases

RUP is chosen as Software Life Cycle. It has an inception phase, elaboration phase, construction phase, and transition phase [Figure 10]. You can see the attributes of each life cycle phase's actual start date, planned start date, updated start date, actual finish date, planned finish date, updated finish date, percentage of work completed, planned cost, and actual cost. Updated start and updated finish date is the modified date based on the project status after the initial planned start and planned finish date. By selecting one of the Phase Names, you can see the object **Plan** and its attributes.



ID	Phase Name	ACTUAL START DATE	PLANNED START DATE	Updated START DATE	ACTUAL FINISH DATE	PLANNED FINISH DATE	Updated FINISH DATE	Percent Work Complete	Planned Cost	Actual Cost
1	Inception Phase	4/28/2003 12:30:00 AM	2003-04-28	2003-04-28	7/25/2003 11:30:00 PM	2003-05-29	2003-05-29	100%	771.0	1408.0
148	Elaboration Phase	5/28/2003 12:30:00 AM	2003-05-29	2003-05-29	NA	2003-06-24	2003-08-04	72%	2679.0	4120.0
389	Construction Phase (First Iteration)	7/20/2003 12:00:00 PM	2003-06-24	2003-08-04	NA	2003-07-22	2003-08-31	1%	3045.0	30.0
511	Construction Phase (Second Iteration)	8/19/2003 10:30:00 PM	2003-07-22	2003-08-31	NA	2003-08-18	2003-10-22	0%	3045.0	10.0
634	Transition Phase	9/24/2003 1:00:00 PM	2003-08-18	2003-10-22	NA	2003-09-15	2003-12-02	1%	2259.0	30.0

Figure 10. Life Cycle Phases

4. Plan

After you choose one of the Phase Names in Life Cycle Phases, you can get into the object **Plan** [Figure 11]. It has attributes such as actual start date, planned start date, updated start date, actual finish date, planned finish date, updated finish date, percentage of work completed, planned cost, and actual cost. When you click one of the Plan Names, you can see the object **Process** and its attributes.

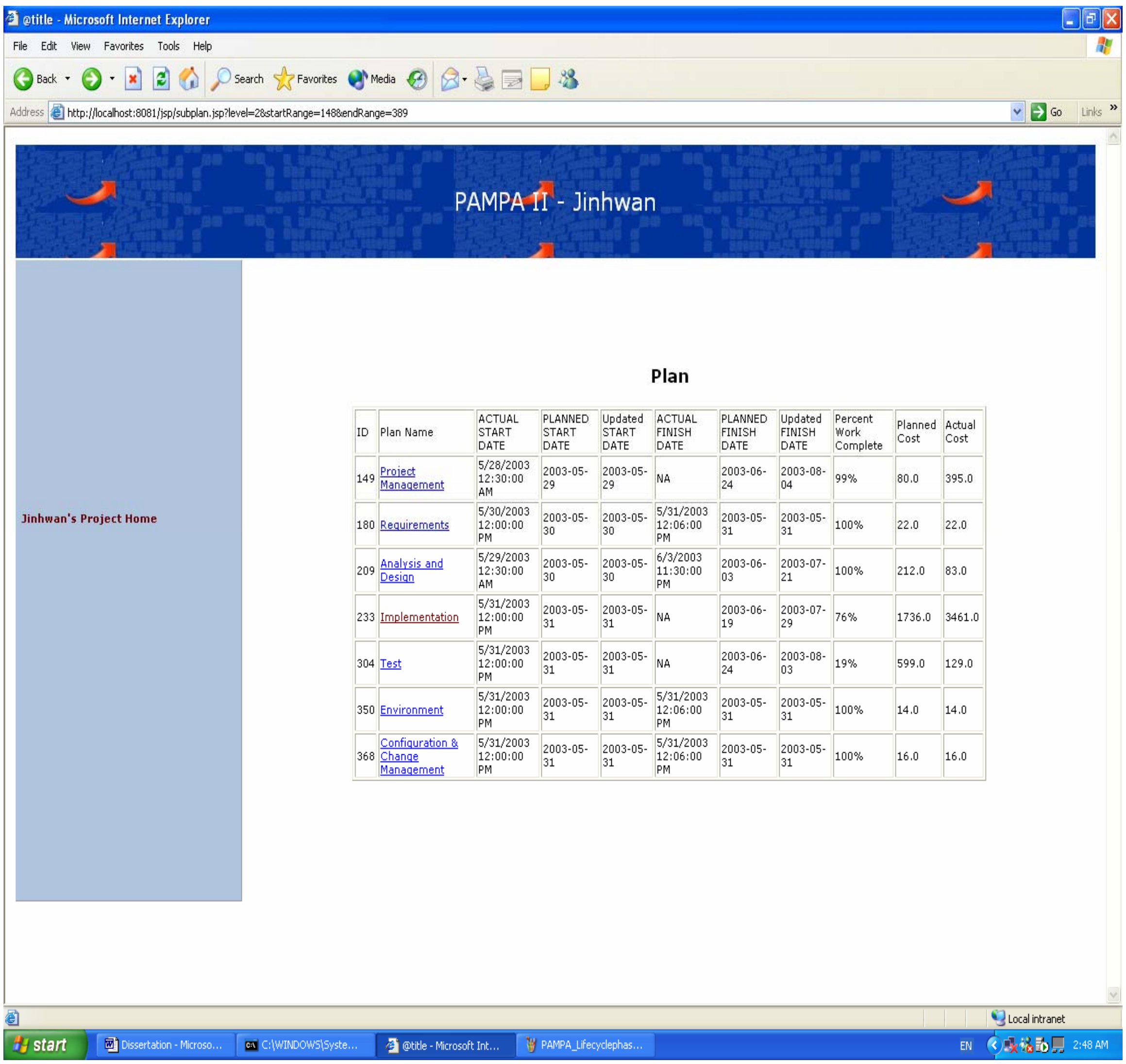


Figure 11. Plan

5. Process

After you choose one of the **Plan**s, you can see several **Process**es [Figure 12]. Each has attributes such as Process's actual start date, planned start date, updated start date, actual finish date, planned finish date, updated finish date, percentage of work completed, planned cost, and actual cost. When you click one of the Process Names, you can see the object **Activit(y)**ies and its attributes.

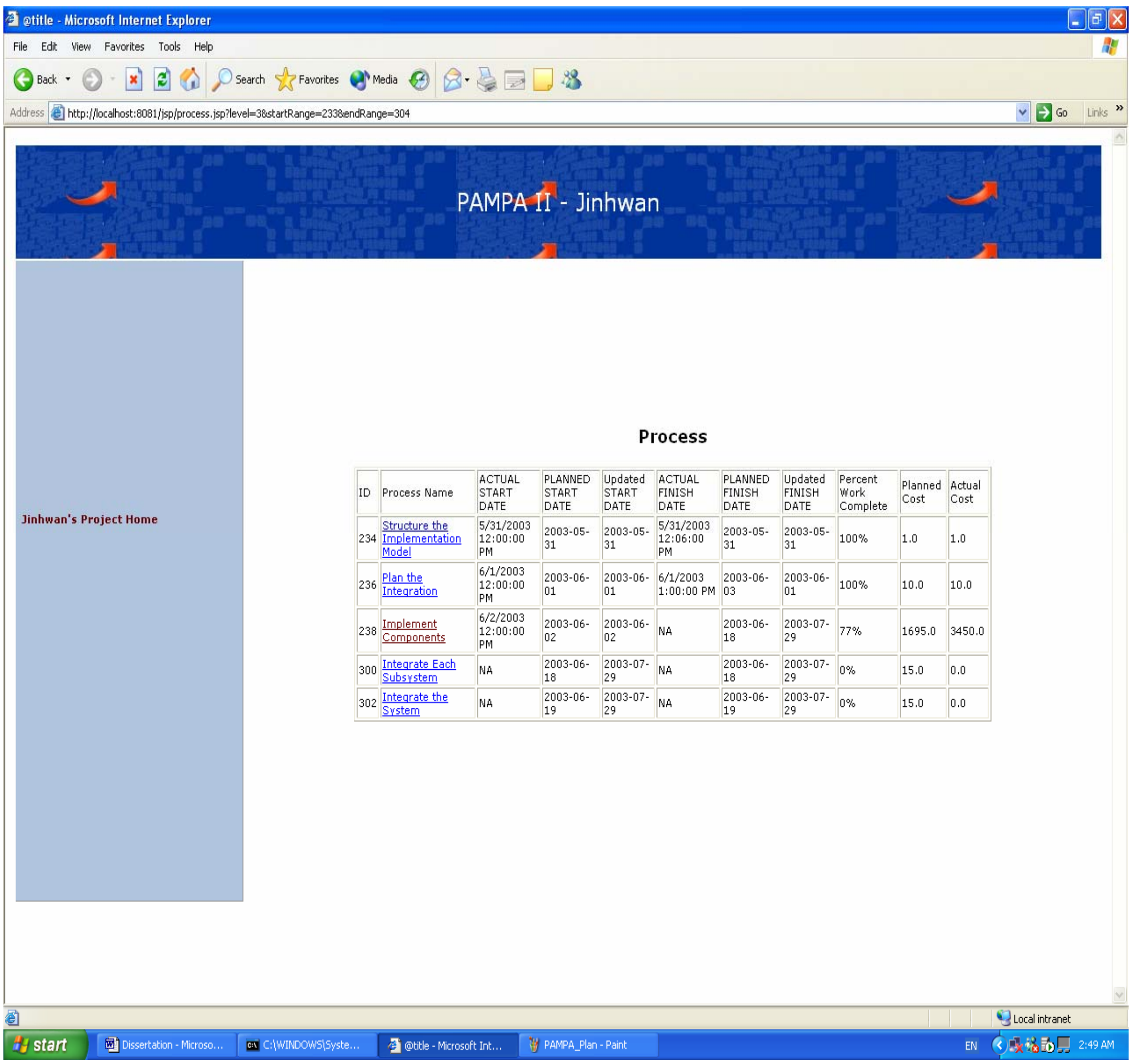
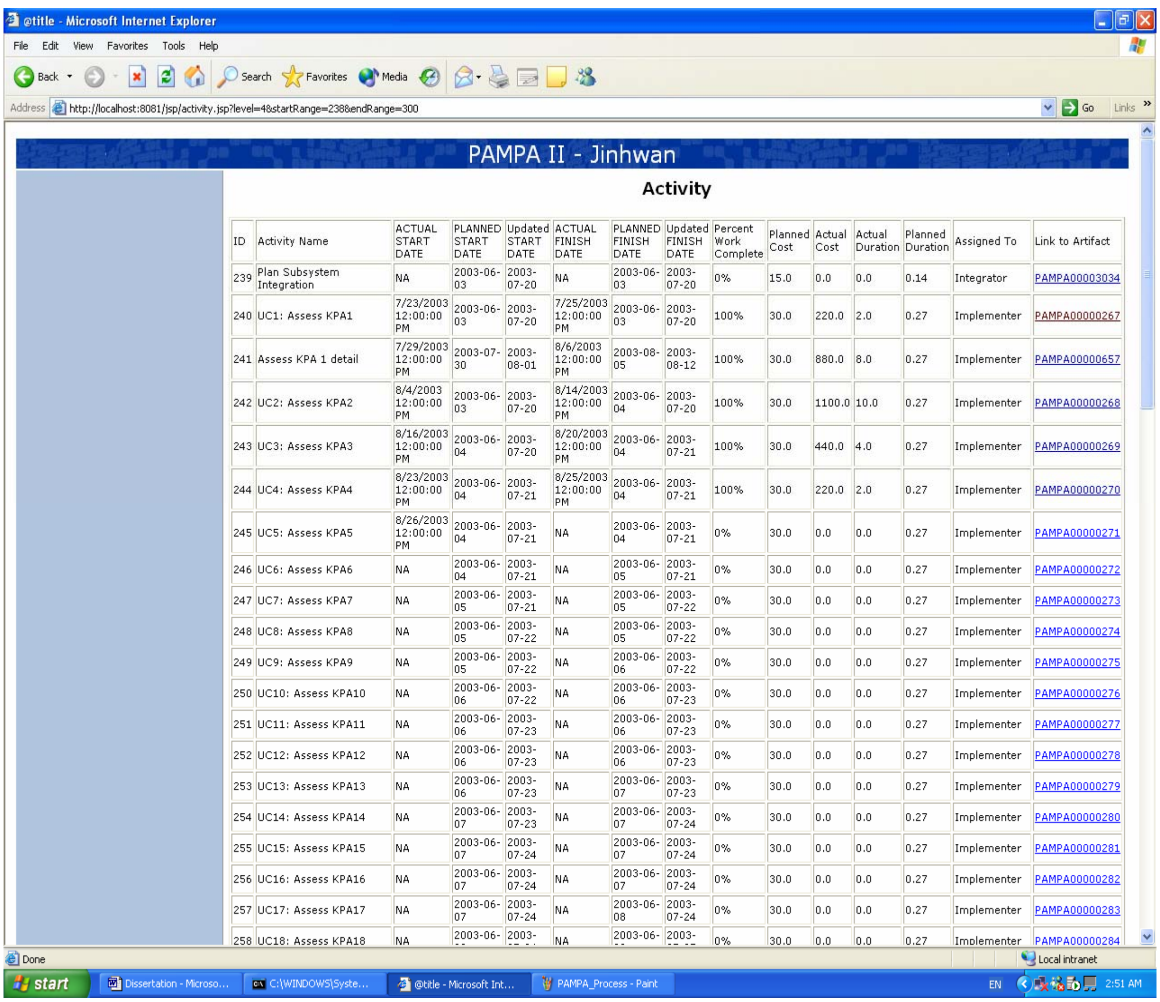


Figure 12. Process

6. Activity

After you select one of the Process Names, you can see object **Activity** and its attributes [Figure 13]. The attributes are activity name, actual start date, planned start date, updated start date, actual finish date, planned finish date, updated finish date, percentage of work completed, planned cost, actual cost, actual duration, planned duration, and assigned resource name. There is a link to object **Artifact**, which is specifically related to one of the **Activit(y)ies**.



ID	Activity Name	ACTUAL START DATE	PLANNED START DATE	Updated START DATE	ACTUAL FINISH DATE	PLANNED FINISH DATE	Updated FINISH DATE	Percent Work Complete	Planned Cost	Actual Cost	Actual Duration	Planned Duration	Assigned To	Link to Artifact
239	Plan Subsystem Integration	NA	2003-06-03	2003-07-20	NA	2003-06-03	2003-07-20	0%	15.0	0.0	0.0	0.14	Integrator	PAMPA00003034
240	UC1: Assess KPA1	7/23/2003 12:00:00 PM	2003-06-03	2003-07-20	7/25/2003 12:00:00 PM	2003-06-03	2003-07-20	100%	30.0	220.0	2.0	0.27	Implementer	PAMPA00000267
241	Assess KPA 1 detail	7/29/2003 12:00:00 PM	2003-07-30	2003-08-01	8/6/2003 12:00:00 PM	2003-08-05	2003-08-12	100%	30.0	880.0	8.0	0.27	Implementer	PAMPA00000657
242	UC2: Assess KPA2	8/4/2003 12:00:00 PM	2003-06-03	2003-07-20	8/14/2003 12:00:00 PM	2003-06-04	2003-07-20	100%	30.0	1100.0	10.0	0.27	Implementer	PAMPA00000268
243	UC3: Assess KPA3	8/16/2003 12:00:00 PM	2003-06-04	2003-07-20	8/20/2003 12:00:00 PM	2003-06-04	2003-07-21	100%	30.0	440.0	4.0	0.27	Implementer	PAMPA00000269
244	UC4: Assess KPA4	8/23/2003 12:00:00 PM	2003-06-04	2003-07-21	8/25/2003 12:00:00 PM	2003-06-04	2003-07-21	100%	30.0	220.0	2.0	0.27	Implementer	PAMPA00000270
245	UC5: Assess KPA5	8/26/2003 12:00:00 PM	2003-06-04	2003-07-21	NA	2003-06-04	2003-07-21	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000271
246	UC6: Assess KPA6	NA	2003-06-04	2003-07-21	NA	2003-06-05	2003-07-21	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000272
247	UC7: Assess KPA7	NA	2003-06-05	2003-07-21	NA	2003-06-05	2003-07-22	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000273
248	UC8: Assess KPA8	NA	2003-06-05	2003-07-22	NA	2003-06-05	2003-07-22	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000274
249	UC9: Assess KPA9	NA	2003-06-05	2003-07-22	NA	2003-06-06	2003-07-22	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000275
250	UC10: Assess KPA10	NA	2003-06-06	2003-07-22	NA	2003-06-06	2003-07-23	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000276
251	UC11: Assess KPA11	NA	2003-06-06	2003-07-23	NA	2003-06-06	2003-07-23	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000277
252	UC12: Assess KPA12	NA	2003-06-06	2003-07-23	NA	2003-06-06	2003-07-23	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000278
253	UC13: Assess KPA13	NA	2003-06-06	2003-07-23	NA	2003-06-07	2003-07-23	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000279
254	UC14: Assess KPA14	NA	2003-06-07	2003-07-23	NA	2003-06-07	2003-07-24	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000280
255	UC15: Assess KPA15	NA	2003-06-07	2003-07-24	NA	2003-06-07	2003-07-24	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000281
256	UC16: Assess KPA16	NA	2003-06-07	2003-07-24	NA	2003-06-07	2003-07-24	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000282
257	UC17: Assess KPA17	NA	2003-06-07	2003-07-24	NA	2003-06-08	2003-07-24	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000283
258	UC18: Assess KPA18	NA	2003-06-	2003-	NA	2003-06-	2003-	0%	30.0	0.0	0.0	0.27	Implementer	PAMPA00000284

Figure 13. Activity

7. Artifact

After you choose one of the Activity Names, you can see the related **Artifact** to the **Activity** [Figure 14]. The object **Artifact** has attributes such as artifact name, and size in byte. When you click the Artifact Name, you can see the working history of **Artifact** such as lines deleted, changed, and added for each **Version**.

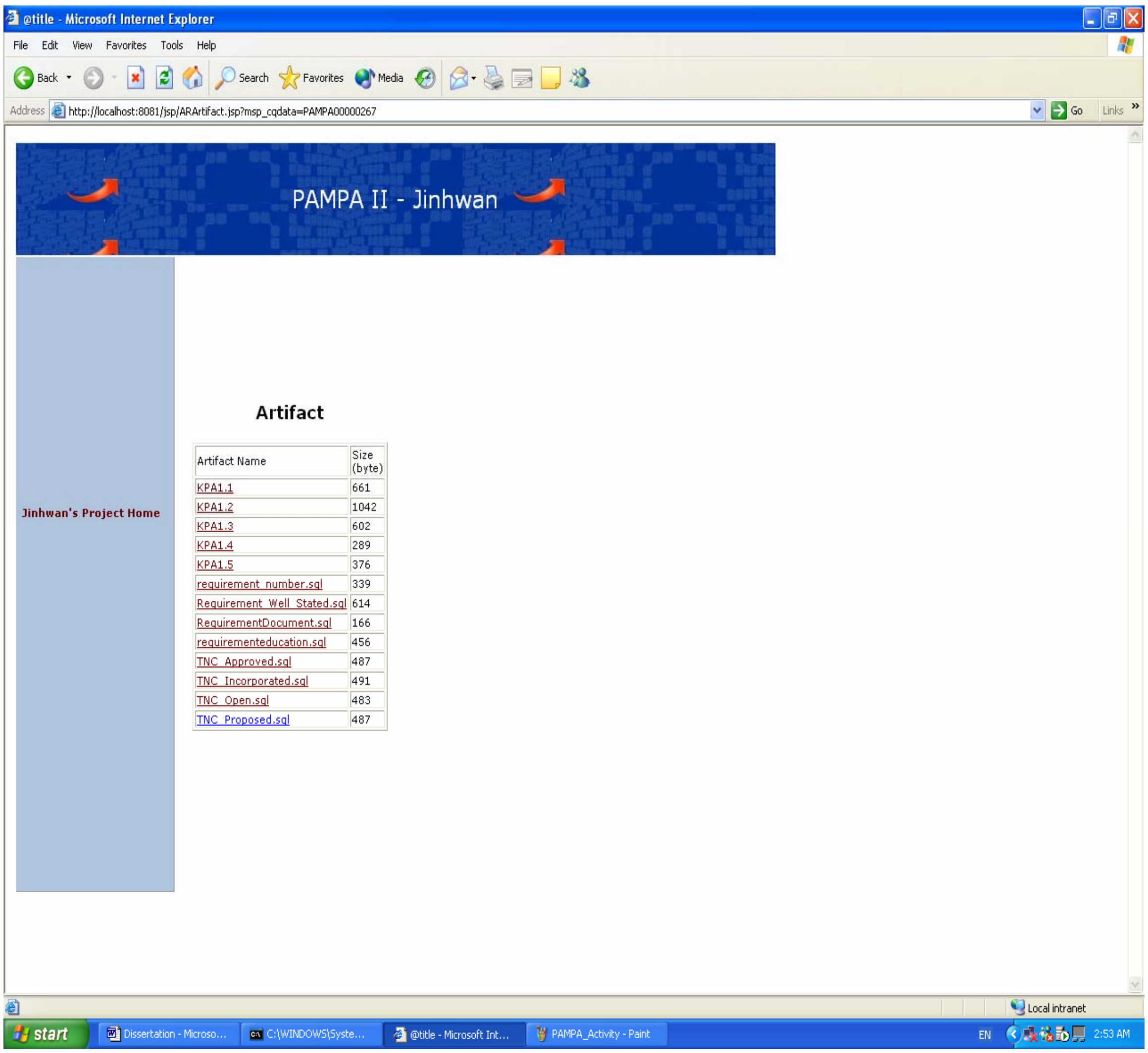


Figure 14. Artifact

8. Rework

After you click one of the Artifact Names, you can see object **Rework** and its attributes such as **Artifact** location, version, lines added, lines changed, and lines added [Figure 15]. **Artifact** location shows that **Artifact** KPA1.1 is created on directory M:\A\AC\KPA1\1, and its version is 1 with 14 lines added.

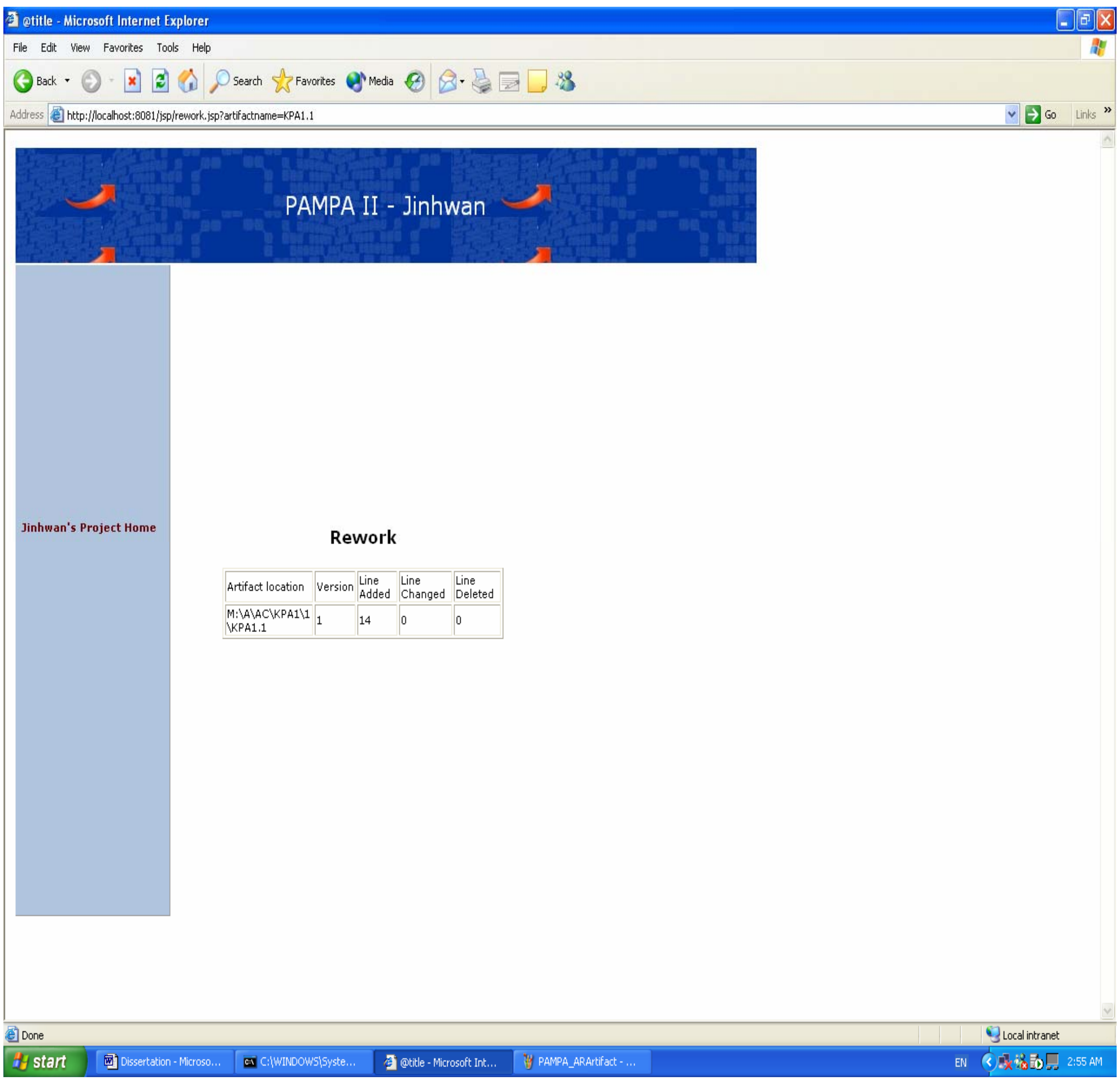


Figure 15. Rework

9. Supplier

Figure 16 shows the object **Supplier** and its attributes such as subcontractor name, technical interchange, performance ability, subcontract manager, and subcontract training experience. By selecting one of Subcontractor Names, you can see the object **ReusableSourceFile** and its attributes under the specific **Supplier**.

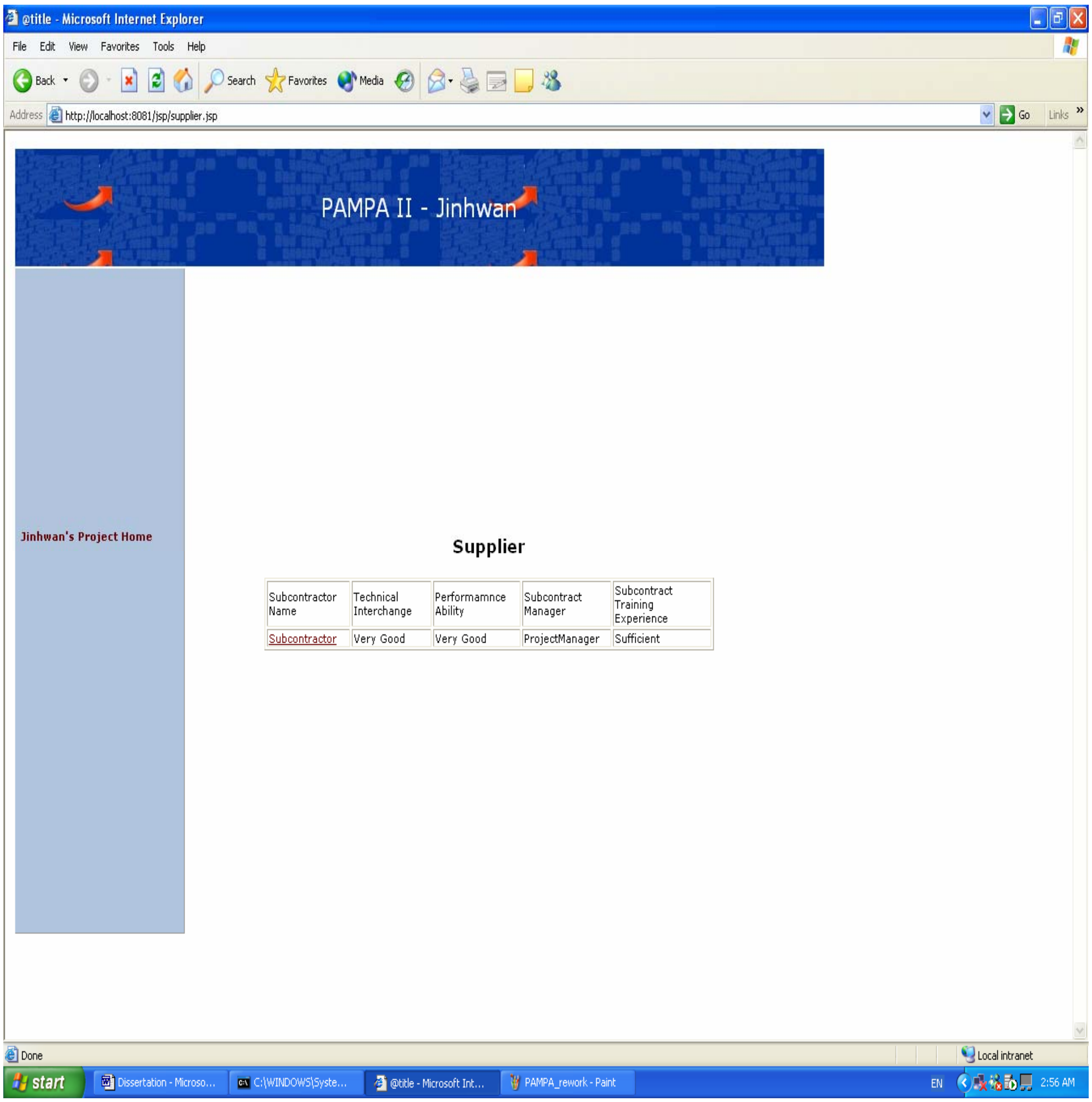


Figure 16. Supplier

10. ReusableSourceFile

ReusableSourceFile is an object under **Supplier**. By selecting the name of **Supplier** you can access to object **ReusableSourceFile** [Figure 17]. The figure below shows its attributes such as activity name, actual start date, planned start date, actual duration, planned duration, percentage of work complete, planned cost, and actual cost. There is a link to **Feature**, which shows you the **Feature** that made the **Supplier** to make the **ReusableSourceFile**.

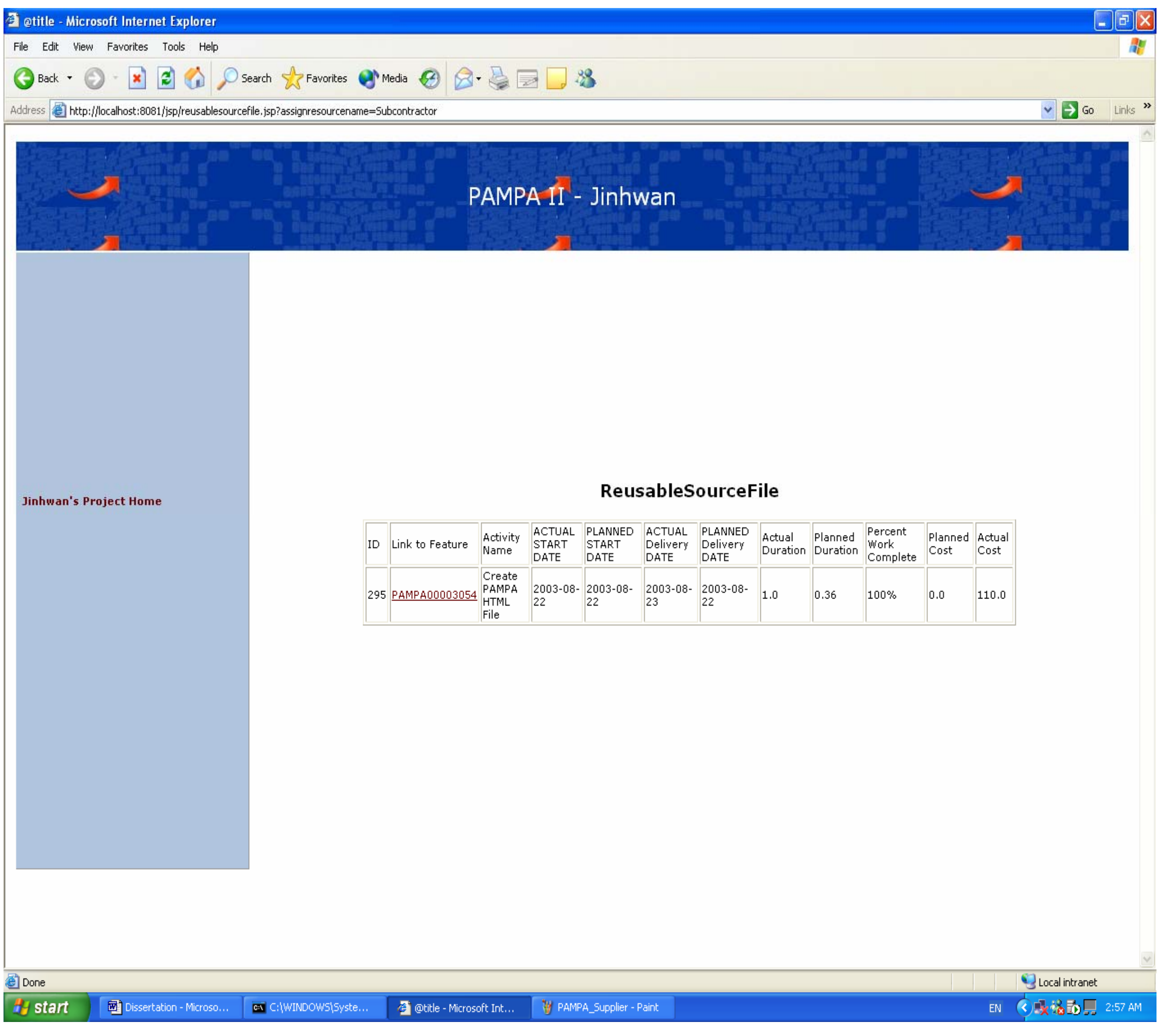


Figure 17. ReusableSourceFile

11. Feature (Supplier related)

After you click the link to **Feature** from **ReusableSourceFile**, you see the specific **Feature** related to the **ReusableSourceFile** [Figure 18]. The **Feature** has attributes such as name, status (proposed, approved, incorporated, validated), assigned resource name, and the description of the **Feature**.

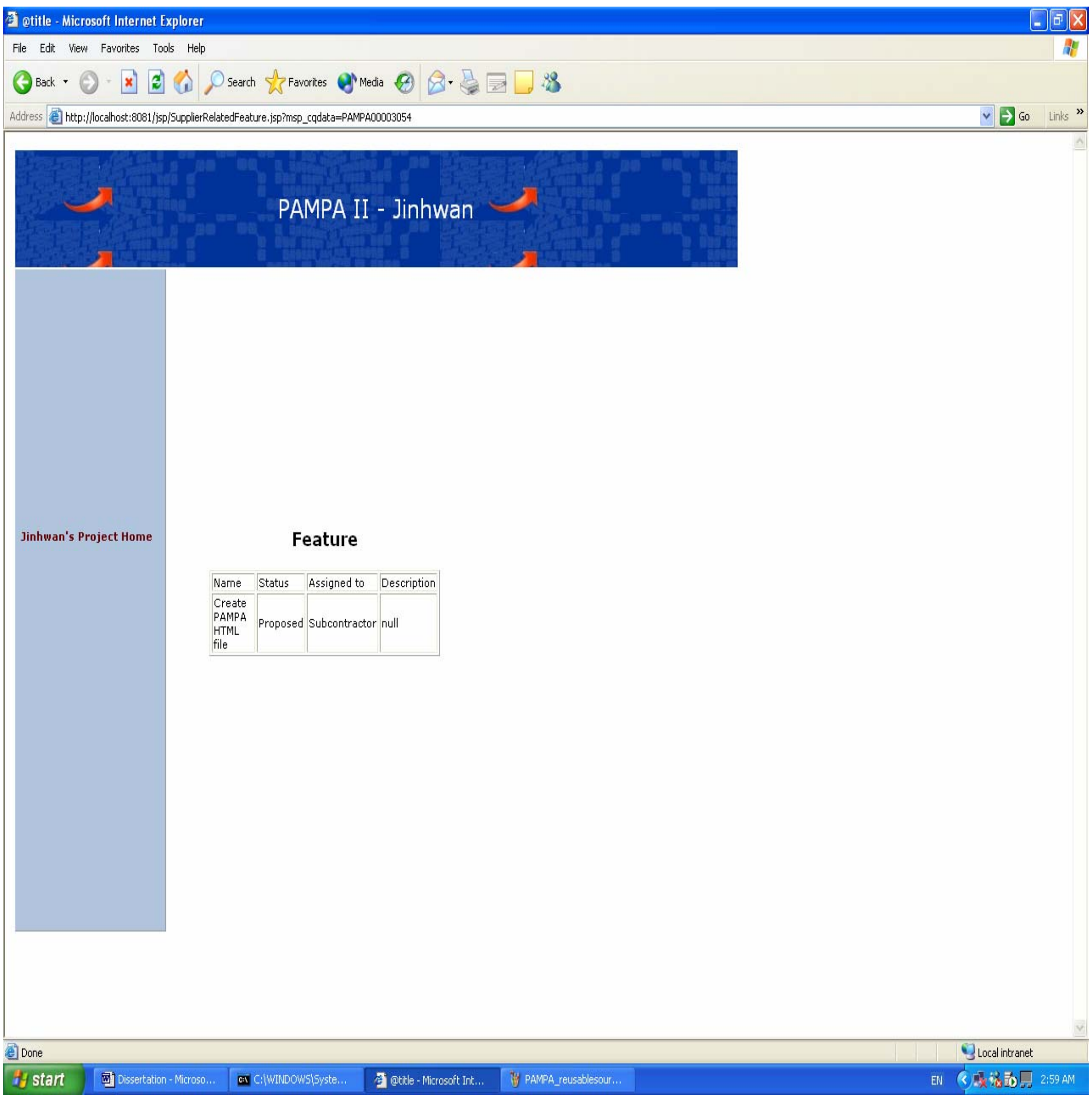


Figure 18. Feature (Supplier related)

12. Organization

Figure 19 shows the object **Organization**. By selecting the name of **Organization**, you can see object **Individual**.

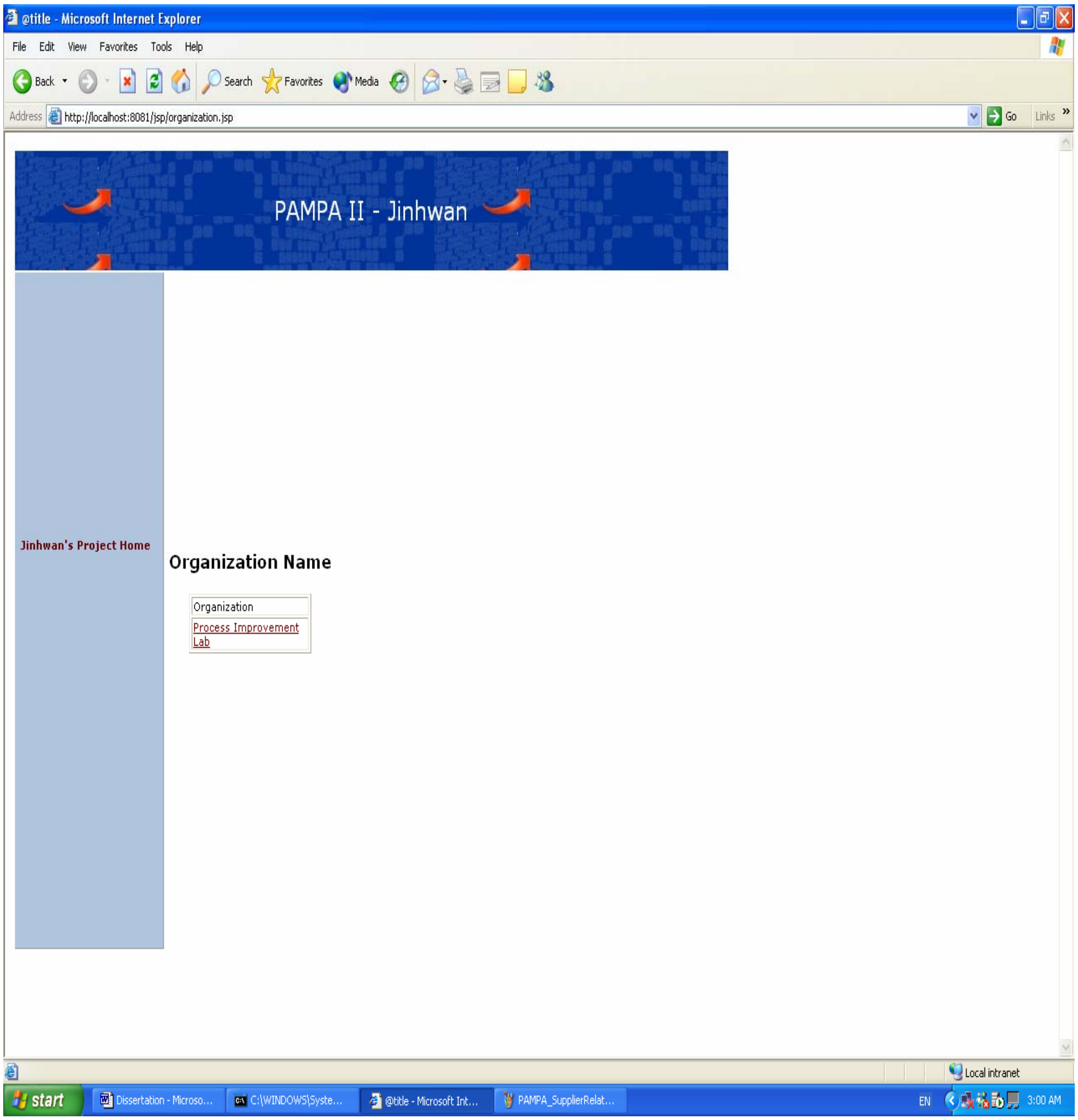


Figure 19. Organization

13. Individual

After you click the name of **Organization** you can see the object **Individual** and its attributes such as name and training experience (sufficient, insufficient) [Figure 20]. If you click name you can see object **WorkBreakDownStructure** under the specific **Individual**.

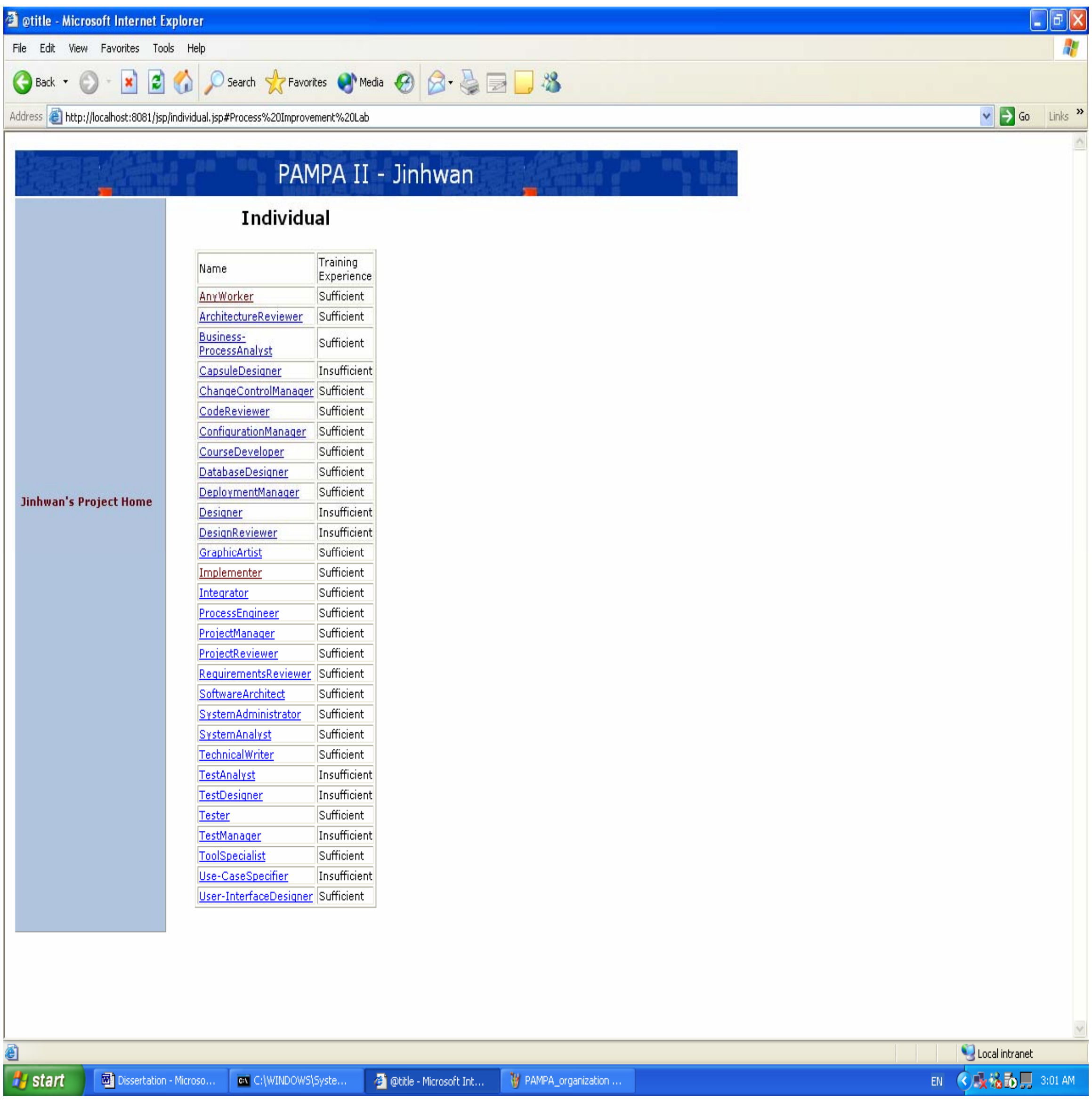
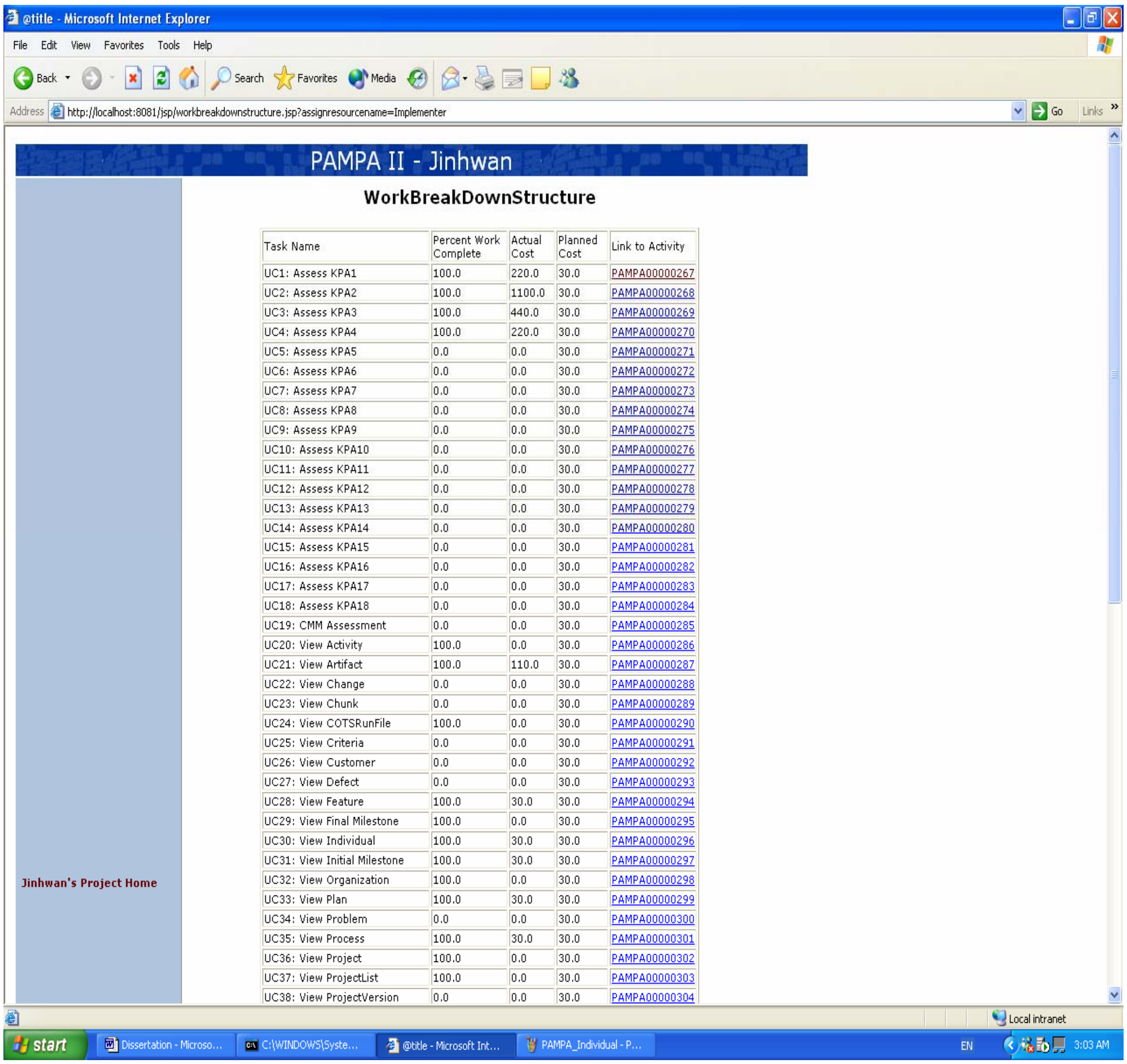


Figure 20. Individual

14. WorkBreakDownStructure

After you click the name of **Individual** you can see the object **WorkBreakDownStructure** and its attributes [Figure 21]. The attributes are task name, percentage of work completed, actual cost, and planned cost. There is a link to **Activity**, which shows you the related **Activity** for the specific **WorkBreakDownStructure**.



Task Name	Percent Work Complete	Actual Cost	Planned Cost	Link to Activity
UC1: Assess KPA1	100.0	220.0	30.0	PAMPA00000267
UC2: Assess KPA2	100.0	1100.0	30.0	PAMPA00000268
UC3: Assess KPA3	100.0	440.0	30.0	PAMPA00000269
UC4: Assess KPA4	100.0	220.0	30.0	PAMPA00000270
UC5: Assess KPA5	0.0	0.0	30.0	PAMPA00000271
UC6: Assess KPA6	0.0	0.0	30.0	PAMPA00000272
UC7: Assess KPA7	0.0	0.0	30.0	PAMPA00000273
UC8: Assess KPA8	0.0	0.0	30.0	PAMPA00000274
UC9: Assess KPA9	0.0	0.0	30.0	PAMPA00000275
UC10: Assess KPA10	0.0	0.0	30.0	PAMPA00000276
UC11: Assess KPA11	0.0	0.0	30.0	PAMPA00000277
UC12: Assess KPA12	0.0	0.0	30.0	PAMPA00000278
UC13: Assess KPA13	0.0	0.0	30.0	PAMPA00000279
UC14: Assess KPA14	0.0	0.0	30.0	PAMPA00000280
UC15: Assess KPA15	0.0	0.0	30.0	PAMPA00000281
UC16: Assess KPA16	0.0	0.0	30.0	PAMPA00000282
UC17: Assess KPA17	0.0	0.0	30.0	PAMPA00000283
UC18: Assess KPA18	0.0	0.0	30.0	PAMPA00000284
UC19: CMM Assessment	0.0	0.0	30.0	PAMPA00000285
UC20: View Activity	100.0	0.0	30.0	PAMPA00000286
UC21: View Artifact	100.0	110.0	30.0	PAMPA00000287
UC22: View Change	0.0	0.0	30.0	PAMPA00000288
UC23: View Chunk	0.0	0.0	30.0	PAMPA00000289
UC24: View COTSRunFile	100.0	0.0	30.0	PAMPA00000290
UC25: View Criteria	0.0	0.0	30.0	PAMPA00000291
UC26: View Customer	0.0	0.0	30.0	PAMPA00000292
UC27: View Defect	0.0	0.0	30.0	PAMPA00000293
UC28: View Feature	100.0	30.0	30.0	PAMPA00000294
UC29: View Final Milestone	100.0	0.0	30.0	PAMPA00000295
UC30: View Individual	100.0	30.0	30.0	PAMPA00000296
UC31: View Initial Milestone	100.0	30.0	30.0	PAMPA00000297
UC32: View Organization	100.0	0.0	30.0	PAMPA00000298
UC33: View Plan	100.0	30.0	30.0	PAMPA00000299
UC34: View Problem	0.0	0.0	30.0	PAMPA00000300
UC35: View Process	100.0	30.0	30.0	PAMPA00000301
UC36: View Project	100.0	0.0	30.0	PAMPA00000302
UC37: View ProjectList	100.0	0.0	30.0	PAMPA00000303
UC38: View ProjectVersion	0.0	0.0	30.0	PAMPA00000304

Figure 21. WorkBreakDownStructure

15. Activity (Related to WorkBreakDownStructure)

Figure 22 shows the **Activity** related to a specific **WorkBreakDownStructure**. You can see the **Artifact** created by this **Activity** by clicking the Link to **Artifact**.

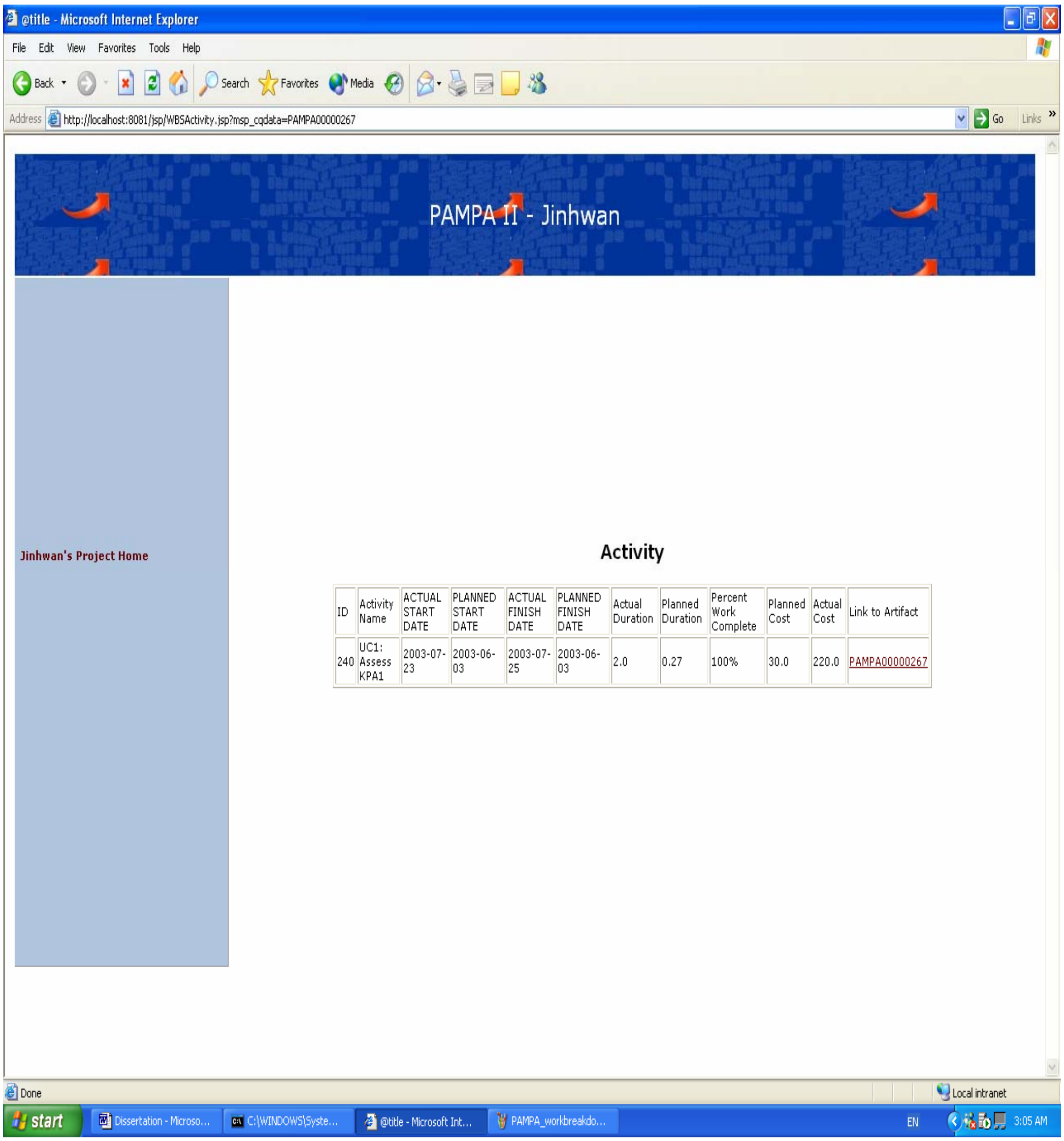


Figure 22. Activity (Related to WorkBreakDownStructure)

16. Software Product

Figure 23 shows the object **SoftwareProduct** and its attributes. You can see object **Feature** and its attributes under **SoftwareProduct** by clicking the name.

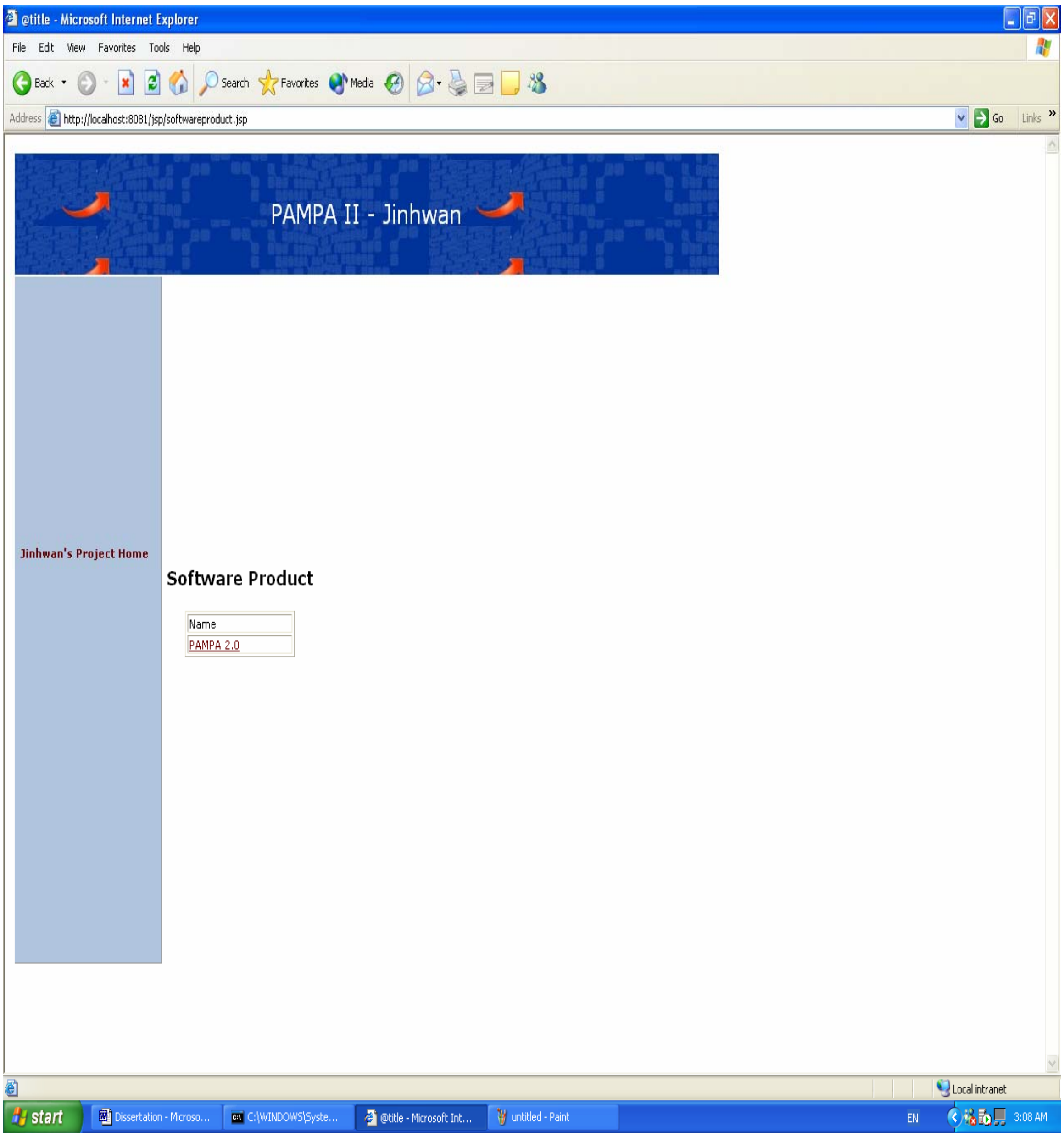


Figure 23. Software Product

17. Feature

Figure 24 shows the object **Feature** and its attribute's name, status, assigned resource name, and description after you click the name of **SoftwareProduct**. The link to **Artifact** shows the related **Artifact** to a specific **Feature**.

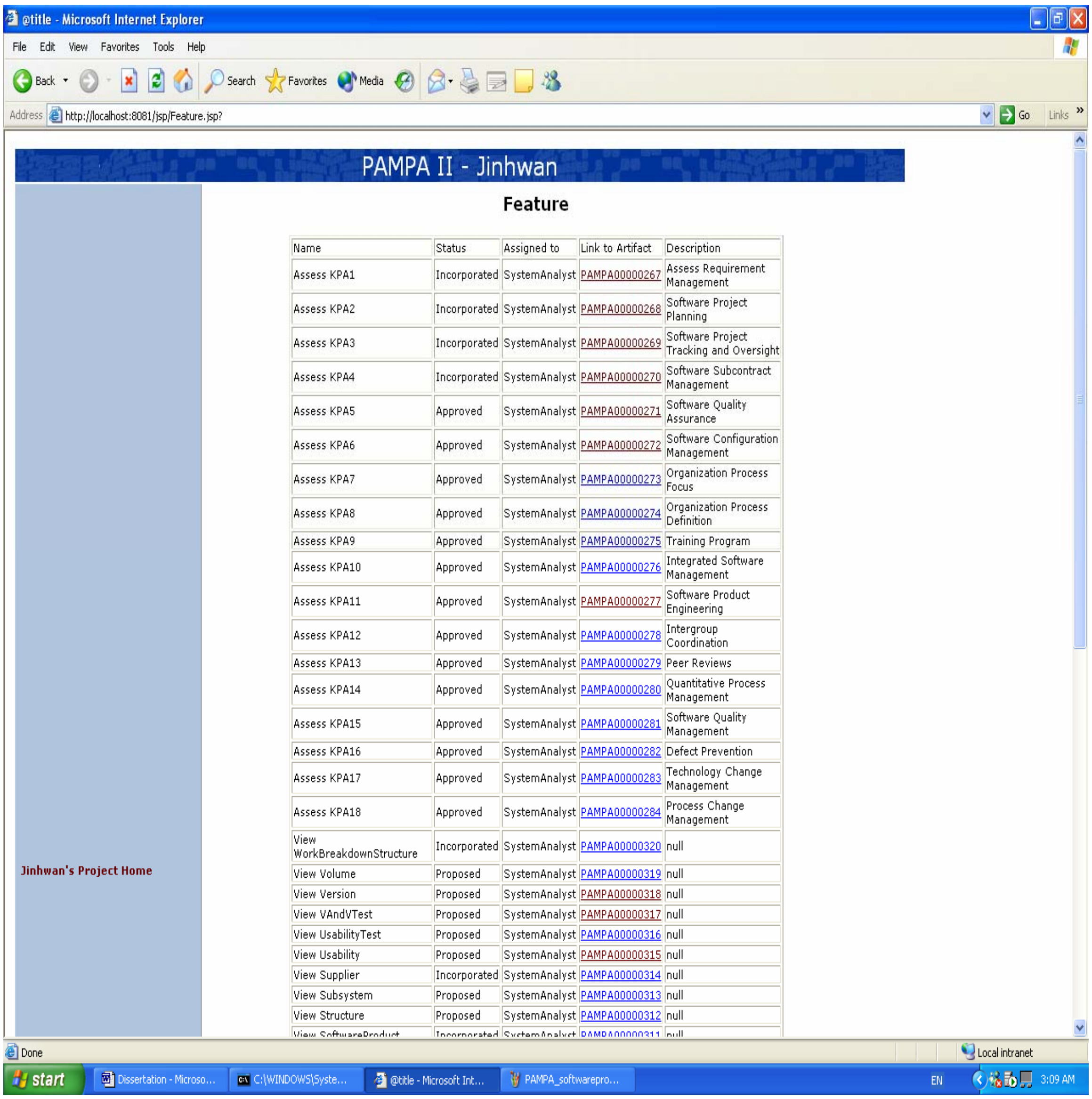


Figure 24. Feature

18. Artifact (Related to Feature)

Figure 25 shows the object **Artifact** and its attributes such as Artifact Name, creater, and files size in byte after you click the link to **Artifact** from **Feature**. The link Associated **Activity** will show you **Activity** related to a specific **Artifact**.

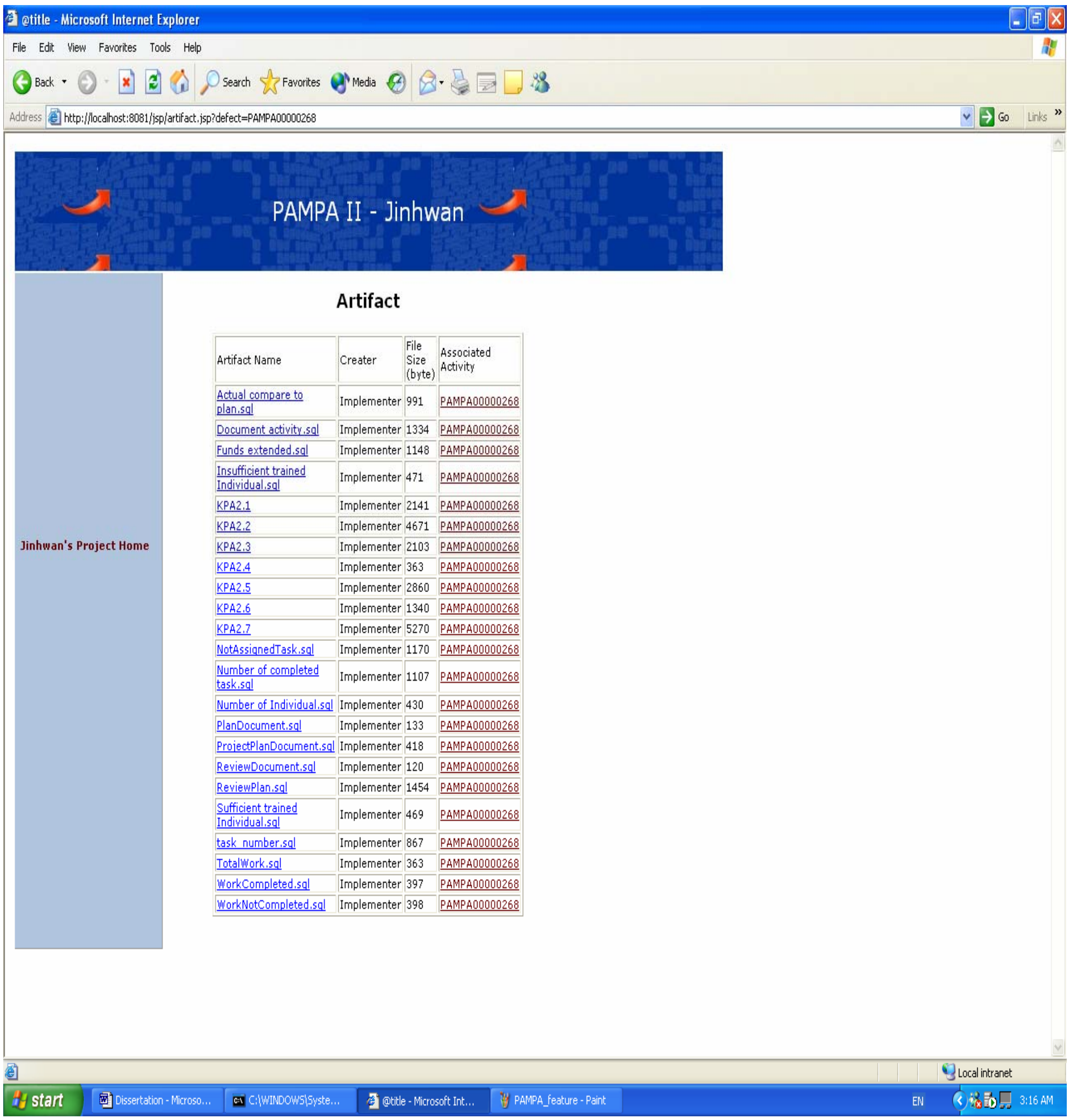


Figure 25. Artifact (Related to Activity)

19. Customer

Figure 26 shows the object **Customer** and its attributes such as customer name, company name, E-Mail address, and phone number. You can see what the Customer requested by clicking Customer name.

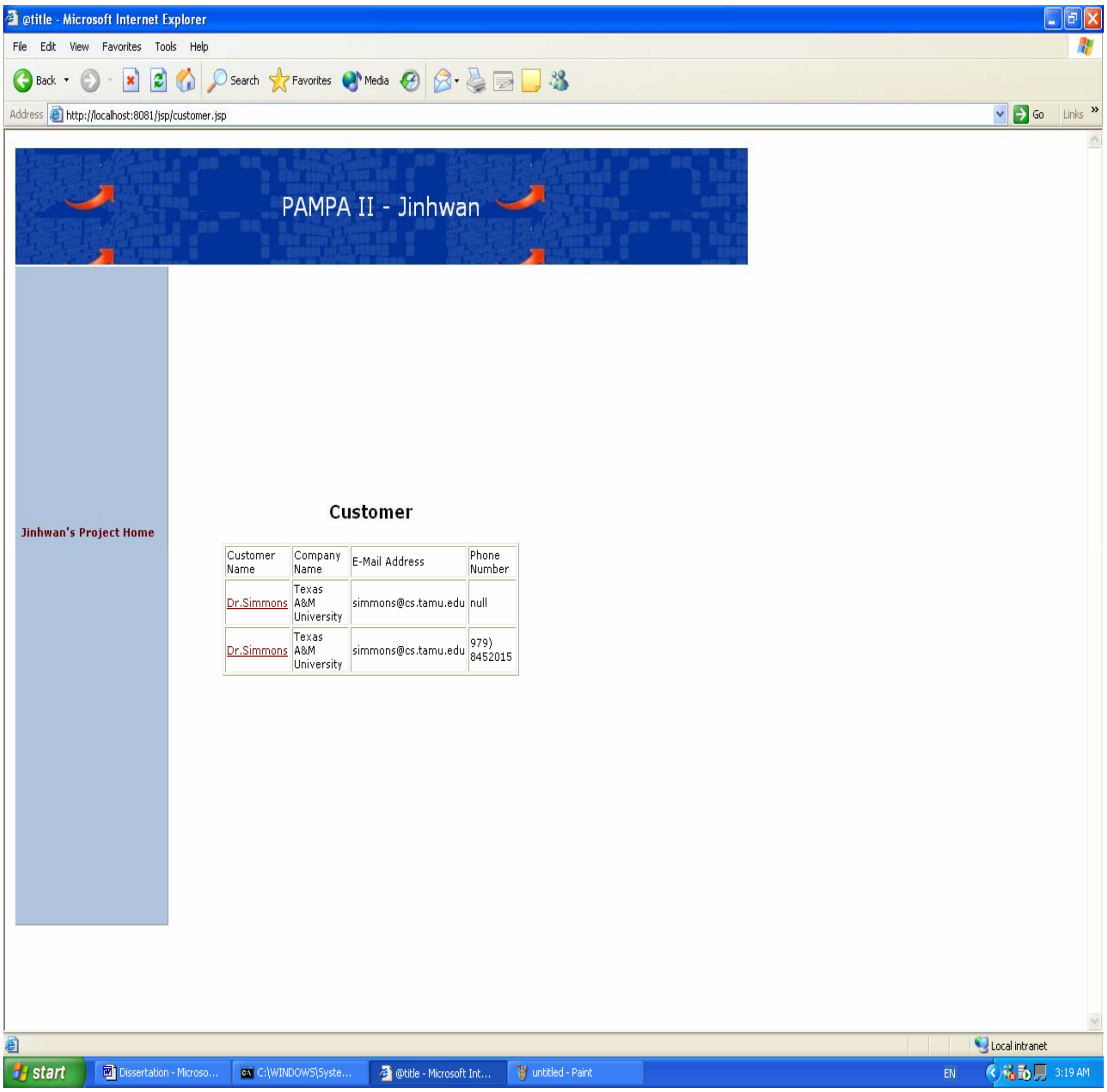


Figure 26. Customer

20. Problem

Figure 27 shows the object **Problem** and its attributes such as task name, priority (high, medium), and description. You can see how this **Problem** is handled by clicking the link to **Activity**.

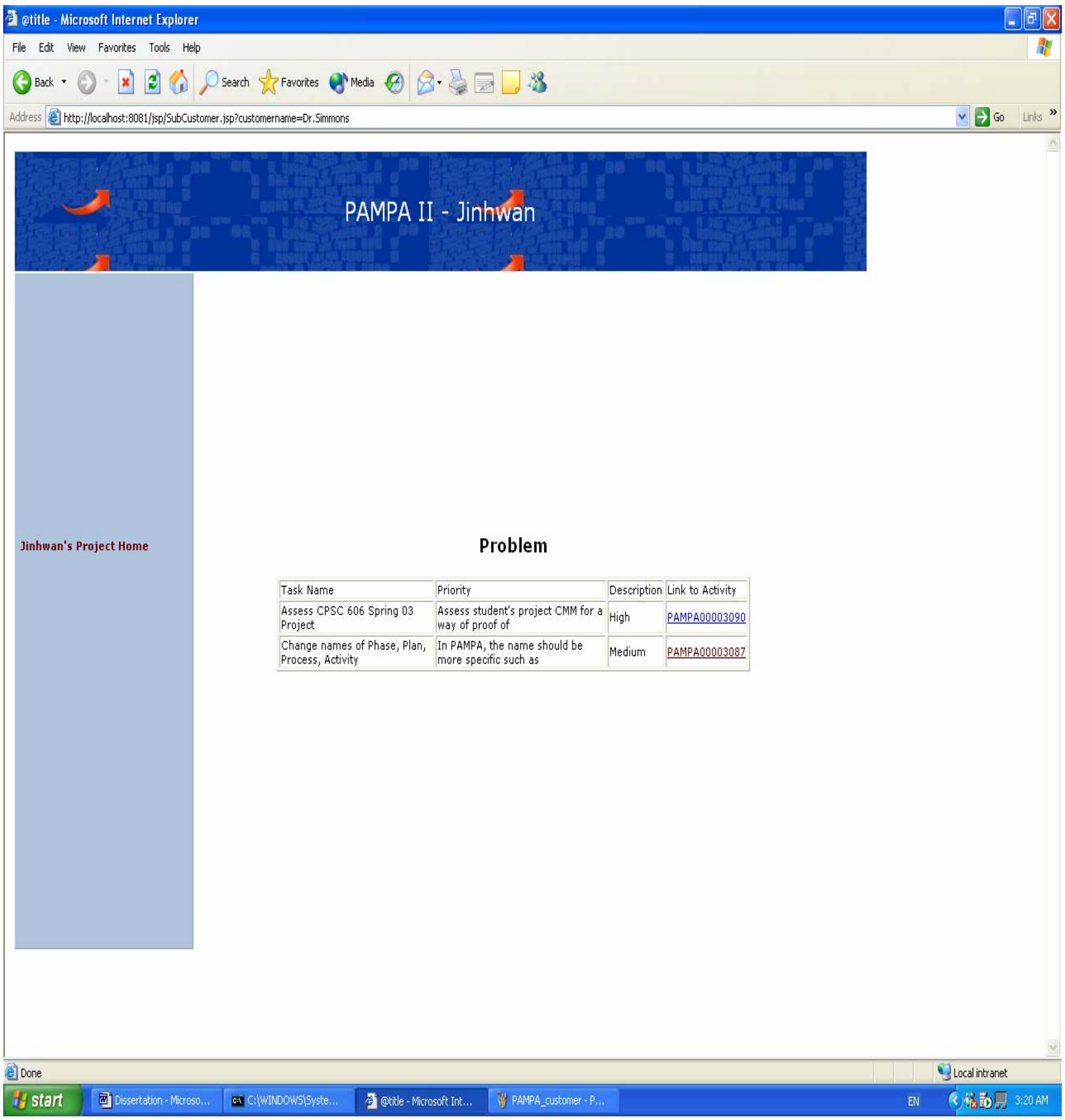


Figure 27. Problem

C. Assess Process Maturity

1. Facts extracted from *Jinhwan's Project*

Figure 28 shows the facts that are extracted from *Jinhwan's Project* attributes.

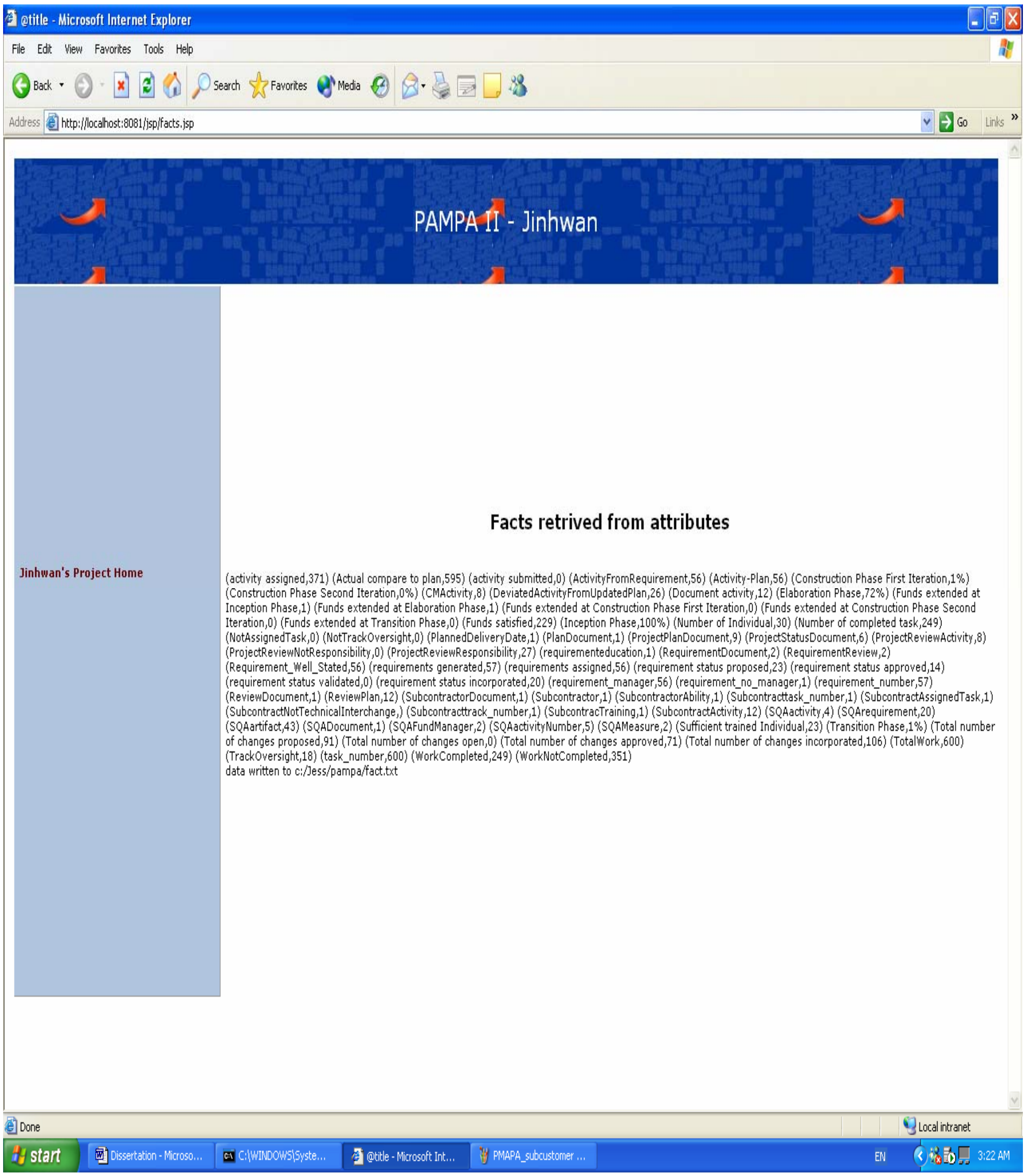


Figure 28. Facts from Jinhwan's Project

2. Facts extracted from CPSC606 Project

Figure 29 shows the facts that are extracted from *CPSC606 Project* attributes.

Figure 29. Facts from CPSC606 Project

3. Process assessment and monitoring for *Jinhwan's Project*

Figure 30 shows the result of CMM assessment and project status of *Jinhwan's Project*.

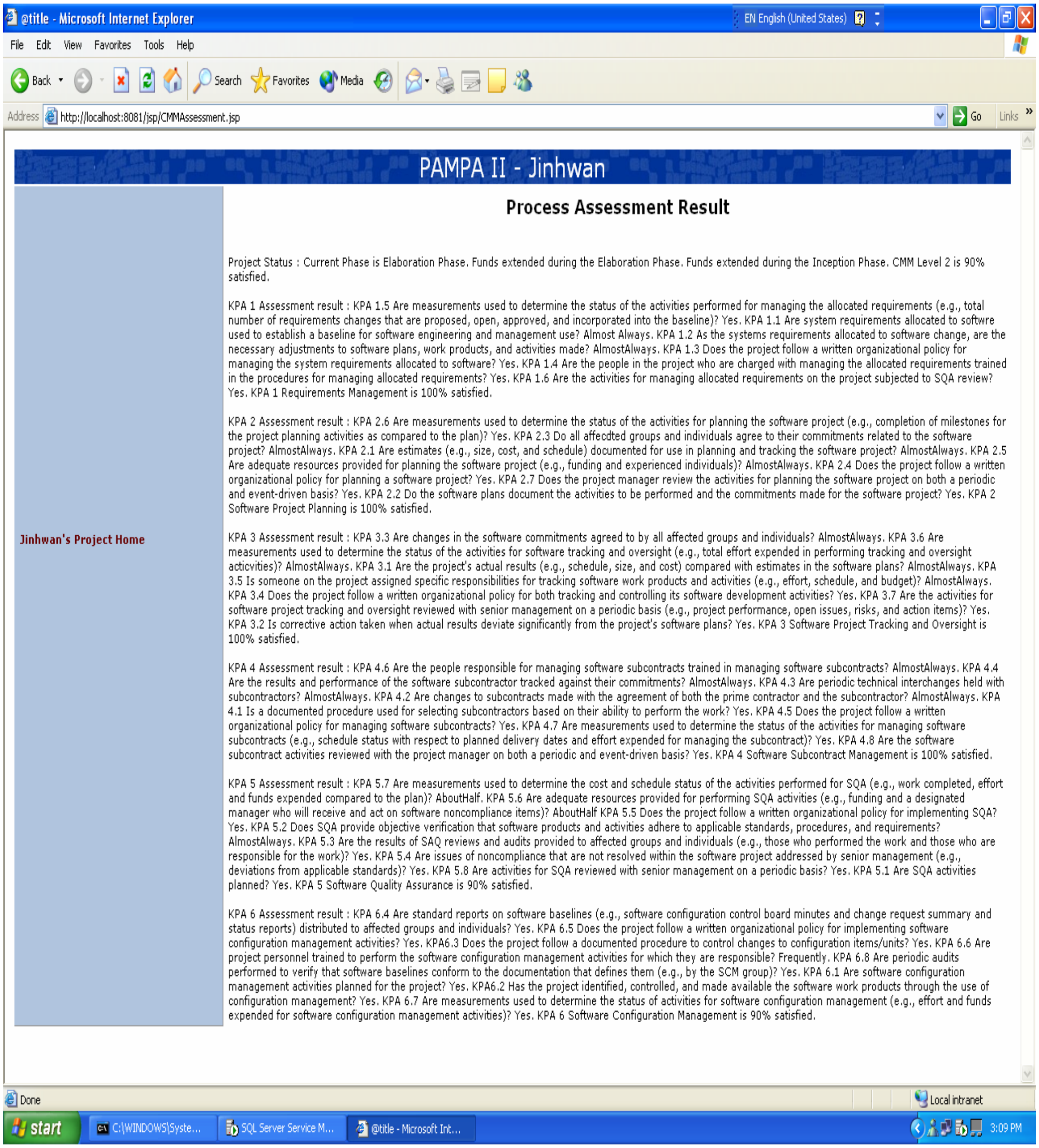


Figure 30. Assessment Result from Jinhwan's Project

4. Process assessment and monitoring for *CPSC606 Project*

Figure 31 shows the result of CMM assessment and project status of *CPSC606 Project*.

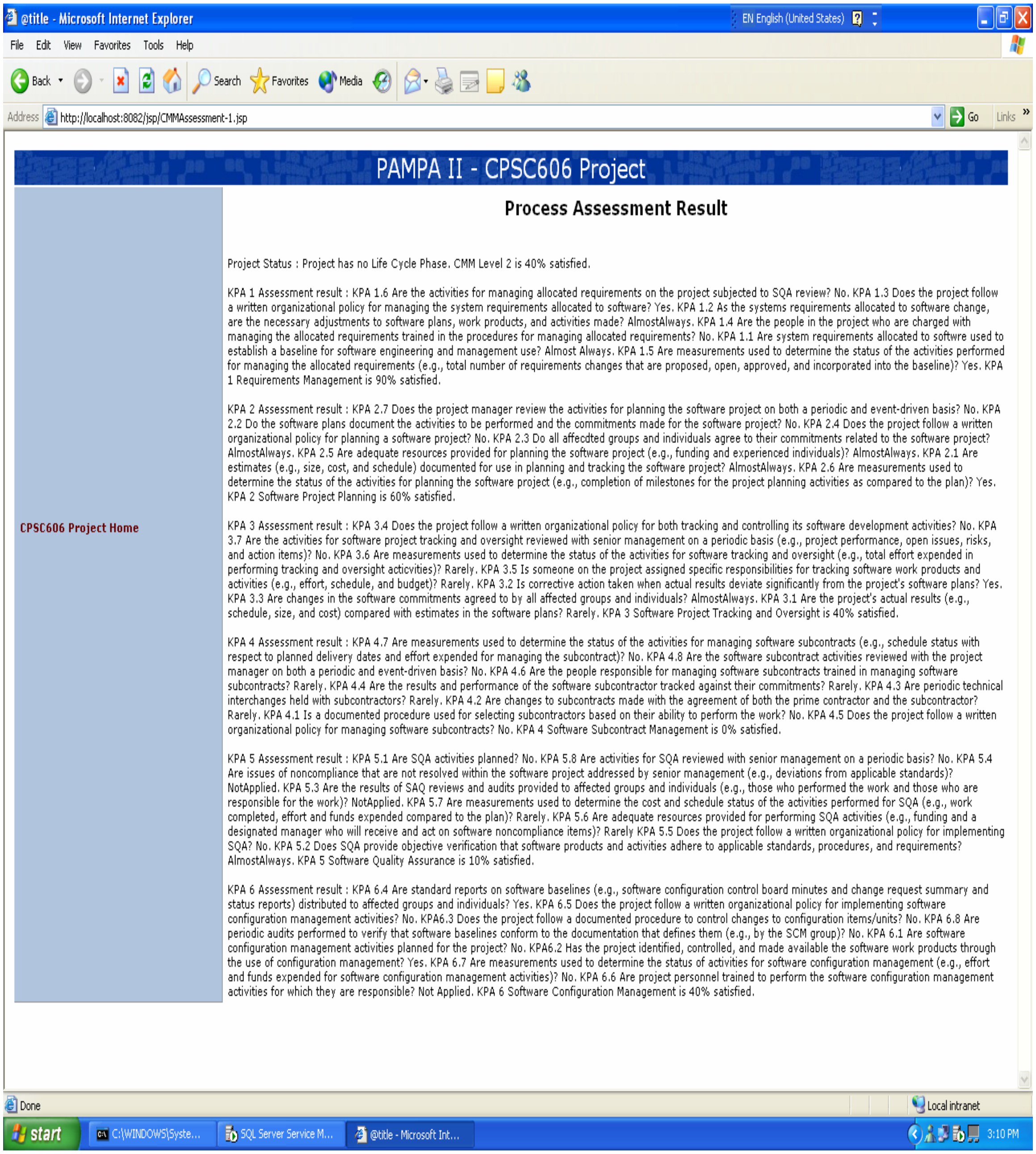


Figure 31. Assessment Result from CPSC606 Project

5. EVALUATION AND CONCLUSION

To check whether the objective assessment was implemented correctly, I compared two projects. *Jinhwan's Project* was implemented by Jinhwan Jung, whose experience is 3 years of software engineering and 2 years of Rational Tools and MS project. With this knowledge I assumed that I could satisfy CMM Level 2 in *Jinhwan's Project.* I did so. *CPSC606 Project* was implemented by students from a spring 2003 course, CPSC 606 Software Engineering. Since most of them are new to Software Engineering and do not have enough knowledge about Rational Tools and MS Project, I assumed that they would not satisfy CMM Level 2. They did not [Table 7].

Table 7. Assessment Results

Project Name	Implement group or individuals	Experience	Hypothesis	PAMPA 2.0 Assessment Result (% Satisfied)
Jinhwan's Project	Jinhwan	3 years of Software Engineering. 2 years of Rational Tools & MS Project.	Satisfies Level 2	KPA 1. Requirements Management : 100 % KPA 2. Software Project Planning : 100 % KPA 3. Software Project Tracking and Oversight : 100 % KPA 4. Software Subcontract Management : 100 %

Table 7 continued.

Jinhwan's Project	Jinhwan	3 years of Software Engineering. 2 years of Rational Tools & MS Project.	Satisfies Level 2	KPA 5. Software Quality Assurance : 90 % KPA 6. Configuration Management : 90% => Satisfies Level 2
CPSC606 Project	CPSC 606 Spring 2003 class students	Most of them are new to Software Engineering and Rational Tools and MS Project	Does not satisfy Level 2	KPA 1. Requirements Management : 90 % KPA 2. Software Project Planning : 60 % KPA 3. Software Project Tracking and Oversight : 40 % KPA 4. Software Subcontract Management : 0 % KPA 5. Software Quality Assurance : 10 % KPA 6. Configuration Management : 40 % => Does not satisfy Level 2

As the software development system environment grows in size, the object and attribute relationships become more complex. Then it becomes more difficult to assess CMM subjectively. In this research I proved that objective assessment CMM using

PAMPA 2.0 is possible, and compared to the subjective CMM assessment, the objective CMM assessment is both less expensive and easier to calculate maturity, and takes less time.

I have shown that PAMPA 2.0 can be used in various criteria:

- Utility to the Project Management: Project manager can measure and monitor the process.
- Utility to Customer: Customers can check status of their requirement and working progress.
- Utility to Individual member of team: Individual developers can check what their task is and working on that task without any confusion.

6. FUTURE WORK

A. Objective COCOMO II Scale Factor Measurement

COCOMO II provides effort and schedule estimates [22]. The Scale factors are size exponent E in the effort prediction equation, which calculates the amount of effort in person-months, PM_{NS}, estimated by the formula:

$$PM_{NS} = 2.94 \times Size^{E} \times \prod_{i=1}^{n} EM_i \text{ where } E = 0.91 + 0.01 \times \sum_{j=1}^{5} SF_j$$

The amount of calendar time, $TDEV_{NS}$, to develop the product is estimated by the formula:

$$TDEV_{NS} = 3.67 \times (PM_{NS})^{F} \text{ where } F = 0.28 + 0.2 \times (E - 0.91)$$

We assume that the cost driver rating levels are nominal so the effort multipliers, EM_i, are all 1.00, and we consider only the exponential scale factors SF_j. Scale factors, which are Precedentedness (PREC), Development Flexibility (FLEX), Architecture/Risk Resolution (RESL), Team Cohesion (TEAM), and Process Maturity (PMAT), have weights according to the range of their rating levels from very low to extra high. These are estimated subjectively, which makes calculation inaccurate and time-consuming.

Subjective measurement can be replaced by an expert system that provides estimates based on objective measurement and knowledge acquisition from experts. Knowledge can be acquired from software development experts to create a knowledge base. Metrics gathered from a development environment can drive an expert system to predict scale factors.

The PAMPA 2.0 tool can be extended to predict COCOMO II scale factors in calculating the equation of size measurement. Figure 32 shows PAMPA 2.0 gathering facts from project files using metrics and CASE tools. These facts are stored to a knowledge base as objects and attributes. They are sent to an Inference Engine to predict COCOMO II model scale factors.

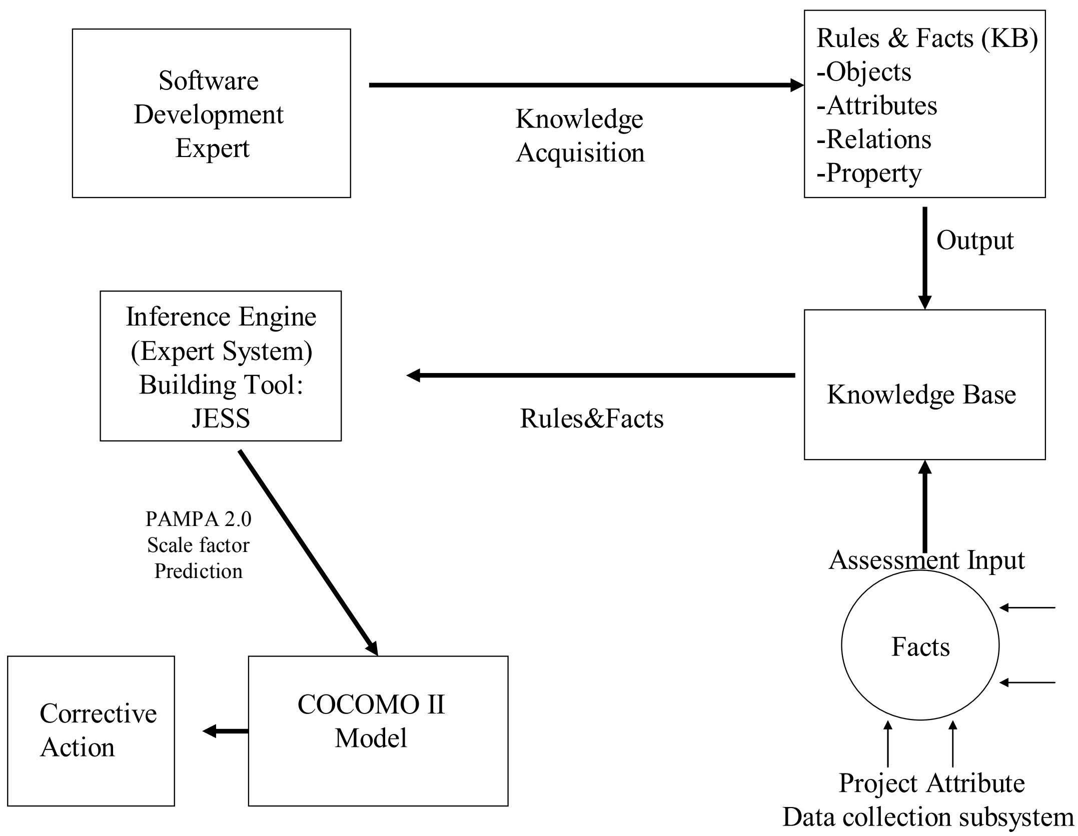


Figure 32. COCOMO II Scale Factor Prediction

Jung and Simmons [7,8] showed objective assessment of a software organization maturity based on a Capability Maturity Model (CMM), using PAMPA 2.0. PMAT, which is one of the scale factors and is predicted directly from CMM, can be measured objectively.

B. Requirement Measurement

It is important to predict how much effort is required for a requirement. In a software development project, if you do not allocate proper resources to the requirement based on correct prediction, it would be difficult to manage a project schedule. This can make the project fail. One of the solutions to predict requirement effort correctly could be by using PAMPA 2.0 and Rational Tools. Requirements generated through RequistePro are sent to Rational Rose for designing. PAMPA 2.0 gathers object **Structure** and its attributes from Rational Rose. **Structure** is the basic element of **Feature**, which is the requirement. Based on the **Structure** attributes, PAMPA 2.0 can predict the amount of resources necessary for implementing the requirements. This information will be sent to the Project Manager for plans [Figure 33].

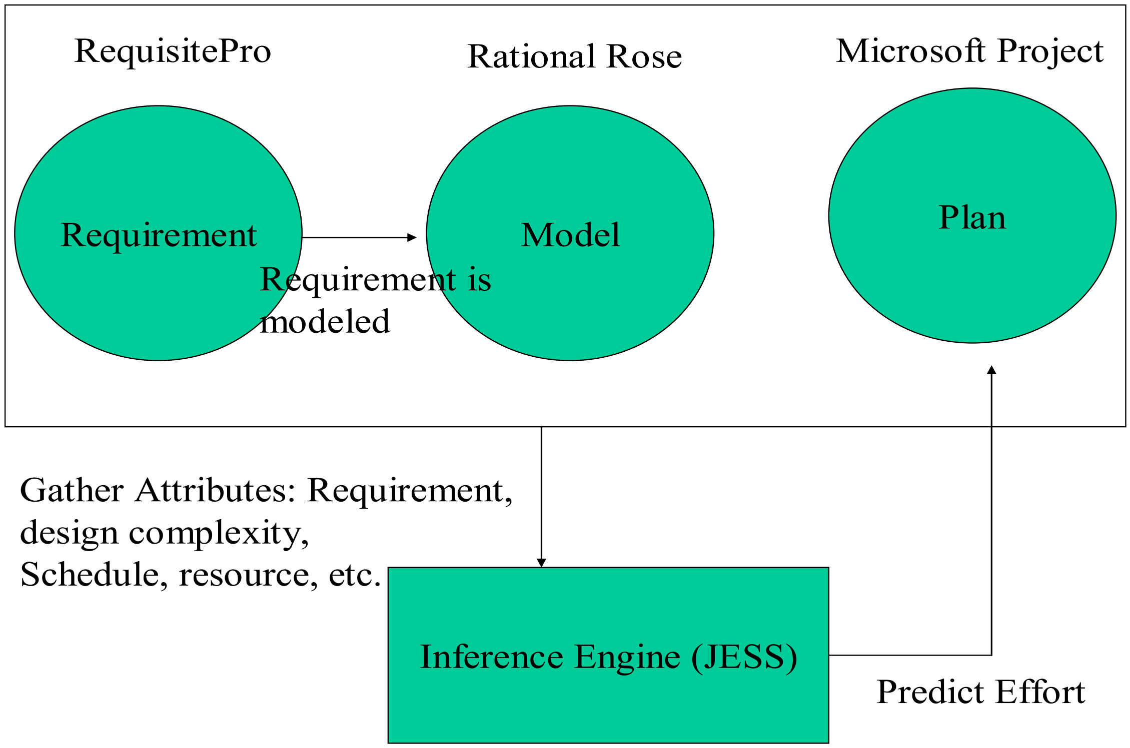


Figure 33. Requirement Measurement

REFERENCES

[1] D. B. Simmons, N. C. Ellis, H. Fujihara, and W. Kuo, *Software Measurement - A Visualization Toolkit for Project Control & Process Improvement,* Prentice Hall, 1998.

[2] D. B. Simmons, A win-win metric based software management approach, *IEEE Transactions on Engineering Management*, Vol. 39, No. 1, February, 1992, pp. 32~41.

[3] D. B. Simmons, N.C. Ellis, and T. D. Escamilla, Manager associate, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 3, June, 1993, pp. 426~438.

[4] K. Huarng, and D. B. Simmons, An object knowledge canonical form for knowledge reuse, *Expert Systems with Applications: An International Journal*, Vol. 10, No. 1, Jan 1996, pp. 135~146.

[5] D. B. Simmons, C. S. Wu, Plan tracking knowledge base, *Proceedings of the Twenty-Fourth annual International Computer Software and Applications Conference*, Taipei, Taiwan, October 25~27, 2000, pp. 299~304.

[6] C. S. Wu, D. B. Simmons, Software Project Planning Associate (SPAA): A knowledge based approach for dynamic software project planning and tracking, Proceedings of the Twenty-Fourth annual International *Computer Software and Applications Conference*, Taipei, Taiwan, October 25~27, 2000, pp. 305~310.

[7] J. Jung, and D. B. Simmons, CASE Tool Based Process Maturity Assessment, *Proceedings of the 2002 International Conference on Software Engineering Research and Practice (SERP'02)*, Las Vegas, Nevada June 24~27, 2002, pp.329~335.

[8] J. Jung, and D. B. Simmons, Capability assessment tool, *Proceedings of the 6th World Multiconference on Systemics, Cybernatics and Informatics,* Orlando, Florida, July 15~18, 2002, pp. 6~11.

[9] A. Perlis, F. Sayward, and M. Shaw, *Software Metrics*, The MIT Press, 404 pgs, 1981.

[10] http://www.sei.cmu.edu, CMM models, Sonftware Engineering Institute, April 2003.

[11] D. M. Ahern, A. Clouse, R. Turner, *CMMI Distilled*, Addison-Wesley Publishing Company, Massachusetts, 2002, 306 pgs.

[12] W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989, 494 pgs.

[13] B. K. Clark, Quantifying the effects of process improvement on effort, IEEE Software Vol. 17, No. 6, 2000, pp 65~70.

[14] J. R. Persse, *Implementing the Capability Maturity Model*, Wiley, 2001.

[15] M. C. Paulk, C. V.Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1995, 441.

[16] W. Li, Software Product metrics, IEEE, 2000, pp.24~27.

[17] L. Tahvildary, A. Singh, Categorization of Object-Oriented Software Metrics, *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, Halifax, Canada, May 2000, pp.235~239.

[18] N. C. Debnath, R. Y.Lee, An analysis of software engineering metrics in OO Environment, *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'01)*, Beirut, Lebanon, June 2001, pp.492~494.

[19] T. M. Khoshgoftaar, E. B.Allen, Using Process History to Predict Software Quality, IEEE, 1998, pp. 66~72.

[20] S.Masters and C.Bothwell, CMM Appraisal Framework, version 1.0, *Technical Report CMU/SEI-95-TR-001, Software Engineering Institute*, 1995.

[21] http://www.incose.org/intro.html, Case Tools, May 2003.

[22] B. W. Boehm, B. Steece, H. Clark, R. Brown, R. M. Chulani, *Software Cost Estimation with COCOMO II*, Prentice Hall, New Jersey, 306 pgs. January 2000.

[23] V. R. Basili. Software modeling and measurement: The Goal/Question/Metric paradigm. *Technical Report CS-TR-2956*, Department of Computer Science, University of Maryland, College Park, MD 20742, September 1992.

[24] G. Natwick, Harris Corporation, Integrated metrics for CMMI and SW-CMM, *The Journal of Defense Software Engineering*, May 2003.

[25] D.Karami and P.W.Garratt SCAMES: An expert system to assist in using CMM, *Proceedings Fourth International Conference on Software Quality*, 1994.

[26] http://www.rational.com, Product review, Feb. 2002.

[27] Using MetricCenter Workstation for Meeting the Requirements of the SEI's Software CMM Level 2, Distributive Software.

[28] Reaching CMM Levels 2 and 3 with the Rational Unified Process, Rational Software Corporation, Sep.1998.

[29] K. E. Wiegers, D. C. Sturzenberger, A modular software process mini-assessment method, *IEEE Software* January/ February 2000, pp. 62~69.

[30] M. Frey-Pucko, R. Novak, G. Kandus, SEPAM–CMM Implementation for Developing Telecommunications Systems, *27th Euromicro Conference 2001: A Net Odyssey (euromicro'01),* Warsaw, Poland, September 04~06, 2001, pp. 272~278.

[31] H. Ghosh, A Comparison of ISO 9000 and SEI/CMM for software engineering organizations, IEEE, 2000, pp.78~83, accessible at : www.comp.dit.ie/rlawlor/Soft-Eng/CMM%and%20iso.pdf.

[32] J. Ferguson and S. Sheard, Leveraging Your CMM Efforts for IEEE/EIA 12207, IEEE, 1998, pp.23-28.

[33] http://www.upedu.org/upedu/references/papers, ISO 12207 RUP, May 2003.

[34] Software Process Maturity Questionnaire, SEI, April 1994, 57 pgs.

APPENDIX A

KPA ASSESSMENT

A. KPA 1. Requirements Management

The purpose of Requirements Management is to establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project.

KPA 1.1: Are system requirements allocated to software used to establish a baseline for software engineering and management use?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated), ***Administer***$_{set}$	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	Requirement Well Stated
	RequirementNumber$_{Calculate}$			Requirement number
	Administer$_{set}$			Requirement manager

2) The heuristics used to make an assessment based on that data

The facts *Requirement_Well_Stated, requirement_number,* and *requirement_manager* are used in the JESS rule. *Requirement_Well_Stated* counts the number of requirements stated well, which have attributes such as name, status, administer, and related **Artifact**s. The *requirement_number* calculates the number of all the requirements in the project including not well stated requirements. The *requirement_manager* counts the number of administers assigned to each requirement. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{\mathrm{Re}quirement\_Well\_Stated}{requirement\_number}+\frac{requirement\_namager}{requirement\_number})\times\frac{1}{2}$ is more than 0.9, it means 90% of the requirements are well stated and have administer to review the requirements.

```
(defrule KPA1-1AlmostAlways
(Requirement_Well_Stated, ?x)  (requirement_number, ?y)  (requirement_manager, ?v)
(requirement_number, ?w)
(test (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?v ?w)))))
=>
(assert (KPA1-1 AlmostAlways))
(printout Result.txt "KPA 1.1 Are system requirements allocated to software used to establish a baseline
for software engineering and management use? Almost Always." crlf))
--------------------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the software engineering group reviews the allocated requirements before they are incorporated into the software project. There should be no incomplete and missing allocated requirements and the allocated requirements are reviewed [28].

The heuristics above finds out the number of incomplete requirements which do not have name, status, administer, and the related **Artifact**s. And it calculates the ratio of administer, who reviews the requirements, to requirements.

4) The limitations of the data and heuristics

None

KPA 1.2: As the systems requirements allocated to software change, are the necessary adjustments to software plans, work products, and activities made?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***Name***$_{set}$, ***Type***$_{set}$(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM,</u>		Project Planning Tool (MS Project)	Activity from Requirement, Activity Number

	PCM)			
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)		Requirements management Tool (RequisitePro)	Requirement Number, Feature Status
Artifact	***Name***$_{set}$,**Type**(Requirements File, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$	**Artifact**s are created by **Activit(y)**ies	Configuration Management Tool (ClearCase) & Design Tool (Rational Rose)	Artifact from Activity

2) The heuristics used to make an assessment based on that data

The facts *Activity from Requirement, Requirement Number, Activity Number, Artifact from Activity,* and *Feature Status* are used in the JESS rule. *Activity from Requirement* counts the number of **Activit(y)**ies initiated by the requirements. *Requirement Number* counts the number of requirements. *Activity Number* counts the number of requirement related **Activit(y)**ies. *Artifact from Activity* counts the number of **Artifact**s created by the **Activit(y)**ies. *Feature Status* checks the existence of requirement status attributes. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{ActivityFrom\mathrm{Re}quirement}{\mathrm{Re}quirementNumber}$ and $\frac{ArtifactFromActivity}{ActivityNumber}$ is more than 0.9, it means 90% of the requirements have initated related **Activit(y)**ies, and the **Activit(y)**ies created **Artifact**s. And it also checks *Feature Status* is more than 1, which means the requirement status attributes are exist.

```
(defrule KPA1-2AlmostAlways
 (Activity from Requirement, ?x)  (Requirement Number, ?y) (Artifact from Activity, ?u) (Activity
Number, ?v)
 (Feature Status, ?w)
 (test (and (<= 0.9 (/ ?x ?y) ) (<= 0.9 (/ ?u ?v)) (<= 1 ?w)))
 =>
 (assert (KPA1-2 AlmostAlways))
 (printout Result.txt "KPA 1.2 As the systems requirements allocated to software change, are the
necessary adjustments to software plans, work products, and activities made? AlmostAlways." crlf))
----------------------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the software engineering group uses the allocated requirements as the basis for software plans, work products, and activities. And changes to the allocated requirements are reviewed and incorporated into the software project [28]. For each requirement, there should be related **Activity** that creates

Artifacts. A Requirement from RequisitePro is linked to MS Project, which is an **Activity** planning tool, as **Activity**. Relationships between the requirement, **Activity,** and **Artifact**s show how the software engineering group uses the allocated requirements as the basis for software plans**.**

The heuristics checks how the software engineering group uses the allocated requirements as the basis for **Activit(y)**ies and **Artifact**s. Whenever the system requirements allocated to software are changed, the affected **Activit(y)**ies, and **Artifact**s are adjusted to remain consistent with the updated requirements. If the requirement is added, there should be related **Activit(y)**ies, and **Artifact**s are created. If the requirement is deleted, the related **Activit(y)**ies and **Artifact**s are deleted. The heuristics also checks the existence of requirement attributes, and if they are exists, it is means that requirement status changes to the allocated requirements are reviewed and incorporated into the software project.

4) The limitations of the data and heuristics

None

KPA 1.3: Does the project follow a written organizational policy for managing the system requirements allocated to software?

1) The data available related to the CMM analysis

PAMPA 2.0	Source	Facts elicited

Object	Attributes	Relationships		
Organization	***Name***$_{set}$***, Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	RM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			RM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			RM Activity

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	RM Document

2) The heuristics used to make an assessment based on that data

The facts *RM Group, RM Tool, RM Activity,* and *RM Document* are used in the JESS rule. *RM Group* counts the number of responsible group or **Individual**s assigned to requirements management. *RM Activity* counts the number of **Activit(y)**ies related to requirements management. *RM Tool* counts the number of tools to use for requirements management. *RM Document* counts the number of documents related to requirements Management. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks that each of *RM Group, RM Tool, RM Activity,* and *RM Document* is more than 1, which means requirements management-related groups or **Individual**s, **Activit(y)**ies, tools and documents are exist.

```
(defrule KPA1-3Yes
 (RM Group, ?w) (RM Tool, ?x) (RM Activity, ?y) (RM Document, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
```

```
 (assert (KPA1-3 Yes))
 (printout Result.txt "KPA 1.3 Does the project follow a written organizational policy for managing the
system requirements allocated to software? Yes." crlf))

(defrule KPA1-3No
 (RM Group, ?w) (RM Tool, ?x) (RM Activity, ?y) (RM Document, ?z)
  (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA1-3 No))
 (printout Result.txt "KPA 1.3 Does the project follow a written organizational policy for managing the
system requirements allocated to software? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, documents, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of groups, **Individual**s, **Activit(y)**ies, tools, and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to requirements management and document related to requirements management should be predefined and recongnized in the knowledge base.

KPA 1.4 Are the people in the project who are charged with managing the allocated requirements trained in the procedures for managing allocated requirements?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(Software development years)		Project Planning Tool (MS Project)	Requirementeducation

2) The heuristics used to make an assessment based on that data

The fact *requirementeducation* is used in the JESS rule. The *requirementeducation* checks whether the System Analyst's experience is sufficient. One of the System Analyst's roles is managing requirements. The fact is gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the *requirementeduation* is 1, which means the System Analyst's experience is sufficient.

```
(defrule KPA1-4Yes
(requirementeducation, ?x)
(test (= 1  ?x ))
```

```
  =>
 (assert (KPA1-4 Yes))
 (printout Result.txt "KPA 1.4 Are the people in the project who are charged with managing the allocated
requirements trained in the procedures for managing allocated requirements? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that members of the software engineering group and other software-related groups are trained to perform their requirements management activities [28]. The System Analyst is assigned to manage requirements. By checking the training experience of the System Analyst, we can understand whether the people in the project who are charged with managing the allocated requirements are trained in the procedures for managing allocated requirements.

4) The limitations of the data and heuristics

None

KPA 1.5 Are measurements used to determine the status of the activities performed for managing the allocated requirements (e.g, total number of requirements changes that are proposed, open, approved, and incorporated into the baseline)?

1) The data available related to the CMM analysis

PAMPA 2.0	Source	Facts elicited

Object	Attributes	Relationships		
Feature	***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)		Requirements management Tool (RequisitePro)	Total number of changes proposed, Total number of changes open, Total number of changes approved, Total number of changes incorporated

2) The heuristics used to make an assessment based on that data

The facts *Total number of changes proposed, Total number of changes open, Total number of changes approved,* and *Total number of changes incorporated* are used in the JESS rule. All the facts count the number of requirements proposed, open, approved, and incorporated in the project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule that checks whether there is any number counted on requirement proposed, open, approved, and incorporated. If the number is counted, we know the measurements are used to determine the status of the **Activit(y)**ies performed for managing the allocated requirements.

```
(defrule KPA1-5Yes
```

(Total number of changes proposed, ?x) (Total number of changes open, ?y) (Total number of changes approved, ?z) (Total number of changes incorporated, ?w) (test (or(< 0 ?x)(< 0 ?y)(< 0 ?z)(< 0 ?w)))

=>

(assert (KPA1-5 Yes))

(printout Result.txt "KPA 1.5 Are measurements used to determine the status of the activities performed for managing the allocated requirements (e.g., total number of requirements changes that are proposed, open, approved, and incorporated into the baseline)? Yes." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the measurements are made and used to determine the status of the **Activit(y)**ies for managing the allocated requirements. Examples of measurements include status of each of the allocated requirements, changing activity for the allocated requirements and cumulative number of changes to the allocated requirements, including total number of changes proposed, open, approved, and incorporated into the system baseline [28]. The heuristics avobe checks the total number of requirements approved, incorporated, open, and proposed.

4) The limitations of the data and heuristics

None

KPA 1.6 Are the activities for managing allocated requirements on the project subjected to SQA review?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	RM Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			RM Review
Artifact	***Name***$_{set}$, **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)			RM Document

2) The heuristics used to make an assessment based on that data

The fact *RM Review, RM Reviewer,* and *RM Document* are used in the JESS rule. *RR Review* counts the number of **Activit(y)**ies related to managing allocated requirements. *RM Reviewer* counts the number of **Individual**s who are assigned to review requirements management **Activit(y)**ies. *RM Document* counts the number of requirements management related **Artifact**s. The fact-related attributes are gathered automatically by querying attributes from PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies, **Artifact**s related to requirements management, and **Individual**s to review requirements management **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA1-6Yes
  (RM Review, ?x)
  (RM Reviewer, ?y)
  (RM Document, ?z)
  (test (and(<= 1 ?x) (<= 1 ?y) (<= 1 ?z) ))
   =>
  (assert (KPA1-6 Yes))
(printout Result.txt " KPA 1.6 Are the activities for managing allocated requirements on the project subjected to SQA review? Yes." crlf))

(defrule KPA1-6No
  (RM Review, ?x)
```

```
 (RM Reviewer, ?y)
 (RM Document, ?z)
 (test (or(= 0 ?x) (= 0 ?y) (= 0 ?z)  ))
  =>
 (assert (KPA1-6 No))
 (printout Result.txt " KPA 1.6 Are the activities for managing allocated requirements on the project subjected to SQA review? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of requirements management **Activit(y)**ies, **Artifact**s, and the assigned **Individual**s (SQA group) to review requirements management **Activit(y)**ies, and **Artifact**s.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to requirements management and documents related to requirements management should be predefined and recongnized in the knowledge base.

D. Software Project Planning

The purpose of Software Project Planning is to establish reasonable plans for performing the software engineering activities and for managing the software project [28].

KPA 2.1 Are estimates (e.g., size, cost, and schedule) documented for use in planning and tracking the software project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Rework	**Adds, Changes, Deletes**	**Rework** contains attributes of an **Artifact**.	Configuration Management Tool (ClearCase)	Lines of Code
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	Number of Feature
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	Assigned Individual
Organization	***Name***$_{set}$, ***Description***$_{set}$ ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$, ***ComputerResource***$_{set}$, ***ToolCompatability***$_{set}$, ***TrainingFacilities***$_{set}$,			

	CourseReview$_{set}$(**Individual**, software managers) ***Training***$_{set}$(audience, objectives, length, lesson plans), ***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)			
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)			Type of Artifact
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			Actual compare to plan, Task number, Activity Type
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		

FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Lines of Code, Number of Feature, Assigned Individual, Type of Artifact, Activity Type, Actual compare to plan,* and *task_number* are used in the JESS rule. *Lines of Code, Number of Feature, Assigned Individual, Type of Artifact, Activity Type* checks the existence of the attributes related to lines of code, number of **Feature**s, assigned **Individual**, type of **Artifact**s & **Activit(y)**ies. *Actual compare to plan* counts the number of **Activit(y)**ies that have attributes such as **InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0), ***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$, ***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$ ***,ActivityNumber***$_{Calculate}$. *task_number* calculates all the **Activity** number in the project. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of the result of *Lines of Code, Number of Feature, Assigned Individual, Type of Artifact, Activity Type* is more than 1, which means there are attributes related to lines of code, number of **Feature**s, assigned **Individual**, type of **Artifact**s & **Activit(y)**ies. And it also checks the equation result ($\frac{Actual\ compare\ to\ plan}{task\_number}$) is more than 0.9, it means 90 % of the

Activit(y)ies have attributes of **InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0), ***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$, ***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$, and ***ActivityNumber***$_{Calculate}$.

```
(defrule KPA2-1AlmostAlways
  (Lines of Code, ?t)
 (Number of Feature, ?u)
 (Assigned Individual, ?v)
 (Type of Artifact, ?w)
 (Activity Type, ?x)
 (Actual compare to plan, ?y)   (task_number, ?z)
 (test (and (<= 1 ?t) (<= 1 ?u) (<= 1 ?v) (<= 1 ?w) (<= 0.9 (/ ?y ?x))))
  =>
 (assert(KPA2-1 AlmostAlways))
 (printout Result.txt "KPA 2.1 Are estimates (e.g., size, cost, and schedule) documented for use in
planning and tracking the software project? AlmostAlways. " crlf))
 --------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that in order to software estimates are documented for use in planning and tracking the software project, it shold specify size measurements, types of work products and activities, groups and individuals who review and agree to size estimates, project costs, critical computer resources, and software schedule [28]. The heuristics above checks the existence of attributes such as lines of code, number of **Feature**s, assigned **Individual**, cost, type of **Artifact**s & **Activit(y)**ies, **InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0), ***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$, ***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$, and ***ActivityNumber***$_{Calculate}$.

4) The limitations of the data and heuristics

None

KPA 2.2 Do the software plans document the activities to be performed and the commitments made for the software project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
SLCModel	**Type**(waterfall, spiral, single prototype, serial build, Other)		Project Planning Tool (MS Project	Software Life Cycle
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		SPP Reviewer

Organization	***Name***$_{set}$, ***Description***$_{set}$, ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***ComputerResource***$_{set}$			Facility, Tool
Risk	***EstimatedRisk***$_{set}$, ***Description***$_{set}$			Risk
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SPP Document

2) The heuristics used to make an assessment based on that data

The facts *Software Life Cycle, SPP Reviewer, Facility, Tool, Risk,* and *SPP Document* are used in the JESS rule. *Software Life Cycle* checks the existence of software life cycle, *SPP Reviewer* checks the assigned **Individual** to review the document, *Facility* checks the existence of software development facility, *Tool* checks the existence of support tools, *Risk* checks the attributes related to risk and *SSP Document* checks the existence of documents related to software project planning. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of the result of *Software Life Cycle, SPP Reviewer, Facility, Tool, Risk,* and *SPP Document* is more than

1, which means there are attributes related to software life cycle, document, assigned **Individual** to review the document, risk, software development facility and support tools.

```
(defrule KPA2-2InceptionPhaseYes
(Software Life Cycle, ?u) (SPP Reviewer, ?v) (Facility, ?w)
 (Tool, ?x) (Risk, ?y) (SPP Document, ?z)
 (test (and (<= 1 ?u) (<= 1 ?v) (<= 1 ?w) (<= 1 ?x) (<= 1 ?y) (<= 1 ?z)))
  =>
 (assert (KPA2-2 Yes))
 (printout Result.txt "KPA 2.2 Do the software plans document the activities to be performed and the commitments made for the software project? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that in order to make software plans document the activities to be performed and the commitments made for the software project, it shold specify software life cycle, document of software development plan, the document reviewer, risks, and software development facilities and support tools [28]. The heuristics above checks the existence of attributes such as software life cycle, document, assigned **Individual** to review the document, risk, software development facility and support tools.

4) The limitations of the data and heuristics

The document related to software project planning should be predefined and recongnized in the knowledge base.

KPA 2.3 Do all affected groups and individuals agree to their commitments related to the software project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActivityNumber***$_{\text{Calculate}}$	**Activity** is owned by an **Individual**.	Project Planning Tool (MS Project	task_number, NotAssignedTask

2) The heuristics used to make an assessment based on that data

The facts *task_number, NotAssignedTask* are used in the JESS rule. *task_number* counts the number of **Activit(y)**ies in the project. *NotAssignedTask* counts the number of **Activit(y)**ies that are not assigned to **Individual**. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If ($\frac{task\_number - NotAssignedTask}{task\_number}$) is more than 0.9, it means 90 % of the **Activit(y)**ies are assigned to **Individual**s.

```
(defrule KPA2-3AlmostAlways
 (task_number, ?x)
 (NotAssignedTask, ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
 (assert (KPA2-3 AlmostAlways))
 (printout Result.txt "KPA 2.3 Do all affected groups and individuals agree to their commitments related
to the software project? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activity** assignment to **Individual**. If there is any **Individual** for the **Activity**, we can assume that the affected groups and **Individual**s agree to their commitments related to the software project.

4) The limitations of the data and heuristics

None

KPA 2.4 Does the project follow a written organizational policy for planning a software project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SPP Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			SPP Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF,			SPP Activity

	OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SPP Document

2) The heuristics used to make an assessment based on that data

The facts *SPP Group, SPP Tool, SPP Activity,* and *SPP Document* are used in the JESS rule. *SPP Group* counts the number of responsible group or **Individual**s assigned to software project planning. *SPP Activity* counts the number of **Activit(y)**ies related to software project planning. *SPP Tool* counts the number of tools to use for software project planning. *SPP Document* counts the number of documents related to software project planning. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *SPP Group, SPP Tool, SPP Activity,* and *SPP Document* is more than 1, which means software project planning related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA2-4Yes
  (SPP Group, ?w) (SPP Tool, ?x) (SPP Activity, ?y) (SPP Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA2-4 Yes))
  (printout Result.txt " KPA 2.4 Does the project follow a written organizational policy for planning a
software project? Yes." crlf))

(defrule KPA2-4No
  (SPP Group, ?w) (SPP Tool, ?x) (SPP Activity, ?y) (SPP Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA2-4 No))
  (printout Result.txt " KPA 2.4 Does the project follow a written organizational policy for planning a
software project? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to software project planning should be predefined and recongnized in the knowledge base.

KPA 2.5 Are adequate resources provided for planning the software project (e.g., funding and experienced individuals)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActualCost***$_{set}$, ***PlannedCost***$_{set}$		Project Planning Tool (MS Project)	Funds satisfied
	Complete$_{set}$(Yes=1, No=blank or 0)			Number of completed task
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			Sufficient trained Individual, Number of Individual

2) The heuristics used to make an assessment based on that data

The facts *Funds satisfied, Number of completed task, Sufficient trained Individual, Number of Individual* are used in the JESS rule. *Funds satisfied* counts the number of **ActualCost**$_{set}$, which is less than the **PlannedCost**$_{set}$. *Number of completed task* counts the number of completed **Activit(y)**ies. *Sufficient trained Individual* counts the number of **Individual**s who have sufficient experience. *Number of Individual* counts the number of

Individuals in the project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{\text{Funds satisfied}}{\text{Number of completed task}} + \frac{\text{Sufficient trained Individual}}{\text{Number of Individual}}) \times \frac{1}{2}$ is more than 0.9, it means 90 % of the **Activit(y)**ies are provided with sufficient funding and the assigned **Individual**s have sufficient experience.

```
(defrule KPA2-5AlmostAlways
  (Funds satisfied, ?x)
  (Number of completed task, ?y)
  (Sufficient trained Individual, ?z)
  (Number of Individual, ?w)
  (test (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?z ?w)))))
   =>
  (assert (KPA2-5 AlmostAlways))
  (printout Result.txt "KPA 2.5 Are adequate resources provided for planning the software project (e.g.,
funding and experienced individuals)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

There are sufficient and insufficient **Individual** experiences. We need to know the ratio of sufficient and insufficient **Individual** experiences to analyze the **Individual** experience level of the project. And by comparing **ActualCost**$_{set}$ to **PlannedCost**$_{set}$, we can analyze the fund status of a project.

4) The limitations of the data and heuristics

None

KPA 2.6 Are measurements used to determine the status of the activities for planning the software project (e.g., completion of milestones for the project planning activities as compared to the plan)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**Complete**$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project)	WorkCompleted, WorkNotCompleted
	ActivityNumber$_{Calculate}$			TotalWork

2) The heuristics used to make an assessment based on that data

The facts *WorkCompleted, WorkNotCompleted,* and *TotalWork* are used in the JESS rule. *WorkCompleted* counts the number of **Activit(y)**ies completed, and

WorkNotCompleted counts the number of **Activit(y)**ies not completed. *TotalWork* counts the number of **Activit(y)**ies in the project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(WorkCompleted + WorkNotCompleted) = TotalWork$ is correct, which means there are measurements used in the project and the measurement is correct.

```
(defrule KPA2-6Yes
  (WorkCompleted, ?x)
  (WorkNotCompleted, ?y)
  (TotalWork, ?z)
  (test (= ?z  (+ ?x  ?y)))
   =>
  (assert(KPA2-6 Yes))
  (printout Result.txt "KPA 2.6 Are measurements used to determine the status of the activities for planning the software project (e.g., completion of milestones for the project planning activities as compared to the plan)? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

If the sum of **Activit(y)**ies from the facts *WorkCompleted,* and *WorkNotCompleted,* are equal to the number of *TotalWork* , it means the measurements are used to determine the status of the activities for planning the software project and the measurements are used correctly.

4) The limitations of the data and heuristics

None

KPA 2.7 Does the project manager review the activities for planning the software project on both a periodic and event-driven basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	SPP Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0),			SPP Review

	Type$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The fact *SPP Review, SPP Reviewer* are used in the JESS rule. *SPP Review* counts the number of review **Activit(y)**ies for planning the software project. *SPP Definiton Reviewer* counts the number of **Individual**s who are assigned to review **Activit(y)**ies for planning the software project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any review **Activit(y)**ies for planning the software project and **Individual**s to review planning the software project **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA2-7Yes
 (SPP, ?x) (SPP, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
 (assert (KPA2-7 Yes))
```

```
 (printout Result.txt " KPA 2.7 Does the project manager review the activities for planning the software project on both a periodic and event-driven basis? Yes." crlf))

(defrule KPA2-7No
 (SPP, ?x) (SPP, ?y)
 (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
 (assert (KPA2-7 No))
 (printout Result.txt " KPA 2.7 Does the project manager review the activities for planning the software project on both a periodic and event-driven basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of review **Activit(y)**ies for planning the software project and the assigned **Individual**s (project manager) to review **Activit(y)**ies for planning the software project.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Softwre Project Planning should be predefined and recongnized in the knowledge base.

E. KPA 3. Software Project Tracking and Oversight

The purpose of Software Project Tracking and Oversight is to provide adequate visibility into actual progress so that management can take corrective actions when the software project's performance deviates significantly from the software plans [28].

KPA 3.1 Are the project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project)	Actual compare to plan
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		
Activity	***ActivityNumber***$_{Calculate}$			TotalWork

2) The heuristics used to make an assessment based on that data

The facts *Actual compare to plan, TotalWork* are used in the JESS rule. *Actual compare to plan* counts the number of **Activit(y)**ies that have attributes such as **InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0), ***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set,}$ ***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$ ***,ActivityNumber***$_{Calculate}$. *TatalWork* calculates all the **Activity** number in the project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{Actual\ \text{compare to plan}}{TotalWork})$ is more than 0.9, it means 90 % of the **Activit(y)**ies have attributes of **InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0), ***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set,}$ ***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$ ***,ActivityNumber***$_{Calculate}$.

```
(defrule KPA3-1AlmostAlways
  (Actual compare to plan, ?x) (TotalWork, ?y)
  (test (<= 0.9 (/ ?x ?y)))
   =>
  (assert (KPA3-1 AlmostAlways))
  (printout Result.txt "KPA 3.1 Are the project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model stats that the software planning data, re-planning data, and the actual measurement data are archived for use by ongoing and future projects [28]. The heuristics above shows the existence of above data and the actual start date, finish date, and costs are compared with estimates in the software plans.

4) The limitations of the data and heuristics

None

3.2 Is corrective action taken when actual results deviate significantly from the project's software plans?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project	DeviatedActivityFromPlan
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
Final-	***PlannedEndDate***$_{set}$,	**FinalMilestone**		

Milstone	***ActualEndDate***$_{set}$	is an attribute of **Activity**.		
Activity	**InitialMilestone**, **FinalMilestone**, ***Complete***$_{set}$(Yes=1, No=blank or 0)			DeviatedActivityFrom UpdatedPlan
Initial-Milestone	***UpdatedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***UpdatedCost***$_{set}$, ***ActualCost***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
Final-Milstone	***UpdatedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *DeviatedActivityFromPlan, DeviatedActivityFromUpdatedPlan* are used in the JESS rule. *DeviatedActivityFromPlan* counts the number of late **Activit(y)**ies deviated from **Plan**. *DeviatedActivityFromUpdatedPlan* counts the number of the late **Activit(y)**ies deviated from updated **Plan**. Updated **Plan** is modified one from the original **Plan**. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If *DeviatedActivityFromPlan* – *DeviatedActivityFromUpdatedPlan* is more than

0, which means there is more significant deviation from **Plan** than Updated **Plan** and it proves that Updated **Plan** is a modified one for the less deviation.

```
(defrule KPA3-2Yes
  (DeviatedActivityFromPlan, ?x) (DeviatedActivityFromUpdatedPlan, ?y)
  (test (< 0 (- ?x ?y)))
   =>
  (assert (KPA3-2 Yes))
  (printout Result.txt "KPA 3.2 Is corrective action taken when actual results deviate significantly from the project's software plans? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

If there is significant deviation from **Plan**, there should be a new baseline. The heuristics above checks actual, planned, and updated schedule and cost. If the actual data is significantly deviated from planned start date, the **Plan** should be modified to make an Updated **Plan**, which has less deviation.

4) The limitations of the data and heuristics

None

KPA 3.3 Are changes in the software commitments agreed to by all affected groups and individuals?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActivityNumber***$_{\text{Calculate}}$		Project Planning Tool (MS Project)	task_number
		Activity is owned by an **Individual**.		NotAssignedTask

2) The heuristics used to make an assessment based on that data

The facts *task_number, NotAssigned* are used in the JESS rule. *task_number* counts the number of **Activit(y)**ies in the project. *NotAssignedTask* counts the number of **Activit(y)**ies that is not assigned to **Individual**s. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{task\_number - NotAssignedTask}{task\_number})$ is more than 0.9, it means 90 % of the **Activit(y)**ies are assigned to **Individual**s.

```
(defrule KPA3-3AlmostAlways
  (task_number, ?x)
  (NotAssignedTask, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
```

(assert (KPA3-3 AlmostAlways))

(printout Result.txt "KPA 3.3 Are changes in the software commitments agreed to by all affected groups and individuals? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activity** assignment to **Individual**. If there is any **Individual** for the **Activity**, we can assume that affected groups and **Individual**s agree to changes in the software commitments.

4) The limitations of the data and heuristics

None

KPA 3.4 Does the project follow a written organizational policy for both tracking and controlling its software development activities?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Organization	***Name***set***, Description***set ,***Type***set(W<u>ord Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)</u>		Project Planning Tool (MS Project)	SPTO Group
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***set			SPTO Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***set**,** ***Description***set**,** **Complete**set(Yes=1, No=blank or 0), **ActualCost**set, **PlannedCost**set, ***Type***set(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM</u>)			SPTO Activity
Artifact	***Name***set **Type**(RequirementsFile, DesignFile, DocumentFile,	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SPTO Document

	SourceFile, Other)			

2) The heuristics used to make an assessment based on that data

The facts *SPTO Group, SPTO Tool, SPTO Activity,* and *SPTO Document* are used in the JESS rule. *SPTO Group* counts the number of responsible group or **Individual**s assigned to tracking and controlling software development. *SPTO Activity* counts the number of **Activit(y)**ies related to tracking and controlling software development. *SPTO Tool* counts the number of tools to use for tracking and controlling software development **Activit(y)**ies. *SPTO Document* counts the number of documents related to tracking and controlling software development **Activit(y)**ies. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *SPTO Group, SPTO Tool, SPTO Activity,* and *SPTO Document* is more than 1, which means tracking and controlling software development related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA3-4Yes
  (SPTO Group, ?w) (SPTO Tool, ?x) (SPTO Activity, ?y) (SPTO Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
```

```
  =>
  (assert (KPA3-4 Yes))
  (printout Result.txt " KPA 3.4 Does the project follow a written organizational policy for both tracking
and controlling its software development activities? Yes." crlf))

(defrule KPA3-4No
  (SPTO Group, ?w) (SPTO Tool, ?x) (SPTO Activity, ?y) (SPTO Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
  (assert (KPA3-4 No))
  (printout Result.txt " KPA 3.4 Does the project follow a written organizational policy for both tracking
and controlling its software development activities? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Project Tracking and Oversight should be predefined and recongnized in the knowledge base.

KPA 3.5 Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual		**Individual** owns **Activit(y)**ies	Project Planning Tool (MS Project)	ProjectReviewResponsibility, ProjectReviewNotResponsibility
Activity	**InitialMilestone, FinalMilestone, *Name***$_{set}$**, Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The facts *ProjectReviewResponsibility,* and *ProjectReviewNotResponsibility* are used in the JESS rule. *ProjectReviewResponsibility* counts the number of resource assignment to **Activit(y)**ies related to tracking software work products and **Activit(y)**ies.

ProjectReviewNotResponsibility counts the number of above **Activit(y)**ies with no **Individual**s assigned. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(ProjectReviewResponsibility - ProjectReviewNotResponsibility)}{ProjectReviewResponsibility}$ is more than 0.9, it means 90 % of the project is assigned.

```
(defrule KPA3-5AlmostAlways
  (ProjectReviewResponsibility, ?x) (ProjectReviewNotResponsibility, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA3-5 AlmostAlways))
  (printout Result.txt "KPA 3.5 Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)? AlmostAlways." crlf))

(defrule KPA3-5Frequently
  (ProjectReviewResponsibility, ?x) (ProjectReviewNotResponsibility, ?y)
  (test (<= 0.6 (/ (- ?x ?y) ?x)))
  (test (>  0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA3-5 Frequently))
  (printout Result.txt "KPA 3.5 Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)? Frequently." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activit(y)**ies related to tracking project and the assignment of **Individual**s to these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 3.6 Are measurements used to determine the status of the activities for software tracking and oversight (e.g., total effort expended in performing tracking and oversight acticvities)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM,		Project Planning Tool (MS Project)	Software Tracking and Oversight, Bad Software Tracking and Oversight

	DP, TCM, PCM), Quantitative Process Management, Software Quality Management, Technology Change Management), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Software Tracking and Oversight* and *Bad Software Tracking and Oversight* are used in the JESS rule. *Software Tracking and Oversight* counts the number of **Activit(y)**ies related to software tracking and oversight. *Bad Software Tracking and Oversight* counts the number of software tracking and oversight **Activit(y)**ies, which do not have proper attributes of schedule and cost. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{Software Tracking and Oversight} - \textit{Bad}\text{Software Tracking and Oversight})}{\text{Software Tracking and Oversight}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA3-6AlmostAlways
  (Software Tracking and Oversight, ?x)
  (Bad Software Tracking and Oversight, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA3-6 AlmostAlways))
  (printout Result.txt " KPA 3.6 Are measurements used to determine the status of the activities for software tracking and oversight (e.g., total effort expended in performing tracking and oversight acticvities)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to software tracking and oversight and the attributes of schedule and the cost of software tracking and oversight **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 3.7 Are the activities for software project tracking and oversight reviewed with senior management on a periodic basis (e.g., project performance, open issues, risks, and action items)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	SPTO Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			SPTO Review

2) The heuristics used to make an assessment based on that data

The fact *SPTO Review, SPTO Reviewer* are used in the JESS rule. *SPTO Review* counts the number of **Activit(y)**ies related to software project tracking and oversight review. *SPTO Reviewer* counts the number of **Individual**s who are assigned to review software project tracking and oversight **Activit(y)**ies. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software project tracking and oversight review and **Individual**s to review software project tracking and oversight **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA3-7Yes
 (SPTO Review, ?x)
 (SPTO Reviewer, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
 (assert (KPA3-7 Yes))
 (printout Result.txt " KPA 3.7 Are the activities for software project tracking and oversight reviewed with senior management on a periodic basis (e.g., project performance, open issues, risks, and action items)? Yes." crlf))

(defrule KPA3-7No
```

```
 (SPTO Review, ?x)
 (SPTO Reviewer, ?y)
 (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
 (assert (KPA3-7 No))
 (printout Result.txt " KPA 3.7 Are the activities for software project tracking and oversight reviewed with
senior management on a periodic basis (e.g., project performance, open issues, risks, and action items)?
No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of software project tracking and oversight review **Activit(y)**ies and the assigned **Individual**s (SQA group) to review software project tracking and oversight **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Software Project Tracking and Oversight should be predefined and recongnized in the knowledge base.

F. KPA 4. Software Subcontract Management

The purpose of Subcontract Management is to select qualified software subcontractors and manage them effectively [28].

KPA 4.1 Is a documented procedure used for selecting subcontractors based on their ability to perform the work?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Supplier	***Name***$_{Set}$, ***Description***$_{Set,}$ ***Experience***$_{Set}$ (software development years)		Project Planning Tool (MS Project	Subcontractor, SubcontractorAbility
Artifact	***Name***$_{set}$, **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SubcontractorDocument

2) The heuristics used to make an assessment based on that data

The facts *Subcontractor, SubcontractorAbility,* and *SubcontractorDocument* are used in the JESS rule. *Subcontractor* counts the number of **Supplier**s. *SubcontractorAbility* checks the number of subcontractors with "Very Good" ability. *SubcontractorDocument* checks the existence of document, Software Development Plan, which includes subcontractor management plan. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $1-(\frac{Subcontractor - SubcontractorAbility}{Subcontractor})$ is more than 0.6, it means 60 % of the subcontractor's ability is very good. And it checks the existence of Software Development Plan.

```
(defrule KPA4-1Yes
  (Subcontractor, ?x) (SubcontractorAbility, ?y) (SubcontractorDocument, ?z)
  (test (and (= 1 ?z) (<= 0.6 (- 1 (/ (- ?x ?y) ?x))) ))
   =>
  (assert (KPA4-1 Yes))
  (printout Result.txt "KPA 4.1 Is a documented procedure used for selecting subcontractors based on their
ability to perform the work? Yes." crlf))

(defrule KPA4-1No
  (Subcontractor, ?x) (SubcontractorAbility, ?y) (SubcontractorDocument, ?z)
  (test (or (= 0 ?z) (> 0.6 ( - 1 (/ (- ?x ?y) ?x))) ))
   =>
  (assert (KPA4-1 No))
  (printout Result.txt "KPA 4.1 Is a documented procedure used for selecting subcontractors based on their
ability to perform the work? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of document, Software Development Plan, which includes subcontractor management plan and the subcontractor's ability. If we find more than 60% of the subcontractors have good ability and there is a document stating selecting subcontractor, we can assume that there is a documented procedure used for selecting subcontractors based on their ability to perform the work.

4) The limitations of the data and heuristics

The documents related to Software Subcontract Management should be predefined and recongnized in the knowledge base.

KPA 4.2 Are changes to subcontracts made with the agreement of both the prime contractor and the subcontractor?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)	**Subcontract Activity** is owned by a **Supplier**.	Project Planning Tool (MS Project)	Subcontract task number, Subcontract Not AssignedTask

2) The heuristics used to make an assessment based on that data

The facts *Subcontracttask_number, SubcontractNotAssignedTask* are used in the JESS rule. *Subcontracttask_number* counts the number of **Activit(y)**ies will be implemented by a supplier. *SubcontractNotAssignedTask* counts the number of supplier **Activit(y)**ies that is not assigned to **Supplier**. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{Subcontracttask\_number - SubcontractNotAssignedTask}{Subcontracttask\_number})$ is more than 0.9, it means 90 % of the **Activit(y)**ies belong to **Supplier** are properly assigned to Subcontractors.

```
(defrule KPA4-2AlmostAlways
 (Subcontracttask_number, ?x) (SubcontractNotAssignedTask, ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
```

```
  =>
  (assert (KPA4-2 AlmostAlways))
  (printout Result.txt "KPA 4.2 Are changes to subcontracts made with the agreement of both the prime contractor and the subcontractor? AlmostAlways." crlf))

(defrule KPA4-2Frequently
  (Subcontracttask_number, ?x) (SubcontractNotAssignedTask, ?y)
  (test (<= 0.6 (/ (- ?x ?y) ?x)))
  (test (>  0.9 (/ (- ?x ?y) ?x)))
  =>
  (assert (KPA4-2 Frequently))
  (printout Result.txt "KPA 4.2 Are changes to subcontracts made with the agreement of both the prime contractor and the subcontractor? Frequently." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the assignment status. If there is a subcontractor who would implement the task, we can assume that there is an agreement.

4) The limitations of the data and heuristics

None

KPA 4.3 Are periodic technical interchanges held with subcontractors?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Supplier	***Name***$_{\text{Set}}$, ***Description***$_{\text{Set}}$		Project Planning Tool (MS Project)	Subcontractor
	TechnicalInterchange$_{\text{Set}}$			Subcontract Not Technical Interchange

2) The heuristics used to make an assessment based on that data

The facts *Subcontractor, SubcontractNotTechnicalInterchange* are used in the JESS rule. *Subcontractor* counts the number of **Supplier**s. *SubcontractNotTechnicalInterchange* counts the number of **TechnicalInterchange** that does not change technical information. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{Subcontractor - SubcontractNotTechnicalInterchange}{Subcontractor})$ is more than 0.9, it means 90 % of the **Supplier**s change technical information well.

```
(defrule KPA4-3AlmostAlways
 (Subcontractor, ?x) (SubcontractNotTechnicalInterchange, ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
```

(assert (KPA4-3 AlmostAlways))

(printout Result.txt "KPA 4.3 Are periodic technical interchanges held with subcontractors? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the status of technical interchange whether it is doing well.

4) The limitations of the data and heuristics

None

KPA 4.4 Are the results and performance of the software subcontractor tracked against their commitments?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM,		Project Planning Tool (MS Project)	Subcontracttask number, Subcontracttrack number

	SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The facts *Subcontracttask_number, Subcontracttrack_number* are used in the JESS rule. *Subcontracttask_number* counts the number of **Activit(y)**ies that will be implemented by a supplier. *Subcontracttrack_number* counts the number of supplier **Activit(y)**ies that have attributes such as percent complete, start, and finish date. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result $1-(\frac{Subcontracttask\_number - Subcontracttrack\_number}{Subcontracttrack\_number})$ is more than 0.9, it means 90 % of the **Activit(y)**ies belong to **Supplier** are properly tracked.

```
(defrule KPA4-4AlmostAlways
 (Subcontracttask_number, ?x)
 (Subcontracttrack_number, ?y)
 (test (<= 0.9  (- 1 (/ (- ?x ?y) ?x))))
  =>
```

(assert (KPA4-4 AlmostAlways))

(printout Result.txt "KPA 4.4 Are the results and performance of the software subcontractor tracked against their commitments? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

In order to track subcontractor **Activity**, we need to know the existence of subcontractor **Activity** attributes such as **InitialMilestone, FinalMilestone, Complete**$_{set}$.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Subcontract Management should be predefined and recongnized in the knowledge base.

KPA 4.5 Does the project follow a written organizational policy for managing software subcontracts?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SSM Group

Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			SSM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			SSM Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SSM Document

2) The heuristics used to make an assessment based on that data

The facts *SSM Group, SSM Tool, SSM Activity,* and *SSM Document* are used in the JESS rule. *SSM Group* counts the number of responsible group or **Individual**s assigned to software subcontract management. *SSM Activity* counts the number of **Activit(y)**ies related to software subcontract management. *SSM Tool* counts the number of tools to use for software subcontract management. *SSM Document* counts the number of documents related to software subcontract management. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *SSM Group, SSM Tool, SSM Activity,* and *SSM Document* is more than 1, which means software subcontract management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA4-5Yes
  (SSM Group, ?w) (SSM Tool, ?x) (SSM Activity, ?y) (SSM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA4-5 Yes))
  (printout Result.txt " KPA4.5 Does the project follow a written organizational policy for managing
software subcontracts? Yes." crlf))

(defrule KPA4-5No
  (SSM Group, ?w) (SSM Tool, ?x) (SSM Activity, ?y) (SSM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA4-5 No))
```

(printout Result.txt " KPA4.5 Does the project follow a written organizational policy for managing software subcontracts? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Subcontract Management should be predefined and recongnized in the knowledge base.

KPA4.6 Are the people responsible for managing software subcontracts trained in managing software subcontract?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Supplier	***Name***$_{Set}$, ***Description***$_{Set}$ ***Experience***$_{set}$(Software development years)		Project Planning Tool (MS	Subcontractor, SubcontractTraining

			Project)	

2) The heuristics used to make an assessment based on that data

The facts *Subcontractor, SubcontractTraining* are used in the JESS rule. *Subcontractor* counts the number of contract manager. *SubcontractTraining* counts the number of subcontract manager with training experience. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $1-(\frac{Subcontractor - SubcontractTraining}{Subcontractor})$ is more than 0.9, it means 90 % of the subcontractor managers have sufficient training.

```
(defrule KPA4-6AlmostAlways
 (Subcontractor, ?x) (SubcontractTraining, ?y)
 (test (<= 0.9  (- 1 (/ (- ?x ?y) ?x))))
  =>
 (assert (KPA4-6 AlmostAlways))
 (printout Result.txt "KPA 4.6 Are the people responsible for managing software subcontracts trained in managing software subcontracts? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of subcontract manager and the training experience status.

4) The limitations of the data and heuristics

None

KPA 4.7 Are measurements used to determine the status of the activities for managing software subcontracts (e.g., schedule status with respect to planned delivery dates and effort expended for managing the subcontract)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), **InitialMilestone, FinalMilestone,** ***Name***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project)	Subcontracttask number, PlannedDeliveryDate, SubcontractActivity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile,	**Artifact**s are related to **Feature**s	Configuration Management Tool	PlanDocument

	DocumentFile, SourceFile, Other)		(ClearCase)	

2) The heuristics used to make an assessment based on that data

The facts *Subcontracttask_number, PlannedDeliveryDate, PlanDocument,* and *SubcontractActivity* are used in the JESS rule. Subcontracttask_number counts the number of subcontract **Activity**. *PlannedDeliverDate* checks the number of subcontract **Activit(y)**ies with schedule status. *PlanDocument* counts the number of document, Software Development Plan. *SubcontractActivity* counts the number of **Activit(y)**ies such as "Define project organization and staffing", "Define Monitoring & Control Processes", "Plan Phases and Iterations", and "Compile Software Development Plan". Find these activities create an artifact "Software Development Plan". These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $1-(\frac{Subcontracttask\_number - PlannedDeliveryDate}{Subcontracttask\_number})$ is more than 0.6, which means more than 60 % of subcontract **Activit(y)**ies have schedule status. If there is a document, Software Development Plan and related Activit(y)ies in Inception Phase, we can assume that measurements are used to determine the status of the activities for managing software subcontracts.

(defrule KPA4-7InceptionPhaseYes

```
  (LifeCycle InceptionPhase )
  (Subcontracttask_number, ?x)
  (PlannedDeliveryDate, ?y)
  (PlanDocument, ?z) (SubcontractActivity, ?v)
  (test (and (<= 0.6 (- 1 (/ (- ?x ?y) ?x))) (= 1 ?z) (< 0 ?v) ))
   =>
  (assert (KPA4-7 Yes))
  (printout Result.txt "KPA 4.7 Are measurements used to determine the status of the activities for managing software subcontracts (e.g., schedule status with respect to planned delivery dates and effort expended for managing the subcontract)? Yes." crlf))

(defrule KPA4-7InceptionPhaseNo
  (LifeCycle InceptionPhase )
  (Subcontracttask_number, ?x)
  (PlannedDeliveryDate, ?y)
  (PlanDocument, ?z) (SubcontractActivity, ?v)
  (test (or (> 0.6 (- 1 (/ (- ?x ?y) ?x))) (= 0 ?z) (= 0 ?v) ))
   =>
  (assert (KPA4-7 No))
  (printout Result.txt "KPA 4.7 Are measurements used to determine the status of the activities for managing software subcontracts (e.g., schedule status with respect to planned delivery dates and effort expended for managing the subcontract)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of actual, planned delivery dates of Subcontract **Activity** and **Activit(y)**ies related to the effort expended for managing the subcontract.

4) The limitations of the data and heuristics

The **Activity** "Define project organization and staffing", "Define Monitoring & Control Processes", "Plan Phases and Iterations", and "Compile Software Development Plan". And the **Artifact,** Software Development Plan only exist in the RUP. If we do not use the RUP, we need to create the **Activity** and the **Artifact**.

KPA 4.8 Are the software subcontract activities reviewed with the project manager on both a periodic and event-driven basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	SSM Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$,			SSM Review

	Complete$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The fact *SSM Review, SSM Reviewer* are used in the JESS rule. *SSM Review* counts the number of **Activit(y)**ies related to software subcontract management review. *SSM Reviewer* counts the number of **Individual**s who are assigned to review software subcontract management **Activit(y)**ies. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software subcontract management review and **Individual**s to review software subcontract management **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA4-8Yes
 (SSM Review, ?x) (SSM Reviewer, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
```

```
 (assert (KPA4-8 Yes))
 (printout Result.txt " KPA 4.8 Are the software subcontract activities reviewed with the project manager on both a periodic and event-driven basis? Yes." crlf))

(defrule KPA4-8No
 (SSM Review, ?x) (SSM Reviewer, ?y)
 (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
 (assert (KPA4-8 No))
 (printout Result.txt " KPA 4.8 Are the software subcontract activities reviewed with the project manager on both a periodic and event-driven basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of training program review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review software subcontract management **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Software Subcontract Management should be predefined and recongnized in the knowledge base.

F. KPA 5. Software Quality Assurance

The purpose of Software Quality Assurance (SQA) is to provide management with appropriate visibility into the process being used by the software project and of the products being built [28].

KPA 5.1 Are SQA activities planned?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Plan (ClearCase)	SQADocument

2) The heuristics used to make an assessment based on that data

The fact *SQADocument* is used in JESS rule. *SQADocument* counts the number of **Artifact**s such as Quality Assurance Plan. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Artifact**s related to Quality Assurance Plan. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA5-1Yes
```

```
 (SQADocument, ?x)
 (test (= 1 ?x) )
  =>
 (assert (KPA5-1 Yes))
 (printout Result.txt " KPA 5.1 Are SQA activities planned? Yes." crlf))

(defrule KPA5-1No
 (SQADocument, ?x)
 (test (= 0 ?x) )
  =>
 (assert (KPA5-1 No))
 (printout Result.txt " KPA 5.1 Are SQA activities planned? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of document, Quality Assurance Plan.

4) The limitations of the data and heuristics

The documents related to Software Quality Assurance should be predefined and recongnized in the knowledge base.

KPA 5.2 Does SQA provide objective verification that software products and activities adhere to applicable standards, procedures, and requirements?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Feature	***Name***set, ***Status***set(Proposed, Approved, Incorporated, Validated)	**Feature**s are related to **Artifact**s.	RequisitePro	SQArequirement
Artifact	***Name***set,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	ClearCase	SQAartifact

2) The heuristics used to make an assessment based on that data

The facts *SQArequirement, SQAartifact* are used in the JESS rule. *SQArequirement* counts the number of requirement, which status is incorporated or validated. *SQAartifact* counts the number of **Artifact**s associated with the requirement, which status is incorporated or validated. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $1-(\frac{SQArequirement - SQAartifact}{SQAartifact})$ is more than 0.9, it means 90 % of the software products adhere to requirements.

```
(defrule KPA5-2AlmostAlways
  (SQArequirement, ?x) (SQAartifact, ?y)
  (test (<= 0.9  (- 1 (/ (- ?x ?y) ?x))))
```

```
=>
(assert (KPA5-2 AlmostAlways))
(printout Result.txt "KPA 5.2 Does SQA provide objective verification that software products and activities adhere to applicable standards, procedures, and requirements? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the ratio of requirements with the status of incorporated, validated to **Artifact**s. Requirement, which status is incorporated or validated, should have associated **Artifact**s.

4) The limitations of the data and heuristics

None

5.3 Are the results of SQA reviews and audits provided to affected groups and individuals (e.g., those who performed the work and those who are responsible for the work)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,** **FinalMilestone,**		Project Planning	Funds extended at Inception Phase,

	Name$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$		Tool (MS Project	Funds extended at Elaboration Phase, Funds extended at Construction Phase First Iteration, Funds extended at Construction Phase Second Iteration, Funds extended at Transition Phase

2) The heuristics used to make an assessment based on that data

The facts *Funds extended at Inception Phase, Funds extended at Elaboration Phase, Funds extended at Construction Phase First Iteration, Funds extended at Construction Phase Second Iteration, Funds extended at Transition Phase* are used in the JESS rule. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort. These facts check fund status by comparing **ActualCost**$_{set}$, **PlannedCost**$_{set}$ at the end of each phase. If **PlannedCost**$_{set}$ is bigger than **ActualCost**$_{set}$, the result is 1, which means funds extended in that phase.

(defrule KPA5-3

(Funds extended at Inception Phase, ?x)

(Funds extended at Elaboration Phase, ?y)

(Funds extended at Construction Phase First Iteration, ?z)

```
(Funds extended at Construction Phase Second Iteration, ?w)
(Funds extended at Transition Phase, ?v)
(test (or (= 1 ?x ) (= 1 ?y ) (= 1 ?z ) (= 1 ?w ) (= 1 ?v )))
=>
(printout Result.txt "KPA 5.3 Are the results of SAQ reviews and audits provided to affected groups and individuals (e.g., those who performed the work and those who are responsible for the work)? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the project status such as fund by the affected groups and individual.

4) The limitations of the data and heuristics

None

KPA 5.4 Are issues of noncompliance that are not resolved within the software project addressed by senior management (e.g., deviations from applicable standards)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$		Project Planning Tool (MS Project)	Funds extended at Inception Phase, Funds extended at Elaboration Phase, Funds extended at Construction Phase First Iteration, Funds extended at Construction Phase Second Iteration, Funds extended at Transition Phase

2) The heuristics used to make an assessment based on that data

The facts *Funds extended at Inception Phase, Funds extended at Elaboration Phase, Funds extended at Construction Phase First Iteration, Funds extended at Construction Phase Second Iteration, Funds extended at Transition Phase* are used in the JESS rule. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort. These facts check fund status by comparing **ActualCost**$_{set}$, **PlannedCost**$_{set}$ at the end of each phase. If **PlannedCost**$_{set}$ is bigger than **ActualCost**$_{set}$, the result is 1, which means funds extended in that phase.

```
(defrule KPA5-4
  (Funds extended at Inception Phase, ?x)
```

```
(Funds extended at Elaboration Phase, ?y)
(Funds extended at Construction Phase First Iteration, ?z)
(Funds extended at Construction Phase Second Iteration, ?w)
(Funds extended at Transition Phase, ?v)
(test (or (= 1 ?x ) (= 1 ?y ) (= 1 ?z ) (= 1 ?w ) (= 1 ?v )))
 =>
(printout Result.txt "KPA 5.4 Are issues of noncompliance that are not resolved within the software project addressed by senior management (e.g., deviations from applicable standards)? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

If there is schedule or fund problem, that should be solved by senior management. The heuristics checks if there is any notification like this.

4) The limitations of the data and heuristics

None

KPA 5.5 Does the project follow a written organizational policy for implementing SQA?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SQA Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			SQA Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			SQA Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile,	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SQA Document

	SourceFile, Other)			

2) The heuristics used to make an assessment based on that data

The facts *SQA Group, SQA Tool, SQA Activity,* and *SQA Document* are used in the JESS rule. *SQA Group* counts the number of responsible group or **Individual**s assigned to implementing software quality assurance. *SQA Activity* counts the number of **Activit(y)**ies related to implementing software quality assurance. *SQA Tool* counts the number of tools to use for implementing software quality assurance. *SQA Document* counts the number of documents related to implementing software quality assurance. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *SQA Group, SQA Tool, SQA Activity,* and *SQA Document* is more than 1, which means software quality assurance group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA5-5Yes
  (SQA Group, ?w) (SQA Tool, ?x) (SQA Activity, ?y) (SQA Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA5-5 Yes))
```

```
 (printout Result.txt " KPA 5.5 Does the project follow a written organizational policy for implementing
SQA?  Yes." crlf))

(defrule KPA5-5No
   (SQA Group, ?w) (SQA Tool, ?x) (SQA Activity, ?y) (SQA Document, ?z)
    (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA5-5 No))
 (printout Result.txt " KPA 5.5 Does the project follow a written organizational policy for implementing
SQA?  No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Qaulity Assurance should be predefined and recongnized in the knowledge base.

KPA 5.6 Are adequate resources provided for performing SQA activities (e.g., funding and a designated manager who will receive and act on software noncompliance items)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActualCost***$_{set}$, ***PlannedCost***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), **Complete**$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project)	Funds satisfied SQA Activity, Number of completed SQA task
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			Sufficient trained SQA Manager, Number of SQA Manager

2) The heuristics used to make an assessment based on that data

The facts *Funds satisfied SQA Activity, Number of completed SQA task, Sufficient trained SQA Manager, Number of SQA Manager* are used in the JESS rule. *Funds satisfied SQA Activity* counts the number of **ActualCost**$_{set}$, which is less than the **PlannedCost**$_{set}$ in SQA related **Activit(y)**ies. *Number of completed SQA task* counts the number of completed SQA related **Activit(y)**ies. *Sufficient trained SQA Manager* counts the number of SQA managers with sufficient experience. *Number of SQA Manger* counts the number of SQA Managers in the project. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{\text{Funds satisfied SQA Activity}}{\text{Number of completed SQA task}} + \frac{\text{Sufficient trained SQA Manager}}{\text{Number of SQA Manager}}) \times \frac{1}{2}$ is more than 0.9, it means 90 % of the SQA related **Activit(y)**ies are provided with sufficient funding and the assigned SQA Managers have sufficient experience.

```
(defrule KPA5-6AlmostAlways
  (Funds satisfied SQA Activity, ?x)
  (Number of completed SQA task, ?y)
  (Sufficient trained SQA Manager, ?z)
  (Number of SQA Manager, ?w)
  (test (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?z ?w)))))
   =>
  (assert (KPA5-6 AlmostAlways))
  (printout Result.txt "KPA 5.6 Are adequate resources provided for performing SQA activities (e.g., funding and a designated manager who will receive and act on software noncompliance items)? AlmostAlways" crlf))
----------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

There are sufficient and insufficient SQA Managers experiences. We need to know the ratio of sufficient and insufficient SQA Managers experiences to analyze the SQA

Manager experience level of the project. By comparing **ActualCost**$_{set}$ to **PlannedCost**$_{set}$ in SQA related **Activit(y)**ies, we can analyze the fund status of a project.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Quality Assurance should be predefined and recongnized in the knowledge base.

KPA 5.7 Are measurements used to determine the cost and schedule status of the activities performed for SQA (e.g., work completed, effort and funds expended compared to the plan)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), **InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$,		Project Planning Tool (MS Project)	SQA Activity, Bad SQA Activity

	PlannedCost$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *SQA Activity,* and *Bad SQA Activity* are used in the JESS rule. *SQA Activity* counts the number of **Activit(y)**ies related to SQA **Activit(y)**ies. *Bad SQA Activity* counts the number of **Activit(y)**ies, which do not have proper attributes of **InitialMilestone, FinalMilestone, *Name***$_{set}$**, Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result $\frac{(SQAActivity - BadSQAActivity)}{SQAActivity}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA5-7AlmostAlways
  (SQAActivity, ?x) (BadSQAActivity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
```

=>

(assert (KPA5-7 AlmostAlways))

(printout Result.txt "KPA 5.7 Are measurements used to determine the cost and schedule status of the activities performed for SQA (e.g., work completed, effort and funds expended compared to the plan)? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to SQA and compares actual effort and funds expended to these **Activit(y)**ies compared to the plan.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Quality Assurance should be predefined and recongnized in the knowledge base.

KPA 5.8 Are activities for SQA reviewed with senior management on a periodic basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	SQA Reviewer
Activity	**InitialMilestone**, **FinalMilestone**,			SQA Review

	Nameset, ***Description***set, **Complete**set(Yes=1, No=blank or 0), ***Type***set(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The facts *SQA Review, SQA Reviewer* are used in the JESS rule. *SQA Review* counts the number of **Activit(y)**ies related to SQA review. *SQA Reviewer* counts the number of **Individual**s who are assigned to review SQA **Activit(y)**ies. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to SQA review and assingemnet of senior manager to review SQA **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA5-8Yes
 (SQA Review, ?x) (SQA Reviewer, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
 (assert (KPA5-8 Yes))
 (printout Result.txt "KPA 5.8 Are activities for SQA reviewed with senior management on a periodic basis? Yes." crlf))
```

```
(defrule KPA5-8No
  (SQA Review, ?x) (SQA Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA5-8 No))
  (printout Result.txt "KPA 5.8 Are activities for SQA reviewed with senior management on a periodic basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of SQA review **Activit(y)**ies and the assignement of **Individual**s to review SQA review **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Software Quality Assurance should be predefined and recongnized in the knowledge base.

G. KPA 6. Software Configuration Management

The purpose of Software Configuration Management (SCM) is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle [28].

KPA 6.1 Are software configuration management activities planned for the project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$, **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Tool (ClearCase)	Configuration Document

2) The heuristics used to make an assessment based on that data

The fact *ConfigurationDocument* is used in the JESS rule. *ConfigurationDocument* counts the number of documents related to configuration management plan. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the *ConfigurationDocument* is more than 1, which means the document related to configuration management plan is exist.

```
(defrule KPA6-1Yes
 (ConfigurationDocument, ?x)
 (test (= 1 ?x) )
  =>
 (assert (KPA6-1 Yes))
```

```
  (printout Result.txt " KPA 6.1 Are software configuration management activities planned for the project?
Yes." crlf))

(defrule KPA6-1No
  (ConfigurationDocument, ?x)
  (test (= 0 ?x) )
   =>
  (assert (KPA6-1 No))
  (printout Result.txt " KPA 6.1 Are software configuration management activities planned for the project?
No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document related to configuration management plan describes the policy for implementing software configuration management **Activit(y)**ies. By checking the existence of configuration management related document, we can assume that the project tries to follow a written organizational policy for implementing software configuration management **Activit(y)**ies.

4) The limitations of the data and heuristics

The documents related to Software Configuration Management should be predefined and recongnized in the knowledge base.

KPA 6.2 Has the project identified, controlled, and made available the software work products through the use of configuration management?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Rework	**Adds, Changes, Deletes**	**Rework** contains attributes of an **Artifact**.	Configuration Management Tool (ClearCase)	CMProduct
Version	**VersionIdentification**	**Version** contains **Subsystem**s.		

2) The heuristics used to make an assessment based on that data

The fact *CMProduct* is used in the JESS rule. *CMProduct* checks the existence of **Artifact**s with the attributes of **Version**, lines added, lines deleted, and lines modified. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the *CMProduct* is more than 1, which means the **Artifact**s with the attributes of **Version**, lines added, lines deleted, and lines modified are exist.

```
(defrule KPA6-2Yes
  (CMProduct, ?x)
  (test (<= 1 ?x) )
```

```
  =>
  (assert (KPA6-2 Yes))
  (printout Result.txt "KPA6.2 Has the project identified, controlled, and made available the software work products through the use of configuration management? Yes." crlf))

(defrule KPA6-2No
  (CMProduct, ?x)
  (test (= 0 ?x) )
  =>
  (assert (KPA6-2 No))
  (printout Result.txt "KPA6.2 Has the project identified, controlled, and made available the software work products through the use of configuration management? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

By checking the existence of **Artifact**s with the attributes of **Version**, lines added, lines deleted, and lines modified, we can assume that the project is identified, controlled, and made available the software work products through the use of configuration management.

4) The limitations of the data and heuristics

None

KPA 6.3 Does the project follow a documented procedure to control changes to configuration items/units?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	Configuration Document

2) The heuristics used to make an assessment based on that data

The fact *ConfigurationDocument* is used in the JESS rule. *ConfigurationDocument* counts the number of document related to configuration management. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the *ConfigurationDocument* is more than 1, which means the document related to configuration management plan is exist.

```
(defrule KPA6-3Yes
  (ConfigurationDocument, ?x)
  (test (= 1 ?x) )
   =>
  (assert (KPA6-3 Yes))
```

```
  (printout Result.txt "KPA6.3 Does the project follow a documented procedure to control changes to configuration items/units? Yes." crlf))

(defrule KPA6-3No
  (ConfigurationDocument, ?x)
  (test (= 0 ?x) )
   =>
  (assert (KPA6-3 No))
  (printout Result.txt "KPA6.3 Does the project follow a documented procedure to control changes to configuration items/units? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document related to the configuration management describes the policy for implementing software configuration management **Activit(y)**ies. By checking the existence of configuration management related document, we can assume that the project tries to follow a written organizational policy for implementing software configuration management **Activit(y)**ies.

4) The limitations of the data and heuristics

The documents related to Software Configuration Management should be predefined and recongnized in the knowledge base.

KPA 6.4 Are standard reports on software baselines (e.g., software configuration control board minutes and change request summary and status reports) distributed to affected groups and individuals?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Rework	**Adds, Changes, Deletes**	**Rework** contains attributes of an **Artifact**.	Configuration Management Tool (ClearCase)	Rework
Version	**VersionIdentification**	**Version** contains **Subsystem**s.		Version
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	SCM Activity
Individual	***Experience***$_{set}$(software development years),	**Individual** owns		SCM Audit

	FirstName$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Activit(y)**ies.		
Organization	***Name***$_{set}$, ***Description***$_{set}$			

2) The heuristics used to make an assessment based on that data

The facts *Rework, Version, SCM Activity,* and *SCM Audit* are used in the JESS rule. *Rework* checks the existence of lines added, lines deleted, lines changed. *Version* checks the existend of **Artifact Version**. *SCM Activity* counts the number of **Activit(y)**ies related to software configuration management. *SCM Audit* counts the number of responsible group or **Individual**s assigned to software configuration management. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *Rework, Version, SCM Activity,* and *SCM Audit* is more than 1, which means configuration management items/units is recoreded, SCM activities are developed, and software baseline audits are conducted according to a documented procedure.

```
(defrule KPA6-4Yes
 (Rework, ?w) (Version, ?x) (SCM Activity, ?y) (SCM Audit, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA6-4 Yes))
```

```
(printout Result.txt "KPA 6.4 Are standard reports on software baselines (e.g., software configuration control board minutes and change request summary and status reports) distributed to affected groups and individuals? Yes." crlf))

(defrule KPA6-4No
  (Rework, ?w) (Version, ?x) (SCM Activity, ?y) (SCM Audit, ?z)
  (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA6-4 No))
 (printout Result.txt "KPA 6.4 Are standard reports on software baselines (e.g., software configuration control board minutes and change request summary and status reports) distributed to affected groups and individuals? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the status of configuration management items/units is recoreded, SCM activities are developed, and software baseline audits are conducted according to a documented procedure. The heusristcs above checks the existence of version, lines deleted, lines added, lindes changed, SCM Activity, and Individuals assigned to audit the software baseline.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Configuration Management should be predefined and recongnized in the knowledge base.

KPA 6.5 Does the project follow a written organizational policy for implementing software configuration management activities?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***set, ***Description***set ,***Type***set(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SCM Group
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***set			SCM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***set**,** ***Description***set**,** **Complete**set(Yes=1, No=blank or 0), **ActualCost**set, **PlannedCost**set, ***Type***set(RM, SPP, SPTO,			SCM Activity

	SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SCM Document

2) The heuristics used to make an assessment based on that data

The facts *SCM Group, SCM Tool, SCM Activity,* and *SCM Document* are used in the JESS rule. *SCM Group* counts the number of responsible group or **Individual**s assigned to software configuration management **Activit(y)**ies. *SCM Activity* counts the number of **Activit(y)**ies related to software configuration management. *SCM Tool* counts the number of tools to use for software configuration management. *SCM Document* counts the number of documents related to software configuration management. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *SCM Group, SCM Tool, SCM Activity,* and *SCM Document* is more than 1, which means software configuration management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA6-5Yes
  (SCM Group, ?w) (SCM Tool, ?x) (SCM Activity, ?y) (SCM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA6-5 Yes))
  (printout Result.txt " KPA 6.5 Does the project follow a written organizational policy for implementing
software configuration management activities? Yes." crlf))

(defrule KPA6-5No
  (SCM Group, ?w) (SCM Tool, ?x) (SCM Activity, ?y) (SCM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA6-5 No))
  (printout Result.txt " KPA 6.5 Does the project follow a written organizational policy for implementing
software configuration management activities? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Configuration Management should be predefined and recongnized in the knowledge base.

KPA 6.6 Are project personnel trained to perform the software configuration management activities for which they are responsible?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$, ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	CM Education

2) The heuristics used to make an assessment based on that data

The fact *CM Education* is used in the JESS rule. The *CM Education* checks whether the Configuration Manager's experience is sufficient. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the *CM Education* is 1, which means Configuration Manager's experience is sufficient.

```
(defrule KPA6-6Yes
(CM Education, ?x)
```

```
 (test (= 1  ?x ))
  =>
 (assert (KPA6-6 Yes))
(printout Result.txt "KPA 6.6 Are project personnel trained to perform the software configuration management activities for which they are responsible? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that members of the software engineering group and other software-related groups are trained to perform their SCM **Activit(y)**ies [28]. Configuration Manager is assigned to manage configurations. By checking the training experience of Configuration Manager, we can understand whether the people in the project who are charged with managing the configuration are trained in the procedures for managing configuration.

4) The limitations of the data and heuristics

None

KPA 6.7 Are measurements used to determine the status of activities for software configuration management (e.g., effort and funds expended for software configuration management activities)?

1) The data available related to the CMM analysis

PAMPA 2.0	Source	Facts elicited

Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Type*$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), *Description*$_{set}$, Complete$_{set}$**(Yes=1, No=blank or 0), **ActualCost$_{set}$, PlannedCost$_{set}$**		Project Planning Tool (MS Project)	CM Activity, Bad CM Activity
InitialMilestone	***PlannedStartDate*$_{set}$, *ActualStartDate*$_{set}$**	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate*$_{set}$, *ActualEndDate*$_{set}$**	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *CM Activity,* and *Bad CM Activity* are used in the JESS rule. *CM Activity* counts the number of **Activit(y)**ies related to Software Configuration Management. *Bad CM Activity* counts the number of software configuration management **Activit(y)**ies, which do not have proper attributes for measuring effort and funds expended. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result $\frac{(CMActivity - BadCMActivity)}{CMActivity}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA6-7AlmostAlways
  (CMActivity, ?x)
  (BadCMActivity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA6-7 AlmostAlways))
  (printout Result.txt "KPA 6.7 Are measurements used to determine the status of activities for software configuration management (e.g., effort and funds expended for software configuration management activities)? Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activit(y)**ies related to Software Configuration Management and the attributes of effort and funds expended for Software Configuration Management **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Configuration Management should be predefined and recongnized in the knowledge base.

KPA 6.8 Are periodic audits performed to verify that software baselines conform to the documentation that defines them (e.g., by the SCM group)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	CM Reviewer
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			CM Review

2) The heuristics used to make an assessment based on that data

The fact *CM Review, CM Reviewer* are used in the JESS rule. *CM Review* counts the number of **Activit(y)**ies related to Configuration Management review. *CM Reviewer* counts the number of **Individual**s who are assigned to review CM **Activit(y)**ies. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to CM review and **Individual**s to review CM **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA6-8Yes
  (CM Review, ?x) (CM Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA6-8 Yes))
  (printout Result.txt "KPA 6.8 Are periodic audits performed to verify that software baselines conform to
the documentation that defines them (e.g., by the SCM group)? Yes." crlf))

(defrule KPA6-8No
  (CM Review, ?x) (CM Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA6-8 No))
  (printout Result.txt "KPA 6.8 Are periodic audits performed to verify that software baselines conform to
the documentation that defines them (e.g., by the SCM group)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of CM review **Activit(y)**ies and the assigned **Individual**s (SCM group) to review CM **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Software Configuration Management should be predefined and recongnized in the knowledge base.

I. KPA 7. Organization Process Focus

The purpose of Organization Process Focus is to establish the organizational responsibility of software process activities that improve the organization's overall software process capability [28].

KPA 7.1 Are the activities for developing and improving the organization's and project's software processes coordinated across the organization (e.g., via a software engineering process group)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	OPF Activity, Process Database
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			OPF Audit, OPF Training
Organization	***Name***$_{set}$**,** ***Description***$_{set}$ ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***Training***$_{set}$(audience, objectives, length, lesson plans)			

2) The heuristics used to make an assessment based on that data

The facts *OPF Activity, Process Database, OPF Audit,* and *OPF Training* are used in the JESS rule. *OPF Activity* counts the number of **Activit(y)**ies related to developing and improving the organization's and project's software processes coordinated across the organization. *Process Database* checks the existence of proecess datat. *OPF Audit* counts the number of responsible group or **Individual**s assigned to developing and improving the organization's and project's software processes coordinated across the organization. *OPF Training* checks the status of training related to developing and improving the organization's and project's software processes coordinated across the organization. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *OPF Activity, Process Database, OPF Audit,* and *OPF Training* is more than 1, which means **Activit(y)**ies for developing and improving software process coordination, **Process** database, **Individual**s to monitor, evaluate new processes, methods, and tools, and Individual training experience are exist.

```
(defrule KPA7-1Yes
 (OPF Activity, ?w) (Process Database, ?x) (OPF Audit, ?y) (OPF Training, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA7-1 Yes))
```

(printout Result.txt " KPA 7.1 Are the activities for developing and improving the organization's and project's software processes coordinated across the organization (e.g., via a software engineering process group)? Yes." crlf))

```
(defrule KPA7-1No
 (OPF Activity, ?w) (Process Database, ?x) (OPF Audit, ?y) (OPF Training, ?z)
  (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA7-1 No))
 (printout Result.txt " KPA 7.1 Are the activities for developing and improving the organization's and project's software processes coordinated across the organization (e.g., via a software engineering process group)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be Activit(y)ies for developing and improving software process coordination, process database, Individuals to monitor, evaluate new processes, methods, and tools, and Individual training experience [28].

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.2 Is your organization's software process assessed periodically?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Description*$_{set}$, Complete$_{set}$**(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	OPF Activity
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		OPF Audit, Tools
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)	**Organization** contains **Individual**s		

2) The heuristics used to make an assessment based on that data

The facts *OPF Activity, OPF Audit,* and *Tools* are used in the JESS rule. *OPF Activity* counts the number of **Activit(y)**ies related to software process assessment. *OPF Audit* counts the number of responsible group or **Individual**s assigned to software process assessment. *Tools* checks the number of tools related to software process assessment. The fact-related attributes are gathered automatically by querying attributes from PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *OPF Activity, OPF Audit,* and *Tools* is more than 1, which means **Activit(y)**ies for software process assessment, **Individual**s to implement s software process assessment, and the necessary tools are exist.

```
(defrule KPA7-2Yes
 (OPF Activity, ?x) (OPF Audit, ?y) (Tools, ?z)
  (test (and (< 0 ?x ) (< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA7-1 Yes))
 (printout Result.txt "KPA 7.2 Is your organization's software process assessed periodically? Yes." crlf))

(defrule KPA7-1No
 (OPF Activity, ?x) (OPF Audit, ?y) (Tools, ?z)
  (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA7-1 No))
```

(printout Result.txt "KPA 7.2 Is your organization's software process assessed periodically? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be **Activit(y)**ies related to software process assessment, group or **Individual**s who are responsible for the **Activit(y)**ies, and necessary tools [28]. The heuristics above checks the existence of software process assessment related **Activit(y)**ies, groups or **Individual**s, and tools.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.3 Does your organization follow a documented plan for developing and improving its software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	OPF Group

Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			OPF Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			OPF Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	OPF Document

2) The heuristics used to make an assessment based on that data

The facts *OPF Group, OPF Tool, OPF Activity,* and *OPF Document* are used in the JESS rule. *OPF Group* counts the number of responsible group or **Individual**s assigned to developing and improving software process. *OPF Activity* counts the number of **Activit(y)**ies related to developing and improving software process. *OPF Tool* counts the number of tools to use for developing and improving software process. *OPF Document* counts the number of documents related to developing and improving software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *OPF Group, OPF Tool, OPF Activity,* and *OPF Document* is more than 1, which means developing and improving software process related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA7-3Yes
  (OPF Group, ?w) (OPF Tool, ?x) (OPF Activity, ?y) (OPF Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA7-3 Yes))
  (printout Result.txt " KPA 7.3 Does your organization follow a documented plan for developing and
improving its software process? Yes." crlf))

(defrule KPA7-3No
  (OPF Group, ?w) (OPF Tool, ?x) (OPF Activity, ?y) (OPF Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
```

(assert (KPA7-3 No))

(printout Result.txt " KPA 7.3 Does your organization follow a documented plan for developing and improving its software process? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.4 Does senior management sponsor the organization's activities for software process development and improvements (e.g., by establishing long-term plans, and by committing resources and funding)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Organization	***Name***set***, Description***set ,***Type***set(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	OPF Group
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***set			OPF Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***set***,*** ***Description***set***,*** **Complete**set(Yes=1, No=blank or 0), **ActualCost**set, **PlannedCost**set, ***Type***set(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			OPF Activity
Artifact	***Name***set **Type**(RequirementsFile, DesignFile, DocumentFile,	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	OPF Document

	SourceFile, Other)			

2) The heuristics used to make an assessment based on that data

The facts *OPF Group, OPF Tool, OPF Activity,* and *OPF Document* are used in the JESS rule. *OPF Group* counts the number of responsible group or **Individual**s assigned to developing and improving software process. *OPF Activity* counts the number of **Activit(y)**ies related to developing and improving software process. *OPF Tool* counts the number of tools to use for developing and improving software process. *OPF Document* counts the number of documents related to developing and improving software process. The fact-related attrinbuters are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *OPF Group, OPF Tool, OPF Activity,* and *OPF Document* is more than 1, which means developing and improving software process related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA7-4Yes
 (OPF Group, ?w) (OPF Tool, ?x) (OPF Activity, ?y) (OPF Document, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA7-4 Yes))
```

```
  (printout Result.txt " KPA 7.4 Does senior management sponsor the organization's activities for software
process development and improvements (e.g., by establishing long-term plans, and by committing
resources and funding)? Yes." crlf))

(defrule KPA7-4No
  (OPF Group, ?w) (OPF Tool, ?x) (OPF Activity, ?y) (OPF Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA7-4 No))
  (printout Result.txt " KPA 7.4 Does senior management sponsor the organization's activities for software
process development and improvements (e.g., by establishing long-term plans, and by committing
resources and funding)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.5 Do one or more individuals have full-time or part-time responsibility for the organization's software process activities (e.g., a software engineering process group)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies	Project Planning Tool (MS Project)	Process Development Responsibility, Not Process Development Responsibility
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**, Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The facts *Process Development Responsibility,* and *Not Process Development Responsibility* are used in the JESS rule. *Process Development Responsibility* counts the number of resource assignment to **Activit(y)**ies related to process development and improvement. *Not Process Development Responsibility* counts the number of above **Activit(y)**ies with no **Individual**s assigned. The fact-related attributers are gathered

automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below.

If $\frac{(ProcessDevelopmentResponsibility - NotProcessDevelopmentResponsibility)}{ProcessDevelopmentResponsibility}$ is more than 0.9, it means 90 % of the project is assigned.

```
(defrule KPA7-5AlmostAlways
  (ProcessDevelopmentResponsibility, ?x)
  (NotProcessDevelopmentResponsibility, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA7-5 AlmostAlways))
  (printout Result.txt " KPA 7.5 Do one or more individuals have full-time or part-time responsibility for the organization’s software process activities (e.g., a software engineering process group)? AlmostAlways." crlf))

(defrule KPA7-5Frequently
  (ProcessDevelopmentResponsibility, ?x)
  (NotProcessDevelopmentResponsibility, ?y)
  (test (<= 0.6 (/ (- ?x ?y) ?x)))
  (test (>  0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA7-5 Frequently))
```

(printout Result.txt " KPA 7.5 Do one or more individuals have full-time or part-time responsibility for the organization's software process activities (e.g., a software engineering process group)? Frequently." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The heuristics above checks the **Activit(y)**ies related to process development and improvement and the assignment of **Individual**s to these **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.6 Are measurements used to determine the status of the activities performed to develop and improve the organization's software process (e.g., effort expended for software process assessment and improvement)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF,		Project Planning Tool (MS Project)	OPF Activity, Bad OPF Activity

	OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *OPF Activity*, and *Bad OPF Activity* are used in the JESS rule. *OPF Activity* counts the number of **Activit(y)**ies performed to develop and improve the organization's software process. *Bad OPF Activity* counts the number of **Activit(y)**ies, which do not have proper attributes for measuring effort expended for performing to develop and improve the organization's software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(OPF\text{Activity} - BadOPF\text{Activity})}{\text{OPF Activity}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA7-6AlmostAlways
  (Process development and improvement Activity, ?x)
  (Bad Process development and improvement Activity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA7-6 AlmostAlways))
  (printout Result.txt " KPA 7.6 Are measurements used to determine the status of the activities performed to develop and improve the organization's software process (e.g., effort expended for software process assessment and improvement)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activit(y)**ies performed to develop and improve the organization's software process and the attributes of effort expended for software process assessment and improvement.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Focus should be predefined and recongnized in the knowledge base.

KPA 7.7 Are the activities performed for developing and improving software processes reviewed periodically with senior management?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	PDI Reviewer
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			PDI Review

2) The heuristics used to make an assessment based on that data

The fact *PDI Review, PDI Reviewer* are used in the JESS rule. *PDI Review* counts the number of **Activit(y)**ies related to developing and improving software processes review. *PDI Reviewer* counts the number of **Individual**s who are assigned to review developing and improving software processes **Activit(y)**ies. The fact-related attributes are gathered

automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to developing and improving software processes review and **Individual**s to review developing and improving software processes **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA7-7Yes
  (PDI Review, ?x)
  (PDI Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA7-7 Yes))
  (printout Result.txt "KPA 7.7 Are the activities performed for developing and improving software processes reviewed periodically with senior management? Yes." crlf))

(defrule KPA7-7No
  (PDI Review, ?x)
  (PDI Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA7-7 No))
  (printout Result.txt "KPA 7.7 Are the activities performed for developing and improving software processes reviewed periodically with senior management? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of developing and improving software processes review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review developing and improving software processes **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Focus should be predefined and recongnized in the knowledge base.

J. KPA 8. Organization Process Definition

The purpose of Organization Process Definition is to develop and maintain a usable set of software process assets that improve process performance across the projects and provide a basis for cumulative, long-term benefits to the organization [28].

KPA 8.1 Has your organization developed, and does it maintain, a standard software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile,	**Artifact**s are related to	Configuration Management	OPD Document

	SourceFile, Other)	**Feature**s.	Tool (ClearCase)	
SLCModel	***Name***$_{set}$, ***Description***$_{set}$		Project Planning Tool (MS Project	Software Life Cycle

2) The heuristics used to make an assessment based on that data

The facts *OPD Document* and *Software Life Cycle* are used in the JESS rule. *OPD Document* counts the number of documents related to developing and maintaining standard software process and related process assets. *Software Life Cycle* counts the number of Software Life Cycle in the project used. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *OPD Document*, and *Software Life Cycle* is more than 1, which means the organization developed, and maintain, a standard software process.

```
(defrule KPA8-1Yes
 (OPD Document, ?x) (Software Life Cycle, ?y)
  (test (and (< 0 ?x ) (< 0 ?y )))
  =>
 (assert (KPA8-1 Yes))
```

```
  (printout Result.txt " KPA 8.1 Has your organization developed, and does it maintain, a standard software
process? Yes." crlf))

(defrule KPA8-1No
  (OPD Document, ?x) (Software Life Cycle, ?y)
   (test (or (= 0 ?x )(= 0 ?y )))
   =>
  (assert (KPA8-1 No))
  (printout Result.txt " KPA 8.1 Has your organization developed, and does it maintain, a standard software
process? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the organizations standard software process is developed and maintained according to a documented procedure. Second, the organization's standard software process is documented according to established organization standards. Third, descriptions of software life cycles that are documented and maintained. Forth, guidelines and criteria for the projects' tailoring of the organization's standard software process are developed and maintained [28]. The heuristics checks the existence of the document about organization's standard software process, and Software Life Cycle.

4) The limitations of the data and heuristics

The documents related to Organization Process Definiton should be predefined and recongnized in the knowledge base.

KPA 8.2 Does the organization collect, review, and make available information related to the use of the organization's standard software process (e.g., estimates and actual data on software size, effort, and cost; productivity data; and quality measurements)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	OPD Activity
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$, ***PlannedCost***$_{set}$, ***ActualCost***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
Final-	***PlannedEndDate***$_{set}$,	**FinalMilestone**		

Milstone	***ActualEndDate***$_{set}$	is an attribute of **Activity**.		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Tool (ClearCase)	OPD Document

2) The heuristics used to make an assessment based on that data

The facts *OPD Activity,* and *OPD Document* are used in the JESS rule. *OPD Activity* checks the estimates and actual data on software size, effort, and cost related to organization's standard software process. *OPD Document* counts the number of documents related to developing and maintaining standard software process and related process assets. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *OPD Activity,* and *OPD Document* is more than 1, which means the organization collects, reviews, and makes available information related to the use of the organization's standard software process.

```
(defrule KPA8-2Yes
 (OPD Activity, ?x) (OPD Document, ?y)
  (test (and (< 0 ?x )(< 0 ?y )))
  =>
```

```
 (assert (KPA8-2 Yes))
 (printout Result.txt " KPA 8.2 Does the organization collect, review, and make available information related to the use of the organization's standard software process (e.g., estimates and actual data on software size, effort, and cost; productivity data; and quality measurements)?  Yes." crlf))

(defrule KPA8-2No
 (OPD Activity, ?x) (OPD Document, ?y)
  (test (or (= 0 ?x )(= 0 ?y )))
  =>
 (assert (KPA8-2 No))
 (printout Result.txt "KPA 8.2 Does the organization collect, review, and make available information related to the use of the organization's standard software process (e.g., estimates and actual data on software size, effort, and cost; productivity data; and quality measurements)?  No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the organization's software process database is established and maintained. Second, a library of software process-related documentation is established and maintained [28]. The heuristics checks the estimates and actual data on software size, effort, and cost related to organization's standard software process. And it checks the existence of software process-related document.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Organization Process Definiton should be predefined and recongnized in the knowledge base.

KPA 8.3 Does the organization follow a written policy for both developing and maintaining its standard software process and related process assets (e.g., descriptions of approved software life cycles)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$**,** ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	OPD Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			OPD Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0),			OPD Activity

	ActualCost$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM</u>)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	OPD Document

2) The heuristics used to make an assessment based on that data

The facts *OPD Group, OPD Tool, OPD Activity,* and *OPD Document* are used in the JESS rule. *OPD Group* counts the number of responsible group or **Individual**s assigned to developing and maintaining standard software process and related process assets. *OPD Activity* counts the number of **Activit(y)**ies related to developing and maintaining standard software process and related process assets. *OPD Tool* counts the number of tools to use for developing and maintaining standard software process and related process assets. *OPD Document* counts the number of documents related to developing and maintaining standard software process and related process assets. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0

Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *OPD Group, OPD Tool, OPD Activity,* and *OPD Document* is more than 1, which means developing and maintaining standard software process and related process assets related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA8-3Yes
  (OPD Group, ?w) (OPD Tool, ?x) (OPD Activity, ?y) (OPD Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA8-3 Yes))
  (printout Result.txt " KPA 8.3 Does the organization follow a written policy for both developing and
maintaining its standard software process and related process assets (e.g., descriptions of approved
software life cycles)? Yes." crlf))
```

```
(defrule KPA8-3No
  (OPD Group, ?w) (OPD Tool, ?x) (OPD Activity, ?y) (OPD Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA8-3 No))
  (printout Result.txt " KPA 8.3 Does the organization follow a written policy for both developing and
maintaining its standard software process and related process assets (e.g., descriptions of approved
software life cycles)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Organization Process Definiton should be predefined and recongnized in the knowledge base.

KPA 8.4 Do individuals who develop and maintain the organization's standard software process receive the required training to perform these activities?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Training***$_{set}$(audience, objectives, length, lesson plans)	**Organization** contains **Individual**s	Project Planning Tool (MS Project)	Process Education
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$			

	FirstName$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			

2) The heuristics used to make an assessment based on that data

The fact *Process Education* is used in the JESS rule. The *Process Education* checks whether the **Individual**s, who develop and maintain the organization's standard software process, have sufficient experiences. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Process Education* is 1, which means the **Individual**s, who develop and maintain the organization's standard software process, have sufficient experiences.

```
(defrule KPA8-4Yes
(Process Education, ?x)
(test (= 1  ?x ))
=>
(assert (KPA8-4 Yes))
(printout Result.txt " KPA 8.4 Do individuals who develop and maintain the organization's standard software process receive the required training to perform these activities? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the **Individual**s who develop and maintain the **Organization**'s standard software **Process** and related **Process** assets receive required training to perform these **Activit(y)**ies [28]. By checking the training experience of **Individual**s who develop and maintain the **Organization**'s standard software **Process**, we can understand whether the **Individual**s who develop and maintain the organization's standard software process receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 8.5 Are measurements used to determine the status of the activities performed to define and maintain the organization's standard software process (e.g., status of schedule milestones and the cost of process definition activities)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Type*$_{set}$**(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE,		Project Management Tool (MS Project)	Process definition Activity, Bad Process definition Activity

	IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Process definition Activity*, and *Bad Process definition Activity* are used in the JESS rule. *Process definition Activity* counts the number of **Activit(y)**ies performed to define and maintain the organization's standard software process. *Bad Process definition Activity* counts the number of process definition **Activit(y)**ies, which do not have proper attributes for status of schedule milestones and the cost. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{Process definition Activity} - Bad\,\Pr ocessdefinitionA\text{tivity})}{\text{Process definiton Activity}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA8-5AlmostAlways
  (Process definition Activity, ?x)
  (Bad Process definition Activity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA8-5 AlmostAlways))
  (printout Result.txt "KPA 8.5 Are measurements used to determine the status of the activities performed to define and maintain the organization's standard software process (e.g., status of schedule milestones and the cost of process definition activities)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **Activit(y)**ies performed to define and maintain the organization's standard software process and the attributes of status of schedule milestones and the cost of process definition activities

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Organization Process Definiton should be predefined and recongnized in the knowledge base.

KPA 8.6 Are the activities and work products for developing and maintaining the organization's standard software process subjected to SQA review and audit?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	Process Definition Reviewer
Organization	***Name***set, ***Description***set	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***set, ***Description***set, **Complete**set(Yes=1, No=blank or 0), ***Type***set(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			Process Definition Review

2) The heuristics used to make an assessment based on that data

The fact *Process Definition Review, Process Definition Reviewer* are used in the JESS rule. *Process Definition Review* counts the number of **Activit(y)**ies related to developing and maintaining the organization's standard software process review. *Process Definiton Reviewer* counts the number of **Individual**s who are assigned to review developing and maintaining the organization's standard software process **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to developing and maintaining the organization's standard software process review and **Individual**s to review developing and maintaining the organization's standard software process **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA8-6Yes
  (Process Definition Review, ?x)
  (Process Definition Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA8-6 Yes))
  (printout Result.txt " KPA 8.6 Are the activities and work products for developing and maintaining the organization's standard software process subjected to SQA review and audit? Yes." crlf))

(defrule KPA8-6No
```

```
(Process Definition Review, ?x)
(Process Definition Reviewer, ?y)
(test (or(= 0 ?x) (= 0 ?y)  ))
 =>
(assert (KPA8-6 No))
(printout Result.txt "KPA 8.6 Are the activities and work products for developing and maintaining the organization's standard software process subjected to SQA review and audit? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of developing and maintaining the organization's standard software process review **Activit(y)**ies and the assigned **Individual**s (SQA group) to review developing and maintaining the organization's standard software process **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Organization Process Definiton should be predefined and recongnized in the knowledge base.

K. KPA 9. Training Program

The purpose of the Training Program key process area is to develop the skills and knowledge of individuals so they can perform their roles effectively and efficiently [28].

KPA 9.1 Are training activities planned?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	TrainingDocument

2) The heuristics used to make an assessment based on that data

The fact *TrainingDocument* is used in the JESS rule. *TrainingDocument* counts the number of document such as "Training Plan". The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *TrainingDocument* is more than 1, which means the documents "Training Plan" is exist.

```
(defrule KPA9-1Yes
  (TrainingDocument, ?x)   (test (<= 1  ?x ))
   =>
  (assert (KPA9-1 Yes))
  (printout Result.txt " KPA 9.1 Are training activities planned? Yes." crlf))

(defrule KPA9-1No
```

```
(TrainingDocument, ?x)
(test (= 0  ?x ))
 =>
(assert (KPA9-1 No))
(printout Result.txt " KPA 9.1 Are training activities planned? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document "Training Plan" states the policy to meet organization's training needs. By checking the existence of "Training Plan" document, we can assume that the project tries to follow a written organizational policy to meet its training needs.

4) The limitations of the data and heuristics

The documents related to Training Program should be predefined and recongnized in the knowledge base.

KPA 9.2 Is training provided for developing the skills and knowledge needed to perform software managerial and technical roles?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set,}$ ***SoftwareTools***$_{set}$(Word Processing		Project Planning	Training Support

	system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***TrainingFacilities***$_{set,}$ ***CourseReview***$_{set}$(**Individual**, software managers), ***Training***$_{set}$(audience, objectives, length, lesson plans)		Tool (MS Project)	

2) The heuristics used to make an assessment based on that data

The fact *Training Support* is used in the JESS rule. *Training Support* checks the status of training attributes such as intended audience, training objectives, length of the training, lesson plans, evaluation, and training material reviewed. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks *Traing Support* is more than 1, which means intended audience, training objectives, length of the training, lesson plans, evaluation, and training material review are exist.

```
(defrule KPA9-2Yes
 (Training Support, ?x)
  (test (and (< 0 ?x )))
  =>
 (assert (KPA9-2 Yes))
```

```
 (printout Result.txt " KPA 9.2 Is training provided for developing the skills and knowledge needed to perform software managerial and technical roles? Yes." crlf))

(defrule KPA9-2No
 (Training Support,  ?x)
  (test (or (= 0 ?x )))
  =>
 (assert (KPA9-2 No))
 (printout Result.txt " KPA 9.2 Is training provided for developing the skills and knowledge needed to perform software managerial and technical roles? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the training courses prepared at the organization level are developed and maintained according to organization standards. The training course should address intended audience, training objectives, length of the training, lesson plans, criteria for determining the students' satisfactory completion, procedures for periodically evaluating the effectiveness of the training. The materials for the training course are reviewed by instructional experts, subject matter experts, and representative students from pilot sessions of the training course being reviewed [28]. The heurisitics checks the existence of training attributes such as intended audience, training objectives, length of the training, lesson plans, evaluation, and training material reviewed.

4) The limitations of the data and heuristics

None

KPA 9.3 Do members of the software engineering group and other software-related groups receive the training necessary to perform their roles?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Training***$_{set}$(audience, objectives, length, lesson plans)	**Organization** contains **Individual**s	Project Planning Tool (MS Project)	Education
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			

2) The heuristics used to make an assessment based on that data

The fact *Education* is used in the JESS rule. The *Education* checks whether the **Individual**s, who are members of the software engineering group and other software-related groups, have sufficient experiences. The fact-related attributes are gathered

automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Education* is 1, which means the **Individual**s, who are members of the software engineering group and other software-related groups, have sufficient experiences.

```
(defrule KPA9-3Yes
(Education, ?x)
(test (= 1  ?x ))
=>
(assert (KPA9-3 Yes))
(printout Result.txt " KPA 9.3 Do members of the software engineering group and other software-related groups receive the training necessary to perform their roles? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The heuristics checks the training experience of **Individual**s, who are members of the software engineering group and other software-related groups.

4) The limitations of the data and heuristics

None

KPA 9.4 Does your organization follow a written organizational policy to meet its training needs?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$**,** ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	TP Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			TP Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			TP Activity

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	TP Document

2) The heuristics used to make an assessment based on that data

The facts *TP Group, TP Tool, TP Activity,* and *TP Document* are used in the JESS rule. *TP Group* counts the number of responsible group or **Individual**s assigned to training program. *TP Activity* counts the number of **Activit(y)**ies related to training program. *TP Tool* counts the number of tools to use for training program. *TP Document* counts the number of documents related to training program. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *TP Group, TP Tool, TP Activity,* and *TP Document* is more than 1, which means training program related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA9-4Yes
 (TP Group, ?w) (TP Tool, ?x) (TP Activity, ?y) (TP Document, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA9-4 Yes))
```

```
 (printout Result.txt " KPA 9.4 Does your organization follow a written organizational policy to meet its training needs?
 Yes." crlf))

(defrule KPA9-4No
  (TP Group, ?w) (TP Tool, ?x) (TP Activity, ?y) (TP Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA9-4 No))
  (printout Result.txt " KPA 9.4 Does your organization follow a written organizational policy to meet its training needs?
 No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Training Program should be predefined and recongnized in the knowledge base.

KPA 9.5 Are adequate resources provided to implement the organization's training program (e.g., funding, software tools, appropriate training facilities)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActualCost***$_{set}$, ***PlannedCost***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), **Complete**$_{set}$(Yes=1, No=blank or 0)		Project Planning Tool (MS Project)	Funds satisfied Training Program Activity, Number of completed Training Program task
Organization	***Name***$_{set}$, ***Description***$_{set}$ ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$, ***TrainingFacilities***$_{set}$			Sufficient tools and facilities

2) The heuristics used to make an assessment based on that data

The facts *Funds satisfied Training Program Activity, Number of completed Training Program task,* and *Sufficient tools and facilities* are used in the JESS rule. *Funds satisfied Training Program Activity* counts the number of **ActualCost**$_{set}$, which is less than the **PlannedCost**$_{set}$ in Training Program related **Activit(y)**ies. *Number of completed Training Program task* counts the number of completed Training Program related

Activit(y)ies. *Sufficient tools and facilities* counts the number of software tools, appropriate training facilities. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{\text{Funds satisfied Training Program Activity}}{\text{Number of completed Training Program task}}$ is more than 0.9, it means 90 % of the Training Program related **Activit(y)**ies are provided with sufficient funding. It also checks whether *Sufficient tools and facilities* is more than 1, which means the organization has software tools, and appropriate training facilities.

```
(defrule KPA9-5AlmostAlways
  (Funds satisfied Training Program Activity, ?x)
  (Number of completed Training Program task, ?y)
  (Sufficient tools and facilities, ?z)
(test (and (= 1 ?z) (<= 0.9  (/ ?x  ?y) ))
   =>
  (assert (KPA9-5 AlmostAlways))
  (printout Result.txt " KPA 9.5 Are adequate resources provided to implement the organization’s training
program (e.g., funding, software tools, appropriate training facilities)? AlmostAlways" crlf))
----------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **ActualCost**$_{set}$ to **PlannedCost**$_{set}$ in Training Program related **Activit(y)**ies and the existence of organization's software tools, and appropriate training facilities.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Training Program should be predefined and recongnized in the knowledge base.

KPA 9.6 Are measurements used to determine the quality of the training program?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	Training Measure
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***CourseReview***$_{set}$(**Individual**, software managers)			

2) The heuristics used to make an assessment based on that data

The fact *Training Measure* is used in the JESS rule. *Training Measure* counts the number of attributes related to determine the quality of the training program such as results of post-training tests, reviews of the courses from the students, and feedback from the software managers. The fact-related attrinbutes are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks *Training Measure* is more than 1, which means attributes of results of post-training tests, reviews of the courses from the students, and feedback from the software managers are exist.

```
(defrule KPA9-6Yes
  (Training Measure, ?x)
  (test (<= 1 ?x ))

   =>
  (assert (KPA9-6 Yes))
  (printout Result.txt " KPA 9.6 Are measurements used to determine the quality of the training program?
Yes." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that, examples of measurement include results of post-training tests, reviews of the courses from the students, and feedback from the software managers [28]. The heuristics checks the existence of attributes such as results of post-training tests, reviews of the courses from the students, and feedback from the software managers.

4) The limitations of the data and heuristics

None

KPA 9.7 Are training program activities reviewed with senior management on a periodic basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	Training Program Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			Training Program Review

2) The heuristics used to make an assessment based on that data

The fact *Training Program Review, Training Program Reviewer* are used in the JESS rule. *Training Program Review* counts the number of **Activit(y)**ies related to training program review. *Training Program Reviewer* counts the number of **Individual**s who are assigned to review training program **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to training program review and **Individual**s to review training program **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA9-7Yes
  (Training Program Review, ?x)
  (Training Program Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA9-7 Yes))
  (printout Result.txt " KPA 9.7 Are training program activities reviewed with senior management on a periodic basis? Yes." crlf))

(defrule KPA9-7No
  (Training Program Review, ?x)
  (Training Program Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
```

(assert (KPA9-7 No))

(printout Result.txt "KPA 9.7 Are training program activities reviewed with senior management on a periodic basis? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of training program review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review training program **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Training Program should be predefined and recongnized in the knowledge base.

K. KPA 10. Integrated Software Management

The purpose of Integrated Software Management is to integrate the software engineering and management activities into a coherent, defined software process that is tailored from the organization's standard software process and related process assets, which are described in Organization Process Definition [28].

KPA 10.1 Was the project's defined software process developed by tailoring the organization's standard software process?

1) The data available related to the CMM analysis

None

2) The heuristics used to make an assessment based on that data

None

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the project's defined software process is developed by tailoring the organization's standard software process according to a documented procedure. Second, each project's defined software process is revised according to a documented procedure. Third, the project's software development plan, which describes the use of the project's defined software process, is developed and revised according to a documented procedure [28].

4) The limitations of the data and heuristics

It is not possible to make objective data from subjective data such as how the software process is developed, and revised.

KPA 10.2 Is the project planned and managed in accordance with the project's defined software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Description*$_{set}$, Complete$_{set}$**(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	Software process database, Effort & cost, ISM Review
Organization	***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***ComputerResource***$_{set}$	**Organization** contains **Individual**s		Critical Resource
Risk	***EstimatedRisk***$_{set,}$ ***Description***$_{set}$			Risk
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase)	ISM Document, Software Product Size

2) The heuristics used to make an assessment based on that data

The facts *Software process database, Effort & cost, ISM Review, Critical Resource, Risk, ISM Document,* and *Software Product Size* are used in the JESS rule. *Software process database* checks the existence of software process attributes. *Effort & cost* checks the existence of effort and cost attributes from Integrated Software Management. *ISM Review* counts the number of **Activit(y)**ies related to Integrated Software Managementy. *Risk* counts the number of **Risk** attributes. *ISM Document* counts the number of documents related to software project planning and managing using the organization's standard software process. *Software Product Size* checks the existence of size attribute in the object **Artifact**. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *Software process database, Effort & cost, ISM Review, Critical Resource, Risk, ISM Document,* and *Software Product Size* is more than 1, which means it is possible that there is software process database, size of the software work product management, effort & cost management, project's critical coumputer resource management, risk management, review of the software project.

```
(defrule KPA10-2Yes
 (Software process database, ?t) (Effort & cost, ?u) (ISM Review, ?v) (Critical Resource, ?w) (Risk, ?x)
(ISM Document, ?y) (Software Product Size, ?z)
  (test (and (< 0 ?t ) (< 0 ?u ) (< 0 ?v ) (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
```

```
  (assert (KPA10-2 Yes))
  (printout Result.txt " KPA 10.2 Is the project planned and managed in accordance with the project's
defined software process? Yes." crlf))

(defrule KPA10-2No
  (Software process database, ?t) (Effort & cost, ?u) (ISM Review, ?v) (Critical Resource, ?w) (Risk, ?x)
(ISM Document, ?y) (Software Product Size, ?z)
   (test (or (= 0 ?t ) (= 0 ?u ) (= 0 ?v ) (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA10-2 No))
  (printout Result.txt " KPA 10.2 Is the project planned and managed in accordance with the project's
defined software process? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capabililty Maturity Model states that there should be explanation of how project's software development plan is developed and revised and how project is managed. And there should be software process database, size of the software work product management, effort & cost management, project's critical coumputer resource management, critical dependencies and critical path management, risk management, review of the software project [28]. The above heuristics checks the existence of software process database, size, effort & cost, project's critical coumputer resource, risk, software project review **Activit(y)**ies, and the document related to software project planning and managing.

4) The limitations of the data and heuristics

It is not possible to make objective data from subjective data such as how project's software development plan is developed and revised, how project is managed, and how critical dependencies and critical path is management.

KPA 10.3 Does the project follow a written organizational policy requiring that the software project be planned and managed using the organization's standard software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	ISM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			ISM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$,			ISM Activity

	Description$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	ISM Document

2) The heuristics used to make an assessment based on that data

The facts *ISM Group, ISM Tool, ISM Activity,* and *ISM Document* are used in the JESS rule. *ISM Group* counts the number of responsible group or **Individual**s assigned to software project planning and managing using the organization's standard software process. *ISM Activity* counts the number of **Activit(y)**ies related to software project planning and managing using the organization's standard software process. *ISM Tool* counts the number of tools to use for software project planning and managing using the organization's standard software process. *ISM Document* counts the number of documents related to software project planning and managing using the organization's standard software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *ISM Group, ISM Tool, ISM Activity,* and *ISM Document* is more than 1, which means software project planning related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA10-3Yes
  (ISM Group, ?w) (ISM Tool, ?x) (ISM Activity, ?y) (ISM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA10-3 Yes))
  (printout Result.txt " KPA 10.3 Does the project follow a written organizational policy requiring that the
software project be planned and managed using the organization's standard software process? Yes." crlf))

(defrule KPA10-3No
  (ISM Group, ?w) (ISM Tool, ?x) (ISM Activity, ?y) (ISM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA10-3 No))
  (printout Result.txt " KPA 10.3 Does the project follow a written organizational policy requiring that the
software project be planned and managed using the organization's standard software process? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Ingegrated Software Management should be predefined and recongnized in the knowledge base.

KPA 10.4 Is training required for individuals tasked to tailor the organization's standard software process to define a software process for a new project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years) ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	Integrated Software Management Education

2) The heuristics used to make an assessment based on that data

The fact *Integrated Software Management Education* is used in the JESS rule. The *Integrated Software Management Education* checks whether the **Individual**s, who tailor the organization's standard software process to define a software process for a new project, have sufficient experiences. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Integrated Software Management Education* is 1, which means the **Individual**s, who tailor the organization's standard software process to define a software process for a new project, have sufficient experiences.

```
(defrule KPA10-4Yes
(Integrated Software Management Education, ?x)
(test (= 1  ?x ))
 =>
(assert (KPA10-4 Yes))
(printout Result.txt " KPA 10.4 Is training required for individuals tasked to tailor the organization's standard software process to define a software process for a new project? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the **Individual**s responsible for developing the project's defined software process receive required training in how to tailor the organization's standard software process and use the related process assets [28]. By checking the training experience of **Individual**s, who tailor the organization's standard software process to define a software process for a new project, we can understand whether the **Individual**s, who tailor the organization's standard software process to define a software process for a new project, receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 10.5 Are measurements used to determine the effectiveness of the integrated software management activities (e.g., frequency, causes and magnitude of replanning efforts)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM),		Project Planning Tool (MS Project)	Integrated Software Management , Bad Integrated Software Management

	Quantitative Process Management, Software Quality Management, Technology Change Management), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Integrated Software Management* and *Integrated Software Management* are used in the JESS rule. *Integrated Software Management* counts the number of **Activit(y)**ies related to integrated software management. *Bad Integrated Software Management* counts the number of integrated software management **Activit(y)**ies, which do not have proper attributes of schedule and cost. These fact-related attributes are gathered automatically by querying attributes from PAMPA 2.0 Knowledge base instead of searching for each attribute in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(IntegratedSoftwareManagement - BadIntegratedSoftwareManagement)}{\text{Integrated Software Management}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA10-5AlmostAlways
  (Integrated Software Management, ?x)
  (Bad Integrated Software Management, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA10-5 AlmostAlways))
  (printout Result.txt " KPA 10.5 Are measurements used to determine the effectiveness of the integrated software management activities (e.g., frequency, causes and magnitude of replanning efforts)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to integrated software management **Activit(y)**ies and the attributes of schedule and the cost of Integrated Software Management **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Ingegrated Software Management should be predefined and recongnized in the knowledge base.

KPA 10.6 Are the activities and work products used to manage the software project subjected to SQA review and audit?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	ISM Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			ISM Program Review

2) The heuristics used to make an assessment based on that data

The fact *ISM Review, ISM Reviewer* are used in the JESS rule. *ISM Review* counts the number of **Activit(y)**ies related to software project management. *ISM Reviewer* counts the number of **Individual**s who are assigned to review software project management **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software project management review and **Individual**s to review software project management **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA10-6Yes
  (ISM Review, ?x)
  (ISM Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
  (assert (KPA10-6 Yes))
  (printout Result.txt "KPA 10.6 Are the activities and work products used to manage the software project
subjected to SQA review and audit? Yes." crlf))

(defrule KPA10-6No
  (ISM Review, ?x)
  (ISM Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
```

(assert (KPA10-6 No))

(printout Result.txt "KPA 10.6 Are the activities and work products used to manage the software project subjected to SQA review and audit? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of software project management review **Activit(y)**ies and the assigned **Individual**s (SQA group) to review software project management **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Ingegrated Software Management should be predefined and recongnized in the knowledge base.

L. KPA 11. Software Product Engineering

The purpose of Software product Engineering is to consistently perform a well-defined engineering process that integrates all the software engineering activities to product correct, consistent software products effectively and efficiently [28].

KPA 11.1 Are the software work products produced according to the project's defined software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***Training***$_{set}$(audience, objectives, length, lesson plans)	**Organization** contains **Individual**s	Project Planning Tool (MS Project)	Tools
VAndVTest	***CoverageVector(*** % by source)	**VAndVtest**s are related to **Feature**s		Testing
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase) & Design Tool (Rational Rose)	SPE Document, Design
Defect	***Name***$_{set}$, ***Description***$_{set}$, ***Severity***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	Defect
WorkBreak	***Name***$_{set}$			Activity From

down Strucuture				Requirement
Feature	***Name***$_{set}$	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	
Activity	***Name***$_{set}$		Project Planning Tool (MS Project)	Activity From Plan
WorkBreak down Strucuture	***Name***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	Requirement

2) The heuristics used to make an assessment based on that data

The facts *Tools, Testing, SPE Document, Design, Defect, Activity From Requirement, Activity From Plan,* and *Requirement* are used in the JESS rule. *Tools* checks the number of software engineering tools. *Testing* checks the existence of integration testing, system and acceptance testing. *SPE Document* checks the number of document related to software process definition. *Design* checks the number of design attributes. *Defect* checks the number of defect attributes. *ActivityFromRequirement* counts the number of **WorkBreakdownStructure** generated by the requirement. *Activity-Plan* counts the number of **WorkBreakdownStructure** in ClearQeust and the related **Activit(y)**ies in MS Project. The *requirement_number* counts the number of requirements in RequisitePro. For each Requirement, there should be related **WorkBreakdownStructure** and **Activity** that produces **Artifact**. Requirements from RequisitePro link to ClearQuest **WorkBreakdownStructure**, which is an assigned work to each **Individual**. And the **WorkBreakdownStructure** from ClearQuest is linked to MS Project, which is an **Activity** planning tool, as **Activity**. Calculate the requirement ratio to **WorkBreakdownStructure**, and **Activity.** These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\left(\frac{ActivityFrom\mathrm{Re}quirement}{requirement\_number}+\frac{Activity-Plan}{requirement\_number}\right)\times\frac{1}{2}$ is more than 0.9, it means 90 % of the requirements have generated related **Activit(y)**ies and **Plan**s. It also checks each of the *Tools, Testing, SPE Document, Design,* and *Defect* is more than 1, which checks the existence of software engineering tools, attributres related to requirements management software desingn, testing, defect identification and documentations describing the software process.

```
(defrule KPA11-1AlmostAlways

 (ActivityFromRequirement, ?x)   (Activity-Plan, ?z)  (requirement_number, ?y)
```

```
(Tools, ?s) (Testing, ?t) (SPE Document, ?w ) (Desing, ?v ) (Defect, ?u )
(test  (and (<= 1 ?s) (<= 1 ?t) (<= 1 ?u) (<= 1 ?v) (<= 1 ?w) (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?z ?y))))))
 =>
(assert (KPA11-1 AlmostAlways))
(printout Result.txt " KPA 11.1 Are the software work products produced according to the project's
defined software process? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that it should have appropriate software engineering methods and tools, requirements are managed, software desingn is managed, software code is developed from requirements and design, testing is managed including integration testing, system and acceptance testing, documentation, defect identification [28]. The heuristics checks the existence of software engineering tools, attributres related to requirements management software desingn, testing, defect identification and documentations describing the software process.

4) The limitations of the data and heuristics

The documents related to Software Product Engineering should be predefined and recongnized in the knowledge base.

KPA 11.2 Is consistency maintained across software work products (e.g., is the documentation tracing allocated requirements through software requirements, design, code, and test cases maintained)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
VAndVTest	***CoverageVector(***% by source), ***Name***$_{set}$	**VAndVtest**s are related to **Feature**s		Testing
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase) & Design Tool (Rational Rose)	Design
WorkBreak down Strucuture	***Name***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	Activity From Requirement
Feature	***Name***$_{set}$	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	
Activity	***Name***$_{set}$		Project	Activity

			Planning Tool (MS Project)	From Plan
WorkBreak down Strucuture	***Name***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	Requirement

2) The heuristics used to make an assessment based on that data

The facts *Testing, Design, Defect, Activity From Requirement, Activity From Plan,* and *Requirement* are used in the JESS rule. *Testing* checks the existence of integration testing, system and acceptance testing for the **Artifact**s created. *Design* checks the number of design attributes related to the requirement. *ActivityFromRequirement* counts the number of **WorkBreakdownStructure** generated by the requirement. *Activity-Plan* counts the number of **WorkBreakdownStructure** in ClearQeust and the related **Activit(y)**ies in MS Project. The *requirement_number* counts the number of requirements in RequisitePro. For each Requirement, there should be related **WorkBreakdownStructure** and **Activity** that produces **Artifact**. Requirements from RequisitePro link to ClearQuest **WorkBreakdownStructure**, which is an assigned work to each **Individual**. And the **WorkBreakdownStructure** from ClearQuest is linked to MS Project, which is an **Activity** planning tool, as **Activity**. Calculate the requirement ratio to **WorkBreakdownStructure**, and **Activity.** These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{ActivityFrom\mathrm{Re}quirement}{requirement\_number} + \frac{Activity - Plan}{requirement\_number}) \times \frac{1}{2}$ is more than 0.9, which means 90 % of the requirements have generated related **Activit(y)**ies and **Plan**s. It also checks each of the *Testing,* and *Design* is more than 1, which means there exist **Artifact-related** test and requirement related design.

```
(defrule KPA11-2AlmostAlways
 (ActivityFromRequirement, ?x)   (Activity-Plan, ?z)  (requirement_number, ?y)
 (Testing, ?v) (Desing, ?w )
 (test  (and  (<= 1 ?v) (<= 1 ?w) (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?z ?y))))))
  =>
 (assert (KPA11-2 AlmostAlways))
 (printout Result.txt " KPA 11.2 Is consistency maintained across software work products (e.g., is the
documentation tracing allocated requirements through software requirements, design, code, and test cases
maintained)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The heuristics above checks the existence of attributres related to requirement, software desingn, activity, and testing.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Product Engineering should be predefined and recongnized in the knowledge base.

KPA 11.3 Does the project follow a written organizational policy for performing the software engineering activities (e.g., a policy which requires the use of appropriate methods and tools for building and maintaining software products)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SPE Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			SPE Tool
Activity	**InitialMilestone, FinalMilestone,**			SPE Activity

	Name$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SPE Document

2) The heuristics used to make an assessment based on that data

The facts *SPE Group, SPE Tool, SPE Activity,* and *SPE Document* are used in the JESS rule. *SPE Group* counts the number of responsible group or **Individual**s assigned to performing the software engineering **Activit(y)**ies. *SPE Activity* counts the number of **Activit(y)**ies related to performing the software engineering. *SPE Tool* counts the number of tools to use for performing the software engineering. *SPE Document* counts the number of documents related to performing the software engineering **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the

PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *SPE Group, SPE Tool, SPE Activity,* and *SPE Document* is more than 1, which means performing the software engineering related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA11-3Yes
  (SPE Group, ?w) (SPE Tool, ?x) (SPE Activity, ?y) (SPE Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA11-3 Yes))
  (printout Result.txt " KPA 11.3 Does the project follow a written organizational policy for performing the
software engineering activities (e.g., a policy which requires the use of appropriate methods and tools for
building and maintaining software products)? Yes." crlf))

(defrule KPA11-3No
  (SPE Group, ?w) (SPE Tool, ?x) (SPE Activity, ?y) (SPE Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA11-3 No))
  (printout Result.txt " KPA 11.3 Does the project follow a written organizational policy for performing the
software engineering activities (e.g., a policy which requires the use of appropriate methods and tools for
building and maintaining software products)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Product Engineering should be predefined and recongnized in the knowledge base.

KPA 11.4 Are adequate resources provided for performing the software engineering tasks (e.g., funding, skilled individuals, and appropriate tools)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	***ActualCost***$_{set}$, ***PlannedCost***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	SE Funds satisfied, Number of completed SE task

Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$			Sufficient trained Individual, Number of Individual
Organization	***Name***$_{set}$, ***Description***$_{set}$ ***SoftwareTools***$_{set}$			Tool

2) The heuristics used to make an assessment based on that data

The facts *SE Funds satisfied, Number of completed SE task, Sufficient trained Individual, Number of Individual,* and *Tool* are used in the JESS rule. *SE Funds satisfied* counts the number of **ActualCost**$_{set}$, which is less than the **PlannedCost**$_{set}$ in Software Product Engineering **Activit(y)**ies. *Number of completed SE task* counts the number of completed Software Product Engineering **Activit(y)**ies. *Sufficient trained Individual* counts the number of **Individual**s who have sufficient experience in Software Product Engineering. *Number of Individual* counts the number of **Individual**s in the project. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $(\frac{\text{Funds satisfied}}{\text{Number of completed task}} + \frac{\text{Sufficient trained Individual}}{\text{Number of Individual}}) \times \frac{1}{2}$ is more than 0.9, it means 90 % of the **Activit(y)**ies are provided with sufficient funding and the assigned **Individual**s have sufficient experience. It also checks *Tool* is more than 1, which means the organization has appropriate software tools.

```
(defrule KPA11-4AlmostAlways
  (SEFunds satisfied, ?x)
  (Number of completed SE task, ?y)
  (Sufficient trained Individual, ?z)
  (Number of Individual, ?w)
  (tool, ?v)
  (test (and (= 1 ?v) (<= 0.9 (/ 2 (+ (/ ?x ?y) (/ ?z ?w))))))
   =>
  (assert (KPA11-4 AlmostAlways))
  (printout Result.txt "KPA 11.4 Are adequate resources provided for performing the software engineering tasks (e.g., funding, skilled individuals, and appropriate tools)? AlmostAlways." crlf))
  ----------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

There are sufficient and insufficient **Individual** experiences. We need to know the ratio of sufficient and insufficient **Individual** experiences to analyze the **Individual** experience level of the project. By comparing $\textbf{ActualCost}_{\texttt{set}}$ to $\textbf{PlannedCost}_{\texttt{set}}$, we can analyze the fund status of a project. It also checks the existence of software tools.

4) The limitations of the data and heuristics

None

KPA 11.5 Are measurements used to determine the functionality and quality of the software products (e.g., numbers, types, and severity of defects identified)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Defect	***Name***$_{set}$, ***escription***$_{set}$, ***Severity***$_{set}$		Defect Tracking Tool (ClearQuest)	Defect Identification

2) The heuristics used to make an assessment based on that data

The fact *Defect Identification* is used in the JESS rule. *Defect Identification* counts the number of attributes used to determine the functionality and quality of the software products such as numbers, types, and severity of defects. These fact-related attributes are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks *Defect Identification* is more than 1, which means attributes of numbers, types, and severity of defects are exist.

```
(defrule KPA11-5Yes
  (Defect Identification, ?x)
  (test (<= 1 ?x ))
```

=>

(assert (KPA11-5 Yes))

(printout Result.txt "KPA 11.5 Are measurements used to determine the functionality and quality of the software products (e.g., numbers, types, and severity of defects identified)? Yes." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that, measurements are made and used to determine the functionality and quality of the software products [28]. The heuristics checks the existence of attributes used to determine the functionality and quality of the software products such as numbers, types, and severity of defects.

4) The limitations of the data and heuristics

None

KPA 11.6 Are the activities and work products for engineering software subjected to SQA reviews and audits (e.g., is required testing performed, are allocated requirements traced through the software requirements, design, code and test cases)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$,	**Individual**	Project	SE Reviewer

	Title$_{set}$	owns **Activit(y)**ies.	Planning Tool (MS Project)	
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			SE Review

2) The heuristics used to make an assessment based on that data

The fact *SE Review,* and *SE Reviewer* are used in the JESS rule. *SE Review* counts the number of **Activit(y)**ies related to software engineering. *SE Reviewer* counts the number of **Individual**s who are assigned to review software engineering **Activit(y)**ies. The fact-related attribnutes are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software engineering review and **Individual**s to review software engineering **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA11-6Yes
  (SE Review, ?x)
  (SE Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA11-6 Yes))
  (printout Result.txt " KPA 11.6 Are the activities and work products for engineering software subjected to SQA reviews and audits (e.g., is required testing performed, are allocated requirements traced through the software requirements, design, code and test cases)? Yes." crlf))

(defrule KPA11-6No
  (SE Review, ?x)
  (SE Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA11-6 No))
  (printout Result.txt "KPA 11.6 Are the activities and work products for engineering software subjected to SQA reviews and audits (e.g., is required testing performed, are allocated requirements traced through the software requirements, design, code and test cases)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of software engineering review **Activit(y)**ies and the assigned **Individual**s (SQA group) to review software engineering **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Software Product Engineering should be predefined and recongnized in the knowledge base.

M. KPA 12. Intergroup Coordination

The purpose of Intergroup Coordination is to establish a means for the software engineering group to participate actively with the other engineering groups so the project is better able to satisfy the customer's needs effectively and efficiently [28].

KPA 12.1 On the project, do the software engineering group and other engineering groups collaborate with the customer to establish the system requirements?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Feature	***Name***$_{set}$, ***Status***$_{set}$(Proposed, Approved, Incorporated, Validated)	**Feature**s are related to **Artifact**s.	Requirements management Tool (RequisitePro)	IC Requirement
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile,		Configuration Management	IC Document

	SourceFile, Other), ***Size***$_{set}$		Tool (ClearCase)	

2) The heuristics used to make an assessment based on that data

The facts *IC Requirement* and *IC Document* are used in the JESS rule. *IC Requirement* checks the existence of attributes related to the **Feature** status such as proposed, approved, incorporated, validated. *IC Document* counts the number of documents, which includes the acceptance criteris for each product delivered to the customer or end user. The fact-related attributes are gathered automatically by querying attributes from PAMPA 2.0 Knowledge base instead of searching for each attribute in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *IC Requirement* and *IC Document* is more than 1, which means software engineering group and other engineering groups collaborate with the customer to establish the system requirements.

```
(defrule KPA12-1Yes
 (IC Requirement, ?x) (IC Document, ?y)
  (test (and (< 0 ?x )(< 0 ?y )))
  =>
 (assert (KPA12-1 Yes))
 (printout Result.txt " KPA 12.1 On the project, do the software engineering group and other engineering
groups collaborate with the customer to establish the system requirements? Yes." crlf))

(defrule KPA12-1No
```

```
 (IC Requirement, ?x) (IC Document, ?y)
  (test (or (= 0 ?x )(= 0 ?y )))
  =>
 (assert (KPA12-1 No))
 (printout Result.txt " KPA 12.1 On the project, do the software engineering group and other engineering
groups collaborate with the customer to establish the system requirements? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be customer's and end users' requirements, and agreement for the requirement between engineering groups and customer, and document, which includes the acceptance criteris for each product delivered to the customer or end user [28]. The heuristics checks the existence of attributes related to the **Feature** status such as proposed, approved, incorporated, validated. It also checks the number of documents, which includes the acceptance criteris for each product delivered to the customer or end user.

4) The limitations of the data and heuristics

None

KPA 12.2 Do the engineering groups agree to the commitments as represented in the overall project plan?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$***, Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	IC Reviewer
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$***,*** ***Dependency***$_{set}$***,*** ***Description***$_{set}$***,*** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			Dependency, IC Review

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	IC Document

2) The heuristics used to make an assessment based on that data

The facts *IC Review, IC Reviewer, Dependency,* and *IC Document* are used in the JESS rule . *IC Review* counts the number of **Activit(y)**ies related to review work products produced as input to other engineering groups to ensure that the work produdcts meet their needs. *IC Reviewer* counts the number of **Individual**s who are assigned to review intergroup coordination **Activit(y)**ies. *Dependency* checks the existence of attributes related to cirtical dependencies between engineering groups. *IC Document* counts the number of a documented plan that is used to commumnicate intergroup commitments and to coordinate and track the work performed. These facts are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching for each attributes in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *IC Review, IC Reviewer, Dependency,* and *IC Document* is more than 1, which means whether there are any **Activit(y)**ies related to work products review and **Individual**s to review work products, cirtical dependency agreement, and documented plan. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA12-2Yes
  (IC Review, ?w) (IC Reviewer, ?x) (Dependency, ?y) (IC Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA12-2 Yes))
  (printout Result.txt " KPA 12.2 Do the engineering groups agree to the commitments as represented in
the overall project plan? Yes." crlf))

(defrule KPA12-2No
  (IC Group, ?w) (IC Tool, ?x) (IC Activity, ?y) (IC Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA2-2 No))
  (printout Result.txt " KPA 12.2 Do the engineering groups agree to the commitments as represented in
the overall project plan? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be a documented plan that is used to commumnicate intergroup commitments and to coordinate and track the work performed. Cirtical dependencies between engineering groups are indentified, negotiated, and tracted. And work products produced as input to other engineering groups are reviewed by representatives of the receiving groups to ensure that the work produdcts meet their needs [28]. It checks the existence of **Activit(y)**ies related to work products review and **Individual**s to review work products, cirtical dependency agreement, and documented plan.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Intergroup Coordination should be predefined and recongnized in the knowledge base.

KPA 12.3 Do the engineering groups identify, track, and resolve intergroup issues (e.g., incompatible schedules, technical risks, or system-level problems)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	IC Reviewer, IC Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Dependency***$_{set}$,			IC Review

	Description$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The facts *IC Review, IC Reviewer,* and *IC Group* are used in the JESS rule. *IC Review* counts the number of **Activit(y)**ies related to review work products produced as input to other engineering groups to ensure that the work produdcts meet their needs. *IC Reviewer* counts the number of **Individual**s who are assigned to review intergroup coordination **Activit(y)**ies. *IC Group* checks the existence of representatives of the project's software engineering group and the other engineering groups. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *IC Review, IC Reviewer,* and *IC Group* is more than 1, which means whether there are any **Activit(y)**ies related to work products review and **Individual**s to review work products, and representatives of other groups. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA12-3Yes
  (IC Review, ?x) (IC Reviewer, ?y) (IC Group, ?z)
   (test (and (< 0 ?x ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA12-3 Yes))
  (printout Result.txt " KPA 12.3 Do the engineering groups identify, track, and resolve intergroup issues
(e.g., incompatible schedules, technical risks, or system-level problems)? Yes." crlf))

(defrule KPA12-3No
  (IC Review, ?x) (IC Reviewer, ?y) (IC Group, ?z)
   (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA2-3 No))
  (printout Result.txt " KPA 12.3 Do the engineering groups identify, track, and resolve intergroup issues
(e.g., incompatible schedules, technical risks, or system-level problems)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the representatives of the project's software engineering group work with representatives of the other engineering groups. Intergroup issues not resolvable by the individual representatives of the project engineering groups are handled according to a documented procedure. And representatives of the project engineering groups conduct periodic technical reviews and interchanges [28]. The heuristics checks whether there are any **Activit(y)**ies related to work products review and **Individual**s to review work products, and representatives of other groups.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Intergroup Coordination should be predefined and recongnized in the knowledge base.

KPA 12.4 Is there a written organizational policy that guides the establishment of interdisciplinary engineering teams?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***set, ***Description***set ,***Type***set(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	IC Group
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***set			IC Tool
Activity	**InitialMilestone, FinalMilestone,**			IC Activity

	Name$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	IC Document

2) The heuristics used to make an assessment based on that data

The facts *IC Group, IC Tool, IC Activity,* and *IC Document* are used in the JESS rule. *IC Group* counts the number of responsible group or **Individual**s assigned to establish interdisciplinary engineering teams. *IC Activity* counts the number of **Activit(y)**ies related to establish interdisciplinary engineering teams. *IC Tool* counts the number of tools to use for establishment of interdisciplinary engineering teams. *IC Document* counts the number of documents related to establish interdisciplinary engineering teams. The fact-related attributes are gathered automatically by querying attributes from the

PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *IC Group, IC Tool, IC Activity,* and *IC Document* is more than 1, which means intergroup coordination group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA12-4Yes
  (IC Group, ?w) (IC Tool, ?x) (IC Activity, ?y) (IC Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA12-4 Yes))
  (printout Result.txt " KPA 12.4 Is there a written organizational policy that guides the establishment of
interdisciplinary engineering teams? Yes." crlf))

(defrule KPA12-4No
  (IC Group, ?w) (IC Tool, ?x) (IC Activity, ?y) (IC Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA2-4 No))
  (printout Result.txt " KPA 12.4 Is there a written organizational policy that guides the establishment of
interdisciplinary engineering teams? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Intergroup Coordination should be predefined and recongnized in the knowledge base.

KPA 12.5 Do the support tools used by different engineering groups enable effective communication and coordination (e.g., compatible word processing systems, database systems, and problem tracking systems)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***SoftwareTools***$_{set}$, ***Type***$_{set}$(W<u>ord Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)</u>			Tool Compatible, Number of Tools

2) The heuristics used to make an assessment based on that data

The fact *Tool Compatible, Number of Tools* are used in the JESS rule. The *Tool Compatible* checks the number of compatible tools. The *Number of Tools* checks the number of tools used in the project. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the result below. If $\frac{\text{Tool Compatible}}{\text{Number of Tools}}$ is more than 0.9, it means 90 % of the Tools are compatible.

```
(defrule KPA12-5AlmostAlways
  (Tool Compatible, ?x)
  (Number of Tools, ?y)
 (test  (<= 0.9  (/ ?x  ?y) )
   =>
  (assert (KPA12-5 AlmostAlways))
  (printout Result.txt "KPA 12.5 Do the support tools used by different engineering groups enable effective communication and coordination (e.g., compatible word processing systems, database systems, and problem tracking systems)? AlmostAlways" crlf))
----------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the support tools used by the different engineering groups should be compatible to enable effective communication and coordination [28]. The heuristics checks the tool compatibility.

4) The limitations of the data and heuristics

None

KPA 12.6 Are measures used to determine the status of the intergroup coordination activities (e.g., effort expended by the software engineering group to support other groups)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Type*$_{set}$**(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM),</u> ***Description*$_{set}$, Complete$_{set}$**(Yes=1, No=blank or 0),		Project Planning Tool (MS Project)	Intergroup Coordination Activity, Bad Intergroup Coordination Activity

	ActualCost$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Intergroup Coordination Activity*, and *Bad Intergroup Coordination Activity* are used in the JESS rule. *Intergroup Coordination Activity* counts the number of **Activit(y)**ies related to intergroup coordination. *Bad Intergroup Coordination Activity* counts the number of **Activit(y)**ies, which do not have proper attributes for measuring effort expended for performing intergroup coordination. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{Intergroup Coordination Activity} - \textit{Bad}\text{Intergroup Coordination Activity})}{\text{Intergroup Coordination Activity}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA12-6AlmostAlways
  (Intergroup Coordination Activity, ?x)
```

(Bad Intergroup Coordination Activity, ?y)

(test (<= 0.9 (/ (- ?x ?y) ?x)))

=>

(assert (KPA12-6 AlmostAlways))

(printout Result.txt "KPA 12.6 Are measures used to determine the status of the intergroup coordination activities (e.g., effort expended by the software engineering group to support other groups)? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies performed to intergroup coordination and the attributes of effort expended for intergroup coordination **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Intergroup Coordination should be predefined and recongnized in the knowledge base.

KPA 12.7 Are the activities for intergroup coordination reviewed with the project manager on both a periodic and event-driven basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$,	**Individual**	Project	IC Reviewer

	LastName$_{set}$, ***Title***$_{set}$	owns **Activit(y)**ies.	Planning Tool (MS Project)	
Organization	***Name***$_{set}$***,*** ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$***,*** ***Description***$_{set}$***,*** **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			IC Review

2) The heuristics used to make an assessment based on that data

The fact *IC Review,* and *IC Reviewer* are used in the JESS rule. *IC Review* counts the number of **Activit(y)**ies related to intergroup coordination. *IC Reviewer* counts the number of **Individual**s who are assigned to review intergroup coordination **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to intergroup coordination review and **Individual**s to review intergroup coordination **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA12-7Yes
  (IC Review, ?x) (IC Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA12-7 Yes))
  (printout Result.txt " KPA 12.7 Are the activities for intergroup coordination reviewed with the project manager on both a periodic and event-driven basis? Yes." crlf))

(defrule KPA12-7No
  (IC Review, ?x) (IC Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA12-7 No))
  (printout Result.txt " KPA 12.7 Are the activities for intergroup coordination reviewed with the project manager on both a periodic and event-driven basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of intergroup coordination review **Activit(y)**ies and the assigned **Individual**s (Project Manager) to review intergroup coordination **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Intergroup Coordination should be predefined and recongnized in the knowledge base.

N. KPA 13. Peer Reviews

The purpose of Peer Reviews is to remove defects from the software work products early and efficiently. An important corollary effect is to develop a better understanding of the software work products and of defects that might be prevented [28].

KPA 13.1 Are peer reviews planned?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	PeerReview Document

2) The heuristics used to make an assessment based on that data

The fact *PeerReviewDocument* is used in the JESS rule. *PeerReviewDocument* counts the number of documents related to Peer Review. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *PeerReviewDocument* is more than 1, which means the document related to Peer Review is exist.

```
(defrule KPA13-1Yes
 (PeerReviewDocument, ?x)   (test (<= 1  ?x ))
  =>
 (assert (KPA13-1 Yes))
 (printout Result.txt " KPA 13.1 Are peer reviews planned? Yes." crlf))

(defrule KPA13-1No
(PeerReviewDocument, ?x)
 (test (= 0  ?x ))
  =>
 (assert (KPA13-1 No))
 (printout Result.txt " KPA 13.1 Are peer reviews planned? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document related to Peer Review states the policy for performing peer reviews. By checking the existence of Peer Review document, we can assume that the project tries to follow a written organizational policy for performing peer reviews.

4) The limitations of the data and heuristics

The documents related to Peer Reviews should be predefined and recongnized in the knowledge base.

KPA 13.2 Are actions associated with defects that are identified during peer reviews tracked until they are resolved?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase)	PR Document
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0),**ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	PR Activity
Defect	***Name***$_{set}$, ***Description***$_{set}$, ***Severity***$_{set}$		Activity & Defect Tracking	Defect

			Tool (ClearQuest)	
Rework	**Adds, Changes, Deletes**	**Rework** contains attributes of an **Artifact**.	Configuration Management Tool (ClearCase)	Rework effort
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	PR Group
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***ComputerResource***$_{set}$, ***ToolCompatability*** $_{set,}$ ***TrainingFacilities***$_{set,}$ ***CourseReview***$_{set}$(**Individual**, software managers) ***Training***$_{set}$(audience, objectives, length, lesson plans)	**Organization** contains **Individual**s		

2) The heuristics used to make an assessment based on that data

The facts *PR Group, PR Activity, Rework Effort, PR Document* and *Defect* are used in the JESS rule. *PR Group* counts the number of responsible group or **Individual**s assigned to peer reviews. *PR Activity* checks the existence of attributes related to peer reviews. *Rework* checks the existence of attributes to measure rework effort. *SPP Document* counts the number of documents related to peer reviews. *Defect* checks the existence of types and number of defects found and fixed. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PR Group, PR Activity, Rework Effort, PR Document*, and *Defect* is more than 1, which means peer review related group or **Individual**s, **Activit(y)**ies, document, and defect are exist.

```
(defrule KPA13-2Yes
  (PR Group, ?w) (PR Activity, ?x) (Rework Effort, ?y) (PR Document, ?z) (Defect, ?v)
   (test (and (< 0 ?v ) (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA13-2 Yes))
  (printout Result.txt " KPA 13.2 Are actions associated with defects that are identified during peer reviews tracked until they are resolved? Yes." crlf))

(defrule KPA13-2No
  (PR Group, ?w) (PR Activity, ?x) (Rework Effort, ?y) (PR Document, ?z) (Defect, ?v)
   (test (or (= 0 ?v ) (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
```

```
(assert (KPA13-2 No))
(printout Result.txt " KPA 13.2 Are actions associated with defects that are identified during peer reviews
tracked until they are resolved? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Matuirity Model states that the peer reviews are performed according to a documented procedure. Data on the conduct and results of the peer reviews are recorded [28]. The heuristics cheks the existence of peer review related group or **Individual**s, **Activit(y)**ies, document, and defect.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Peer Reviews should be predefined and recongnized in the knowledge base.

KPA 13.3 Does the project follow a written organizational policy for performing peer reviews?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	PR Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			PR Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			PR Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile,	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PR Document

	SourceFile, Other)			

2) The heuristics used to make an assessment based on that data

The facts *PR Group, PR Tool, PR Activity,* and *PR Document* are used in the JESS rule. *PR Group* counts the number of responsible group or **Individual**s assigned to peer reviews. *PR Activity* counts the number of **Activit(y)**ies related to peer reviews. *SPP Tool* counts the number of tools to use for peer reviews. *SPP Document* counts the number of documents related to peer reviews. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PR Group, PR Tool, PR Activity,* and *PR Document* is more than 1, which means peer review related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA13-3Yes
  (PR Group, ?w) (PR Tool, ?x) (PR Activity, ?y) (PR Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA13-3 Yes))
  (printout Result.txt " KPA 13.3 Does the project follow a written organizational policy for performing
peer reviews? Yes." crlf))
```

```
(defrule KPA13-3No
  (PR Group, ?w) (PR Tool, ?x) (PR Activity, ?y) (PR Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA13-3 No))
  (printout Result.txt " KPA 13.3 Does the project follow a written organizational policy for performing
peer reviews? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Peer Reviews should be predefined and recongnized in the knowledge base.

KPA 13.4 Do participants of peer reviews receive the training required to perform their roles?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	Peer Review Education

2) The heuristics used to make an assessment based on that data

The fact *Peer Review Education* is used in the JESS rule. The *Peer Review Education* checks whether the **Individual**s, who participate in peer reviewing, have sufficient experiences. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Peer Review Education* is 1, which means the **Individual**s, who participate in peer reviewing, have sufficient experiences.

```
(defrule KPA13-4Yes
(Peer Review Education, ?x)
(test (= 1  ?x ))
 =>
(assert (KPA13-4 Yes))
(printout Result.txt " KPA 13.4 Do participants of peer reviews receive the training required to perform their roles? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the peer review leaders receive required training in how to lead peer reviews [28]. By checking the training experience of **Individual**s, who participate in peer reviewing, have sufficient experiences, we can understand whether the **Individual**s, who participate in peer reviewing, receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 13.5 Are measurements used to determine the status of peer review activities (e.g., number of peer reviews performed, effort expended on peer reviews, and number of work products reviewed compared to the plan)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Type*$_{set}$**(RM, SPP, SPTO, SSM, SQA, SCM, OPF,		Project Planning Tool (MS Project)	Peer Review Activity, Bad Peer Review Activity

	OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
InitialMilestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**InitialMilestone** is an attribute of **Activity**.		
FinalMilstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Peer Review Activity* and *Bad Peer Review Activity* are used in the JESS rule. *Peer Review Activity* counts the number of **Activit(y)**ies related to peer review. *Bad Peer Review* counts the number of peer review **Activit(y)**ies, which do not have proper attributes of schedule and cost. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result $\frac{(\text{PeerReviewActivity} - \textit{Bad}\text{PeerReviewActivity})}{\text{PeerReviewActivity}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA13-5AlmostAlways
  (Peer Review Activity, ?x)
  (Bad Peer Review Activity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA13-5 AlmostAlways))
  (printout Result.txt " KPA 13.5 Are measurements used to determine the status of peer review activities (e.g., number of peer reviews performed, effort expended on peer reviews, and number of work products reviewed compared to the plan)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to peer review **Activit(y)**ies and the attributes of schedule and the cost to determine the number of peer reviews performed, effort expended on peer reviews, and number of work products reviewed compared to the plan.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Peer Reviews should be predefined and recongnized in the knowledge base.

KPA 13.6 Are peer review activities and work products subjected to SQA review and audit (e.g., planned reviews are conducted and follow-up actions are tracked)?

1) The data available related to the CMM analysis

<table>
<tr><th colspan="3">PAMPA 2.0</th><th rowspan="2">Source</th><th rowspan="2">Facts elicited</th></tr>
<tr><th>Object</th><th>Attributes</th><th>Relationships</th></tr>
<tr><td>Individual</td><td>FirstName$_{set}$, LastName$_{set}$, Title$_{set}$</td><td>Individual owns Activit(y)ies.</td><td rowspan="4">Project Planning Tool (MS Project)</td><td rowspan="2">PR Reviewer</td></tr>
<tr><td>Organization</td><td>Name$_{set}$, Description$_{set}$</td><td>Organization contains Individuals</td></tr>
<tr><td>Activity</td><td>InitialMilestone, FinalMilestone, Name$_{set}$, Description$_{set}$, Complete$_{set}$(Yes=1, No=blank or 0), Type$_{set}$(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM</u>)</td><td></td><td>PR Review</td></tr>
<tr><td>Artifact</td><td>Name$_{set}$,Type(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)</td><td></td><td>PR Document</td></tr>
</table>

2) The heuristics used to make an assessment based on that data

The fact *PR Review, PR Reviewer,* and *PR Document* are used in the JESS rule. *PR Review* counts the number of **Activit(y)**ies related to peer review. *PR Reviewer* counts the number of **Individual**s who are assigned to review peer review **Activit(y)**ies. *PR Document* counts the number of peer review related **Artifact**s. These fact-related attributes are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies, **Artifact**s related to peer review, and **Individual**s to review peer review **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA13-6Yes
  (PR Review, ?x) (PR Reviewer, ?y) (PR Document, ?z)
  (test (and(<= 1 ?x) (<= 1 ?y) (<= 1 ?z) ))
   =>
  (assert (KPA13-6 Yes))
(printout Result.txt " KPA 13.6 Are peer review activities and work products subjected to SQA review and audit (e.g., planned reviews are conducted and follow-up actions are tracked)? Yes." crlf))

(defrule KPA13-6No
  (PR Review, ?x) (PR Reviewer, ?y) (PR Document, ?z)
  (test (or(= 0 ?x) (= 0 ?y) (= 0 ?z)  ))
```

=>

(assert (KPA13-6 No))

(printout Result.txt " KPA 13.6 Are peer review activities and work products subjected to SQA review and audit (e.g., planned reviews are conducted and follow-up actions are tracked)? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of peer review **Activit(y)**ies, **Artifact**s and the assigned **Individual**s (SQA group) to review peer review **Activit(y)**ies, **Artifact**s.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Peer Reviews should be predefined and recongnized in the knowledge base.

O. KPA 14. Quantitative Process Management

The purpose of Quantitative Process management is to control the process performance of the software project quantitatively [28].

KPA 14.1 Does the project follow a documented plan for conducting quantitative process management?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	QPMDocument

2) The heuristics used to make an assessment based on that data

The fact *QPMDocument* is used in the JESS rule. *QPMDocument* counts the number of document such as "Quantitative Process Management Plan". The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *QPMDocument* is more than 1, which means the document "Quantitative Process Management Plan" is exist.

```
(defrule KPA14-1Yes
  (QPMDocument, ?x)   (test (<= 1  ?x ))
   =>
  (assert (KPA14-1 Yes))
  (printout Result.txt " KPA 14.1 Does the project follow a documented plan for conducting quantitative
process management? Yes." crlf))

(defrule KPA14-1No
 (QuantitativeProcessManagementDocument, ?x)
 (test (= 0  ?x ))
   =>
```

(assert (KPA14-1 No))

(printout Result.txt "KPA 14.1 Does the project follow a documented plan for conducting quantitative process management? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document "Quantitative Process Management Plan" states the policy for measuring and controlling the performance of the project's defined software process. By checking the existence of "Quantitative Process Management Plan" document, we can assume that the project tries to follow a written organizational policy for measuring and controlling the performance of the project's defined software process.

4) The limitations of the data and heuristics

The documents related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

KPA 14.2 Is the performance of the project's defined software process controlled quantitatively (e.g., through the use of quantitative analytic methods)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,** **FinalMilestone,**		Project Planning Tool	QPM Activity

	Name$_{set}$, ***Dependency***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		(MS Project)	
Defect	***Name***$_{set}$, ***Description***$_{set}$, ***Severity***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	QPM Measure-ment
Individual	***Experience***$_{set}$(software development years), ***TestResult***$_{set}$ ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan,	**Organization** contains **Individual**s		

	DB, Activity Tracking), ***ComputerResource***$_{set}$, ***ToolCompatability***$_{set}$, ***TrainingFacilities***$_{set}$, ***CourseReview***$_{set}$(**Individual**, software managers) ***Training***$_{set}$(audience, objectives, length, lesson plans)			
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Dependency***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0),**ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase)	QPM Document

2) The heuristics used to make an assessment based on that data

The facts *QPM Activity, QPM Measurement* and *QPM Document* are used in the JESS rule. *QPM Activity* counts the number of **Activit(y)**ies related to measuring and controlling the performance of the project's defined software process. *QPM Measurement* checks the existence of quatitative process management related attributes such as planned and actual data on software size, cost, and schedule, productivity data, training, test, and defect. *QPM Document* counts the number of documents related to software project's quantitative process management **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *QPM Activity, QPM Measurement* and *QPM Document* is more than 1, which means we can assume that the performance of the project's defined software process is controlled quantitatively.

```
(defrule KPA14-2Yes
 (QPM Measurement, ?x) (QPM Activity, ?y) (QPM Document, ?z)
  (test (and (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA14-2 Yes))
```

```
 (printout Result.txt " KPA 14.2 Is the performance of the project's defined software process controlled
quantitatively (e.g., through the use of quantitative analytic methods)? Yes." crlf))

(defrule KPA14-2No
 (QPM Measurement, ?x) (QPM Activity, ?y) (QPM Document, ?z)
  (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA14-2 No))
 (printout Result.txt " KPA 14.2 Is the performance of the project's defined software process controlled
quantitatively (e.g., through the use of quantitative analytic methods)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the software project's quantitative process management activities are performed in accordance with the project's quantitative process management plan. Second, the measurement data used to control the project's defined software process quantitatively are collected according to a documented procedure. Third, the project's defined software process is analyzed and brought under quantitative control. Forth, reports documenting the results of the software project's quantitative process management activities are prepared and distributed [28]. The heuristics above checks the existence of quantitative process management **Activit(y)**ies, measurement attributes such as planned and actual data on software size, cost, and schedule, productivity data, training, test, defect, and document related to software project's quantitative process management **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

KPA 14.3 Is the process capability of the organization's standard software process known in quantitative terms?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Dependency*$_{set}$, *Description*$_{set}$, Complete**$_{set}$(Yes=1, No=blank or 0),**ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	Process Database, Activity
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$,	**Initial-Milestone** is an		

	PlannedCost$_{set}$, ***ActualCost***$_{set}$	attribute of **Activity**.		
Final-Milstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		
Artifact	***Name***$_{set}$,**Type**(Requirements File, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase)	QPM Document
Defect	***Name***$_{set}$, ***Description***$_{set}$, ***Severity***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	Defect Trend

2) The heuristics used to make an assessment based on that data

The facts *Process Database, Activity, QPM Document* and *Defect Trend* are used in the JESS rule. *QPM Database* checks that existence of process related attributes. *Activity* checks the existence of attributes related to the reason of defects such as labor-intensive activities. *QPM Document* counts the number of documents related to process capability baseline. *Defect Trend* checks the existence of **Defect** prediction related attributes. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *Process Database, Activity, QPM Document,* and *Defect* is more than 1, which means we can assume that the process capability of the organization's standard software process is known in quantitative terms.

```
(defrule KPA14-3Yes
  (Process Database, ?w) (Activity, ?x) (QPM Document, ?y) (Defect, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA14-3 Yes))
  (printout Result.txt " KPA 14.3 Is the process capability of the organization's standard software process known in quantitative terms?  Yes." crlf))

(defrule KPA14-3No
  (Process Database, ?w) (Activity, ?x) (QPM Document, ?y) (Defect, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA14-3 No))
  (printout Result.txt " KPA 14.3 Is the process capability of the organization's standard software process known in quantitative terms?  No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be a database for project's software process data, a document about process capability baseline, defect prediction, Activity status [28]. The heuristics checks the existence of **Process** related attributes, a document related process capability baseline, **Defect** prediction related attributes, and **Activity** attributes.

4) The limitations of the data and heuristics

None

KPA 14.4 Does the project follow a written organizational policy for measuring and controlling the performance of the project's defined software process (e.g., projects plan for how to identify, analyze, and control special causes of variations)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	QPM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			QPM Tool
Activity	**InitialMilestone, FinalMilestone,**			QPM Activity

	Name$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	QPM Document

2) The heuristics used to make an assessment based on that data

The facts *QPM Group, QPM Tool, QPM Activity,* and *QPM Document* are used in the JESS rule. *QPM Group* counts the number of responsible group or **Individual**s assigned to measuring and controlling the performance of the project's defined software process. *QPM Activity* counts the number of **Activit(y)**ies related to measuring and controlling the performance of the project's defined software process. *QPM Tool* counts the number of tools to use for measuring and controlling the performance of the project's defined software process. *QPM Document* counts the number of documents related to measuring and controlling the performance of the project's defined software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *QPM Group, QPM Tool, QPM Activity,* and *QPM Document* is more than 1, which means quantitative process management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA14-4Yes
  (QPM Group, ?w) (QPM Tool, ?x) (QPM Activity, ?y) (QPM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA14-4 Yes))
  (printout Result.txt " KPA 14.4 Does the project follow a written organizational policy for measuring and
controlling the performance of the project's defined software process (e.g., projects plan for how to
identify, analyze, and control special causes of variations)? Yes." crlf))

(defrule KPA14-4No
  (QPM Group, ?w) (QPM Tool, ?x) (QPM Activity, ?y) (QPM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA14-4 No))
```

(printout Result.txt " KPA 14.4 Does the project follow a written organizational policy for measuring and controlling the performance of the project's defined software process (e.g., projects plan for how to identify, analyze, and control special causes of variations)? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

KPA 14.5 Are adequate resources provided for quantitative process management activities (e.g., funding, software support tools, and organizational measurement program)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	Funds satisfied Quantitative Process Management Activity, Number of completed Quantitative Process Management task
Organization	***Name***$_{set}$, ***Description***$_{set}$ ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$, ***TrainingFacilities***$_{set}$			Sufficient tools and measurement Program

2) The heuristics used to make an assessment based on that data

The facts *Funds satisfied Quantitative Process Management Activity, Number of completed Quantitative Process Management task,* and *Sufficient tools and measurement Program* are used in the JESS rule. *Funds satisfied Quantitative Process Management Activity* counts the number of **ActualCost**$_{set}$, which is less than the **PlannedCost**$_{set}$ in Quantitative Process Management related **Activit(y)**ies. *Number of completed Quantitative Process Management task* counts the number of completed Quantitative Process Management related **Activit(y)**ies. *Sufficient tools and measurement Program* counts the number of software tools, and organizational measurement program. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below.

If $\frac{\text{Funds satisfied Quantitative Process Management Activity}}{\text{Number of completed Quantitiativge Process Management task}}$ is more than 0.9, it means 90 % of the and Quantitative Process Management related **Activit(y)**ies are provided with sufficient funding and it checks *Sufficient tools and measurement Program* is more than 1, which means the organization has software tools, and organizational measurement program.

```
(defrule KPA14-5AlmostAlways
  (Funds satisfied Quantitative Process Management Activity, ?x)
  (Number of completed Quantitative Process Management task, ?y)
  (Sufficient tools and measurement Program, ?z)
(test (and (= 1 ?z) (<= 0.9  (/ ?x  ?y) ))
   =>
  (assert (KPA14-5 AlmostAlways))
  (printout Result.txt " KPA 14.5 Are adequate resources provided for quantitative process management activities (e.g., funding, software support tools, and organizational measurement program)? AlmostAlways" crlf))
----------------------
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the **ActualCost**$_{set}$ to **PlannedCost**$_{set}$ in Quantitative Process Management related **Activit(y)**ies and the existence of organization's software tools, and measurement program.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

KPA 14.6 Are measurements used to determine the status of the quantitative process management activities (e.g., cost of quantitative process management activities and accomplishment of milestones for quantitative process management activities)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM),		Project Planning Tool (MS Project)	Quantitative Process Management, Bad Quantitative Process Management

	Description$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			
Initial Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**Initial Milestone** is an attribute of **Activity**.		
Final Milstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**FinalMilestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Quantitative Process Management* and *Bad Quantitative Process Management* are used in the JESS rule. *Quantitative Process Management* counts the number of **Activit(y)**ies related to quantitative process management. *Bad Quantitative Process Management* counts the number of quantitative **Activit(y)**ies, which do not have proper attributes of schedule and cost. The fact-related attributres are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{QuantitativeProcessManagement} - \textit{Bad}\text{QuantitativeProcessManagement})}{\text{QuantitativeProcessManagement}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA14-6AlmostAlways
 ( Quantitative Process Management , ?x)
 ( Bad Quantitative Process Management , ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
 (assert (KPA14-6 AlmostAlways))
 (printout Result.txt "KPA 14.6 Are measurements used to determine the status of the quantitative process management activities (e.g., cost of quantitative process management activities and accomplishment of milestones for quantitative process management activities)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to quantitative process management **Activit(y)**ies and the attributes of schedule and the cost to determine cost of quantitative process management **Activit(y)**ies and accomplishment of milestones for quantitative process management **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

KPA 14.7 Are the activities for quantitative process management reviewed with the project manager on both a periodic and event-driven basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	PR Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			PR Review
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile,			PR Document

	SourceFile, Other)			

2) The heuristics used to make an assessment based on that data

The fact *PR Review, PR Reviewer,* and *PR Document* are used in the JESS rule. *PR Review* counts the number of **Activit(y)**ies related to peer review. *PR Reviewer* counts the number of **Individual**s who are assigned to review peer review **Activit(y)**ies. *PR Document* counts the number of peer review related **Artifact**s. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies, **Artifact**s related to peer review, and **Individual**s to review peer review **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA14-7Yes
  (PR Review, ?x)
  (PR Reviewer, ?y)
  (PR Document, ?z)
  (test (and(<= 1 ?x) (<= 1 ?y) (<= 1 ?z) ))
   =>
  (assert (KPA14-7 Yes))
(printout Result.txt " KPA 14.7 Are the activities for quantitative process management reviewed with the project manager on both a periodic and event-driven basis?Yes." crlf))

(defrule KPA14-7No
```

```
 (PR Review, ?x)
 (PR Reviewer, ?y)
 (PR Document, ?z)
 (test (or(= 0 ?x) (= 0 ?y) (= 0 ?z)  ))
  =>
 (assert (KPA14-7 No))
 (printout Result.txt KPA 14.7 Are the activities for quantitative process management reviewed with the project manager on both a periodic and event-driven basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of peer review **Activit(y)**ies, **Artifact**s and the assigned **Individual**s (SQA group) to review peer review **Activit(y)**ies, **Artifact**s.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Quantitative Process Management should be predefined and recongnized in the knowledge base.

P. KPA 15. Software Quality Management

The purpose of Software Quality Management is to develop a quantitative understanding of the quality of the project’s software propdcuts and achieve specific quality goals [28].

KPA 15.1 Are the activities for managing software quality planned for the project?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	SoftwareQuality Management Document

2) The heuristics used to make an assessment based on that data

The fact *SoftwareQualityManagementDocument* is used in the JESS rule. *SoftwareQualityManagementDocument* counts the number of document such as "Software Quality Management Plan". The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *SoftwareQualityManagementDocument* is more than 1, which means the document "Software Quality Management Plan" is exist.

```
(defrule KPA15-1Yes
  (SoftwareQualityManagementDocument, ?x)   (test (<= 1  ?x ))
   =>
  (assert (KPA15-1 Yes))
  (printout Result.txt " KPA 15.1 Are the activities for managing software quality planned for the project?
Yes." crlf))
```

```
(defrule KPA15-1No
 (SoftwareQualityManagementDocument, ?x)
 (test (= 0  ?x ))
   =>
  (assert (KPA15-1 No))
  (printout Result.txt " KPA 15.1 Are the activities for managing software quality planned for the project?
No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document "Software Quality Management Plan" states the policy for managing software quality. By checking the existence of "Software Quality Management Plan" document, we can assume that the project tries to follow a written organizational policy for managing software quality.

4) The limitations of the data and heuristics

The documents related to Software Quality Management should be predefined and recongnized in the knowledge base.

KPA 15.2 Does the project use measurable and prioritized goals for managing the quality of its software products (e.g., functionality, reliability, maintainability and usability)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
UsabilityTest	***Name***$_{set}$, ***Description***$_{set,}$ ***Date***$_{set}$, ***Status***$_{set}$(Failed/Passed, i.e.Usability Test is ready or not ready to be run)		Test Tool (Rationl Robot)	Usability
Defect	***Name***$_{set}$, ***Description***$_{set,}$ ***Identification***$_{set}$(number)			Reliability, Functionality

2) The heuristics used to make an assessment based on that data

The facts *Usability, Reliability,* and *Functionality* are used in the JESS rule. *Usability, Reliablity,* and *Functinality* count the number of usability, reliability and fucntionality result. These fact-related attributes are gathered automatically by querying attributes from PAMPA 2.0 Knowledge base instead of searching for each attribute in the project environment by human effort.

Below is the JESS rule to measure the questionnaire. It checks each of *Usability, Reliability,* and *Functionality* is more than 1, which means the project uses measurable and prioritized goals for managing the quality of its software products.

```
(defrule KPA15-2Yes
 (Usability, ?x) (Reliablity, ?y) (Functionality, ?z)
  (test (and (< 0 ?x ) (< 0 ?y )(< 0 ?z)))
```

```
  =>
  (assert (KPA15-2 Yes))
  (printout Result.txt " KPA 15.2 Does the project use measurable and prioritized goals for managing the quality of its software products (e.g., functionality, reliability, maintainability and usability)? Yes." crlf))

(defrule KPA15-2No
  (Usability, ?x) (Reliablity, ?y) (Functionality, ?z)
   (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA15-2 No))
  (printout Result.txt " KPA 15.2 Does the project use measurable and prioritized goals for managing the quality of its software products (e.g., functionality, reliability, maintainability and usability)? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the examples of software product quality charactersitcs include functionality, reliability, maintainability, and usability [28]. The heuristics checks the existence of functionality, reliability, and usability test result.

4) The limitations of the data and heuristics

None

KPA 15.3 Are measurements of quality compared to goals for software product quality to determine if the quality goals are satisfied?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$		Configuration Management Tool (ClearCase) Design Tool (Rational Rose)	SQM Document
Defect	***Name***$_{set}$, ***Description***$_{set}$, ***Severity***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	Quality Measurement
Usability	***Confidence***$_{set}$, ***Difficulty***$_{set}$		Test Tool (Test Robot)	

2) The heuristics used to make an assessment based on that data

The facts *SQM Document* and *Quality Measurement* are used in the JESS rule. *SQM Document* counts the number of documents related to software quality management. *Quality Measurement* checks the existence of attributes to measure quality of the project's software product. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *SQM Document* and *Quality measurement* is more than 1, which means we can assume that the quality is measureed.

```
(defrule KPA15-3Yes
  (SQM Document, ?z) (Quality Measurement, ?y)
   (test (and (< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA15-3 Yes))
  (printout Result.txt " KPA 15.3 Are measurements of quality compared to goals for software product
quality to determine if the quality goals are satisfied? Yes." crlf))

(defrule KPA15-3No
  (SQM Document, ?z) (Quality Measurement, ?y)
   (test (or (= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA15-3 No))
  (printout Result.txt " KPA 15.3 Are measurements of quality compared to goals for software product
quality to determine if the quality goals are satisfied? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be a project's software quality plan and the quality of the project's software products is measured, analyzed, and compared to the products' quantitative quality goals on an event-driven basis [28]. The heuristics checks the existence of documents related to software quality plan and attributes to measure quality of the project's software product.

4) The limitations of the data and heuristics

The documents related to Software Quality Management should be predefined and recongnized in the knowledge base.

KPA 15.4 Does the project follow a written organizational policy for managing software quality?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$***, Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	SQM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			SQM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,**			SQM Activity

	Description$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	SQM Document

2) The heuristics used to make an assessment based on that data

The facts *SQM Group, SQM Tool, SQM Activity,* and *SQM Document* are used in the JESS rule. *SQM Group* counts the number of responsible group or **Individual**s assigned to software quality management. *SQM Activity* counts the number of **Activit(y)**ies related to software quality management. *SQM Tool* counts the number of tools to use for software quality management. *SQM Document* counts the number of documents related to software quality management. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *SQM Group, SQM Tool, SQM Activity,* and *SQM Document* is more than 1, which means software quality management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA15-4Yes
  (SQM Group, ?w) (SQM Tool, ?x) (SQM Activity, ?y) (SQM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA15-4 Yes))
  (printout Result.txt " KPA 15.4 Does the project follow a written organizational policy for managing
software quality? Yes." crlf))

(defrule KPA15-4No
  (SQM Group, ?w) (SQM Tool, ?x) (SQM Activity, ?y) (SQM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA15-4 No))
  (printout Result.txt " KPA 15.4 Does the project follow a written organizational policy for managing
software quality? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Software Quality Management should be predefined and recongnized in the knowledge base.

KPA 15.5 Do members of the software engineering group and other software-related groups receive required training in software quality management (e.g., training in collecting measurement data and benefits of quantitatively managing product quality)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years) ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	Software Quality Management Education

2) The heuristics used to make an assessment based on that data

The fact *Software Quality Management Education* is used in the JESS rule. The *Software Quality Management Education* checks whether the **Individual**s, who manage software quality, have sufficient experiences. The fact-related attributes are gathered

automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Software Quality Management* is 1, which means the **Individual**s, who manage software quality, have sufficient experiences.

```
(defrule KPA15-5Yes
(Software Quality Management Education, ?x)
(test (= 1  ?x ))
 =>
(assert (KPA15-5 Yes))
(printout Result.txt "KPA 15.5 Do members of the software engineering group and other software-related groups receive required training in software quality management (e.g., training in collecting measurement data and benefits of quantitatively managing product quality)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the members of the software engineering group and other software-related groups receive required training in software quality management [28]. By checking the training experience of **Individual**s, who manages software quality, have sufficient experiences, we can understand whether the **Individual**s, who manages software quality, receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 15.6 Are measurements used to determine the status of the activities for managing software quality (e.g., the cost of poor quality)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$		Project Planning Tool (MS Project)	Quality Management, Bad Quality Management
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**Initial-Milestone** is an attribute of **Activity**.		
Final-	***PlannedEndDate***$_{set}$,	**FinalMilesto**		

Milstone	***ActualEndDate***$_{set}$	**ne** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Quality Management* and *Bad Quality Management* are used in the JESS rule. *Quality Management* counts the number of **Activit(y)**ies related to software quality management. *Bad Quality Management* counts the number of software quality management **Activit(y)**ies, which do not have proper attributes of cost. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{QualityManagement} - \textit{Bad}\text{QualityManagement})}{\text{QualityManagement}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA15-6AlmostAlways
 (Quality Management, ?x)
 (Bad Quality Management, ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
 (assert (KPA15-6 AlmostAlways))
```

(printout Result.txt "KPA 15.6 Are measurements used to determine the status of the activities for managing software quality (e.g., the cost of poor quality)? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to software quality management **Activit(y)**ies and the attributes of cost to determine cost of achieving the quality goals.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Quality Management should be predefined and recongnized in the knowledge base.

KPA 15.7 Are the activities performed for software quality management reviewed with senior management on a periodic basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	SQM Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		

Activity	**InitialMilestone, FinalMilestone,** ***Name*** set, ***Description*** set, **Complete** set (Yes=1, No=blank or 0), ***Type*** set (RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			SQM Review

2) The heuristics used to make an assessment based on that data

The fact *SQM Review,* and *SQM Reviewer* are used in the JESS rule. *SQM Review* counts the number of **Activit(y)**ies related to software quality management. *SQM Reviewer* counts the number of **Individual**s who are assigned to review software quality management **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software quality management review and **Individual**s to review software quality management **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA15-7Yes
```

```
 (SQM Review, ?x)
 (SQM Reviewer, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
 (assert (KPA15-7 Yes))
 (printout Result.txt " KPA 15.7 Are the activities performed for software quality management reviewed with senior management on a periodic basis? Yes." crlf))

(defrule KPA15-7No
 (SQM Review, ?x)
 (SQM Reviewer, ?y)
 (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
 (assert (KPA15-7 No))
 (printout Result.txt " KPA 15.7 Are the activities performed for software quality management reviewed with senior management on a periodic basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of software quality management review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review software quality management **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Software Quality Management should be predefined and recongnized in the knowledge base.

Q. KPA 16. Defect Prevention

The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring [28].

KPA 16.1 Are defect prevention activities planned?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s.	Configuration Management Plan (ClearCase)	DefectPrevention Document

2) The heuristics used to make an assessment based on that data

The fact *DefectPreventionDocument* is used in the JESS rule. *DefectPreventionDocument* counts the number of document such as Defect Prevention Plan. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *DefectPreventionDocument* is more than 1, which means the document related to defect prevention plan is exist.

```
(defrule KPA16-1Yes
  (DefectPreventionDocument, ?x)   (test (<= 1  ?x ))
   =>
  (assert (KPA16-1 Yes))
  (printout Result.txt " KPA 16.1 Are defect prevention activities planned? Yes." crlf))

(defrule KPA16-1No
 (DefectPreventionDocument, ?x)
 (test (= 0  ?x ))
   =>
  (assert (KPA16-1 No))
  (printout Result.txt " KPA 16.1 Are defect prevention activities planned? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The document such as Defect Prevention Plan states the policy for defect prevention **Activit(y)**ies. By checking the existence of Defect Prevention Plan related documents, we can assume that the project tries to follow a written organizational policy for defect prevention **Activit(y)**ies.

4) The limitations of the data and heuristics

The documents related to Defect Prevention should be predefined and recongnized in the knowledge base.

KPA 16.2 Does the project conduct causal analysis meetings to identify common causes of defects?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Dependency***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	Defect Meeting
Defect	***Name***$_{set}$, ***Description***$_{set}$,		Activity &	Defect Tracking

	Severity$_{set}$		Defect Tracking Tool (ClearQuest)	

2) The heuristics used to make an assessment based on that data

The facts *Defect Meeting,* and *Defect Tracking* are used in the JESS rule. *Defect Meeting* counts the number of meetings related to defect prevention. *Defect Tracking* checks the esistence of attributes related to defect tracking. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *Defect Meeting,* and *Defect Tracking* is more than 1, which means the project conducts causal analysis meetings to identify common causes of defects.

```
(defrule KPA16-2Yes
  (Defect Meeting, ?x) (Defect Tracking, ?y)
   (test (and  (< 0 ?x )(< 0 ?y ) ))
   =>
  (assert (KPA16-2 Yes))
  (printout Result.txt " KPA 16.2 Does the project conduct causal analysis meetings to identify common causes of defects? Yes." crlf))

(defrule KPA16-2No
```

```
(Defect Meeting, ?x) (Defect Tracking, ?y)
 (test (or (= 0 ?x )(= 0 ?y )))
 =>
(assert (KPA16-2 No))
(printout Result.txt " KPA 16.2 Does the project conduct causal analysis meetings to identify common causes of defects? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be a meeting and the defect prevention data are documented and tracked [28]. The heuristics checks the existence of attributes related to defect prevention **Activit(y)**ies such as meeting, and defect tracking.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Defect Prevention should be predefined and recongnized in the knowledge base.

KPA 16.3 Once identified, are common causes of defects prioritized and systematically eliminated?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,**		Project	Defect

	FinalMilestone, ***Name***$_{set}$**,** ***Dependency***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Planning Tool (MS Project)	Meeting
Version	**VersionIdentification**	**Version** contains **Subsystem**s.	Configuration Management Tool (ClearCase)	Defect Document
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other), ***Size***$_{set}$			
Individual	***Experience***$_{set}$(software development years), ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Aartfact**s.	MS Project	Defect Ownership

2) The heuristics used to make an assessment based on that data

The facts *Defect Meeting, Defect Document,* and *Defect Ownership* are used in the JESS rule. *Defect Meeting* counts the number of meetings related to defect prevention. *DP Document* counts the number of versioned documents related to defect prevention.

Defect Ownership checks the **Individual** ownership of a document. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *Defect Meeting, DP Document* and *Defect Ownership* is more than 1, which means it is possible that once the defect is identified the causes of defects is prioritized and systematically eliminated.

```
(defrule KPA16-3Yes
  (Defect Meeting, ?x) (DP Document, ?y) (Defect Ownership, ?z)
   (test (and (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA16-3 Yes))
  (printout Result.txt " KPA 16.3 Once identified, are common causes of defects prioritized and
systematically eliminated? Yes." crlf))

(defrule KPA16-3No
  (Defect Meeting, ?x) (DP Document, ?y) (Defect Ownership, ?z)
   (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA16-3 No))
  (printout Result.txt " KPA 16.3 Once identified, are common causes of defects prioritized and
systematically eliminated? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should be a defect prevention activities meeting, incorporate revision, and members of the software engineering group and software related groups receive feedback on the status [28]. The heuristics checks the existence of meeting, versioned document and **Individual** ownership of the document.

4) The limitations of the data and heuristics

The documents related to Defect Prevention should be predefined and recongnized in the knowledge base.

KPA 16.4 Does the project follow a written organizational policy for defect prevention activities?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	DP Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns		

		Activit(y)ies.		
Organization	***SoftwareTools***$_{set}$			DP Tool
Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			DP Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	DP Document

2) The heuristics used to make an assessment based on that data

The facts *DP Group, DP Tool, DP Activity,* and *DP Document* are used in the JESS rule. *DP Group* counts the number of responsible group or **Individual**s assigned to defect prevention **Activit(y)**ies. *DP Activity* counts the number of **Activit(y)**ies related to defect prevention. *DP Tool* counts the number of tools to use for defect prevention. *DP Document* counts the number of documents related to defect prevention. The fact-related attrubites are gathered automatically by querying attributes from The PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *DP Group, DP Tool, DP Activity,* and *DP Document* is more than 1, which means defect prevention related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA16-4Yes
  (DP Group, ?w) (DP Tool, ?x) (DP Activity, ?y) (DP Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA16-4 Yes))
  (printout Result.txt " KPA 16.4 Does the project follow a written organizational policy for
defect prevention activities? Yes." crlf))

(defrule KPA16-4No
   (DP Group, ?w) (DP Tool, ?x) (DP Activity, ?y) (DP Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA16-4 No))
```

(printout Result.txt " KPA 16.4 Does the project follow a written organizational policy for defect prevention activities? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Defect Prevention should be predefined and recongnized in the knowledge base.

KPA 16.5 Do members of the software engineering group and other software-related groups receive required training to perform their defect prevention activities (e.g., training in defect prevention methods and the conduct of task kick-off or causal analysis meetings)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Individual	***Title***$_{set}$, ***Experience***$_{set}$(software development years) ***FirstName***$_{set}$, ***LastName***$_{set}$		Project Planning Tool (MS Project)	Defect Prevention Education

2) The heuristics used to make an assessment based on that data

The fact *Defect Prevention Education* is used in the JESS rule. The *Defect Prevention Education* checks whether the **Individual**s, who perform defect prevention **Acitivt(y)**ies , have sufficient experiences. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Defect Prevention Education* is 1, which means the **Individual**s, who perform defect prevention **Acitivt(y)**ies, have sufficient experiences.

```
(defrule KPA16-5Yes
(Defect Prevention Education, ?x)
(test (= 1  ?x ))
=>
(assert (KPA16-5 Yes))
(printout Result.txt " KPA 16.5 Do members of the software engineering group and other software-related groups receive required training to perform their defect prevention activities (e.g., training in defect prevention methods and the conduct of task kick-off or causal analysis meetings)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the members of the software engineering group and other software-related groups receive required training to perform their defect prevention **Activit(y)**ies [28]. By checking the training experience of **Individual**s, who perform defect prevention **Acitivt(y)**ies, have sufficient experiences, we can understand whether the **Individual**s, who perform defect prevention **Acitivt(y)**ies, receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 16.6 Are measurements used to determine the status of defect prevention activities (e.g., the time and cost for identifying and correcting defects and the number of action items proposed, open, and completed)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA,		Project Planning Tool (MS Project)	Defection Prevention Activity, Bad Defect

	SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$			Prevention Activity
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**Initial-Milestone** is an attribute of **Activity**.		
Final-Milstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**Final-Milestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Defect Prevention Activity* and *Bad Defect Prevention Activity* are used in the JESS rule. *Defect Prevention Activity* counts the number of **Activit(y)**ies related to defect prevention. *Bad Quantitative Process Management* counts the number of quantitative **Activit(y)**ies, which do not have proper attributes of schedule and cost. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{Defect Prevention Activity} - \textit{Bad}\text{Defect Prevention Activity})}{\text{Defect Prevention Activity}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA16-6AlmostAlways
  (Defect Prevention Activity, ?x)
  (Bad Defect Prevention Activity, ?y)
  (test (<= 0.9 (/ (- ?x ?y) ?x)))
   =>
  (assert (KPA16-6 AlmostAlways))
  (printout Result.txt "KPA 16.6 Are measurements used to determine the status of defect prevention activities (e.g., the time and cost for identifying and correcting defects and the number of action items proposed, open, and completed)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to defect prevention **Activit(y)**ies and the attributes of schedule and the cost to determine the time and cost for identifying and correcting defects.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Defect Prevention should be predefined and recongnized in the knowledge base.

KPA 16.7 Are the activities and work products for defect prevention subjected to SQA review and audit?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	DP Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			DP Review
Artifact	***Name***$_{set}$, **Type**(Requirements File, DesignFile, DocumentFile, SourceFile,			DP Document

	Other)			

2) The heuristics used to make an assessment based on that data

The fact *DP Review, DP Reviewer,* and *DP Document* are used in the JESS rule. *DP Review* counts the number of **Activit(y)**ies related to defect prevention review. *DP Reviewer* counts the number of **Individual**s who are assigned to review defect prevention **Activit(y)**ies. *DP Document* counts the number of defect prevention related **Artifact**s. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies, **Artifact**s related to defect prevention, and **Individual**s to review defect prevention **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA16-7Yes
  (DP Review, ?x) (DP Reviewer, ?y) (DP Document, ?z)
  (test (and(<= 1 ?x) (<= 1 ?y) (<= 1 ?z) ))
   =>
  (assert (KPA16-7 Yes))
(printout Result.txt " KPA 16.7 Are the activities and work products for defect prevention subjected to SQA review and audit? Yes." crlf))

(defrule KPA16-7No
```

```
(DP Review, ?x) (DP Reviewer, ?y) (DP Document, ?z)
(test (or(= 0 ?x) (= 0 ?y) (= 0 ?z)  ))
 =>
(assert (KPA16-7 No))
(printout Result.txt " KPA 16.7 Are the activities and work products for defect prevention subjected to SQA review and audit? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of defect prevention **Activit(y)**ies, **Artifact**s and the assigned **Individual**s (SQA group) to defect prevention review **Activit(y)**ies, **Artifact**s.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Defect Prevention should be predefined and recongnized in the knowledge base.

R. KPA 17. Technology Change Management

The purpose of Technology Change Management is to identify new technologies (i.e., tools, methods, and process) and transition them into the organization in an orderly manner.

KPA 17.1 Does the organization follow a plan for managing technology changes?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$***, Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	TCM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			TCM Activity

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PCM Document

2) The heuristics used to make an assessment based on that data

The facts *TCM Group, TCM Activity,* and *TCM Document* are used in the JESS rule. *PCM Group* counts the number of responsible group or **Individual**s assigned to managing technology changes. *PCM Activity* counts the number of **Activit(y)**ies related to managing technology changes. *PCM Document* counts the number of documents related to managing technology changes. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PCM Group, PCM Activity,* and *PCM Document* is more than 1, which means technology changes management related group or **Individual**s, **Activit(y)**ies, and document are exist.

```
(defrule KPA17-1Yes
 (PCM Group, ?x) (PCM Activity, ?y) (PCM Document, ?z)
  (test (and (< 0 ?x ) (< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA17-1 Yes))
 (printout Result.txt " KPA 17.1 Does the organization follow a plan for managing technology changes?
```

```
Yes." crlf))

(defrule KPA17-1No
 (PCM Group, ?x) (PCM Activity, ?y) (PCM Document, ?z)
  (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA17-1 No))
 (printout Result.txt " KPA 17.1 Does the organization follow a plan for managing technology changes?
No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the organization develops and maintains a plan for technology change management [28]. The heuristics checks the existence of group, **Individual**s, **Activit(y)**ies, and documents for technology change management.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Technology Change Management should be predefined and recongnized in the knowledge base.

KPA 17.2 Are new technologies evaluated to determine their effect on quality and productivity?

1) The data available related to the CMM analysis

PAMPA 2.0	Source	Facts elicited

Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$, ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking), ***TrainingFacilities***$_{set}$, ***CourseReview***$_{set}$(**Individual**, software managers) ***Training***$_{set}$(audience, objectives, length, lesson plans)		Project Planning Tool (MS Project)	TCM Education
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			TCM Activity

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	TCM Document

2) The heuristics used to make an assessment based on that data

The facts *TCM Education, TCM Activity,* and *TCM Document* are used in the JESS rule. *TCM Education* checks the education experience of the responsible group or **Individual**s assigned to incorporating new technologies into the organization's standard software process. *TCM Activity* counts the number of **Activit(y)**ies related to incorporating new technologies into the organization's standard software process. *TCM Document* counts the number of documents related to incorporating new technologies into the organization's standard software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *TCM Education, TCM Activity,* and *TCM Document* is more than 1, which means new technologies are evaluated to determine their effect on quality and productivity.

```
(defrule KPA17-2Yes
 (TCM Education, ?x) (TCM Activity, ?y) (TCM Document, ?z)
  (test (and (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA17-2 Yes))
```

```
  (printout Result.txt " KPA 17.2 Are new technologies evaluated to determine their effect on quality and productivity?  Yes." crlf))

(defrule KPA17-2No
  (TCM Education, ?x) (TCM Activity, ?y) (TCM Document, ?z)
   (test (or  (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA17-2 No))
  (printout Result.txt " KPA 17.2 Are new technologies evaluated to determine their effect on quality and productivity?  No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the group responsible for the organization's technology change management activities works with the software projects in identifying areas of technology change. Second, the group responsible for the organization's technology change management systematically analyzes the organizatiojn's standard software process to identify areas that need or could benefit from new technology. Third, technologies are selected and acquired for the organization and software projects according to a documented procedure. Forth, pilot efforts for improving technology are conducted, where appropriate, before a new technology is introduced into normal practice [28]. The heuristics checks the existence of technology change management related **Activit(y)**ies, document and education.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Technology Change Management should be predefined and recongnized in the knowledge base.

KPA 17.3 Does the organization follow a documented procedure for incorporating new technologies into the organization's standard software process?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	TCM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			TCM Tool
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0),			TCM Activity

	ActualCost$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	TCM Document

2) The heuristics used to make an assessment based on that data

The facts *TCM Group, TCM Tool, TCM Activity,* and *TCM Document* are used in the JESS rule. *TCM Group* counts the number of responsible group or **Individual**s assigned to incorporating new technologies into the organization's standard software process. *TCM Activity* counts the number of **Activit(y)**ies related to incorporating new technologies into the organization's standard software process. *TCM Tool* counts the number of tools to use for incorporating new technologies into the organization's standard software process. *TCM Document* counts the number of documents related to incorporating new technologies into the organization's standard software process. The fact-related attributes are gathered automatically by querying attributes from the

PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *TCM Group, TCM Tool, TCM Activity,* and *TCM Document* is more than 1, which means technology change management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA17-3Yes
  (TCM Group, ?w) (TCM Tool, ?x) (TCM Activity, ?y) (TCM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA17-3 Yes))
  (printout Result.txt " KPA 17.3 Does the organization follow a documented procedure for incorporating
new technologies into the organization's standard software process? Yes." crlf))

(defrule KPA17-3No
  (TCM Group, ?w) (TCM Tool, ?x) (TCM Activity, ?y) (TCM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA17-3 No))
  (printout Result.txt " KPA 17.3 Does the organization follow a documented procedure for incorporating
new technologies into the organization's standard software process? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Technology Change Management should be predefined and recongnized in the knowledge base.

KPA 17.4 Does senior management sponsor the organization's activities for managing technology change (e.g., by establishing long-term plans and commitments for funding, staffing, and other resources)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$***, Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	TCM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		

Organization	***SoftwareTools***$_{set}$			TCM Tool
Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(<u>RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM</u>)			TCM Activity
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	TCM Document

2) The heuristics used to make an assessment based on that data

The facts *TCM Group, TCM Tool, TCM Activity,* and *TCM Document* are used in the JESS rule. *TCM Group* counts the number of responsible group or **Individual**s assigned to incorporating new technologies into the organization's standard software process. *TCM Activity* counts the number of **Activit(y)**ies related to incorporating new technologies into the organization's standard software process. *TCM Tool* counts the number of tools to use for incorporating new technologies into the organization's standard software process. *TCM Document* counts the number of documents related to incorporating new technologies into the organization's standard software process. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *TCM Group, TCM Tool, TCM Activity,* and *TCM Document* is more than 1, which means technology change management related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA17-4Yes
  (TCM Group, ?w) (TCM Tool, ?x) (TCM Activity, ?y) (TCM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA17-4 Yes))
  (printout Result.txt " KPA 17.4 Does senior management sponsor the organization's activities for
managing technology change (e.g., by establishing long-term plans and commitments for funding, staffing,
and other resources)? Yes." crlf))

(defrule KPA17-4No
  (TCM Group, ?w) (TCM Tool, ?x) (TCM Activity, ?y) (TCM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
```

=>

(assert (KPA17-4 No))

(printout Result.txt " KPA 17.4 Does senior management sponsor the organization's activities for managing technology change (e.g., by establishing long-term plans and commitments for funding, staffing, and other resources)? No." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Technology Change Management should be predefined and recongnized in the knowledge base.

KPA 17.5 Do process data exist to assist in the selection of new technology?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		

Activity	**InitialMilestone, FinalMilestone, *Name*$_{set}$, *Description*$_{set}$, Complete$_{set}$**(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)		Project Planning Tool (MS Project)	Schedule, Resource, Activity type
Organization	***Name***$_{set}$***, Description***$_{set}$ ***Funding***$_{set}$(Training Progran), ***SoftwareTools***$_{set}$, ***ToolCompatability***$_{set,}$ ***TrainingFacilities***$_{set,}$ ***CourseReview***$_{set}$ (**Individual**, software managers) ***Training***$_{set}$(audience, objectives, length, lesson plans), ***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan,			Tool

	DB, Activity Tracking)			
Artifact	***Name***$_{set}$,**Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)		Configuration Management Tool (ClearCase)	Configuration
Rework	**Adds, Changes, Deletes**	**Rework** contains attributes of an **Artifact**.		
Version	**VersionIdentification**	**Version** contains **Subsystem**s.		
Defect	***Name***$_{set}$, ***Description***$_{set}$		Activity & Defect Tracking Tool (ClearQuest)	Defect

2) The heuristics used to make an assessment based on that data

The facts *Schedule, Resource, Activity type, Tool, Configuration, Defect* are used in the JESS rule. These facts check the existence of attributes related to schedule, resource, **Activity** type, tools, configuration management, and defect. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Schedule, Resource, Activity type, Tool, Configuration, Defect* are all 1, which means the process data exist to assist in the selection of new technology.

```
(defrule KPA17-5Yes
(Schedule, ?x)
(Resource, ?Y)
(Activity type, ?Z)
(Tool, ?U)
(Configuration, ?V)
(Defect, ?W)
(test (and (= 1  ?x ) (= 1  ?Y ) (= 1  ?Z ) (= 1  ?U ) (= 1  ?V ) (= 1  ?W )))
=>
(assert (KPA17-5 Yes))
(printout Result.txt " KPA 17.5 Do process data exist to assist in the selection of new technology?
AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that there should appropriate data on the software processes and software work products to support analyses performed to evaluate and select technology changes [28]. By checking the existence of attributes related to software processes and software work products, we can assume that the process data exist to assist in the selection of new technology.

4) The limitations of the data and heuristics

None

KPA 17.6 Are measurements used to determine the status of the organization's activities for managing technology change (e.g., the effect of implementing technology changes)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$		Project Planning Tool (MS Project)	Technology Change Management, Bad Technology Change Management
Initial Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**Initial-Milestone** is an attribute of **Activity**.		
Final	***PlannedEndDate***$_{set}$,	**FinalMilestone**		

Milstone	***ActualEndDate***$_{set}$	is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Technology Change Management* and *Bad Technology Change Management* are used in the JESS rule. *Technology Change Management* counts the number of **Activit(y)**ies related to technology change management. *Bad Technology Change Management* counts the number of quantitative **Activit(y)**ies, which do not have proper attributes of cost and schedule. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If

$$\frac{(\text{Technology Change Management} - \mathit{Bad}\text{Technology Change Management})}{\text{Technology Change Management}}$$ is more

than 0.9, it means 90 % of the project is measured.

```
(defrule KPA17-6AlmostAlways
 (Technology Change Management, ?x)
 (Bad Technology Change Management, ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
 (assert (KPA17-6 AlmostAlways))
```

(printout Result.txt "KPA 17.6 Are measurements used to determine the status of the organization's activities for managing technology change (e.g., the effect of implementing technology changes)? AlmostAlways." crlf))

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to technology change management **Activit(y)**ies and the attributes of cost and schedule to determine the number, type of technology change **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Technology Change Management should be predefined and recongnized in the knowledge base.

KPA 17.7 Are the organization's activities for managing technology change reviewed with senior management on a periodic basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	TCM Reviewer

Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			TCM Review

2) The heuristics used to make an assessment based on that data

The facts *TCM Review,* and *TCM Reviewer* are used in the JESS rule. *TCM Review* counts the number of **Activit(y)**ies related to technology change management. *TCM Reviewer* counts the number of **Individual**s who are assigned to review technology change management **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to technology change management review and **Individual**s to review technology change management **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA17-7Yes
  (TCM Review, ?x) (TCM Reviewer, ?y)
  (test (and(<= 1 ?x) (<= 1 ?y)  ))
   =>
  (assert (KPA17-7 Yes))
  (printout Result.txt " KPA 17.7 Are the organization’s activities for managing technology change reviewed with senior management on a periodic basis? Yes." crlf))

(defrule KPA17-7No
  (TCM Review, ?x) (TCM Reviewer, ?y)
  (test (or(= 0 ?x) (= 0 ?y)  ))
   =>
  (assert (KPA17-7 No))
  (printout Result.txt " KPA 17.7 Are the organization’s activities for managing technology change reviewed with senior management on a periodic basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of technology change management review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review technology change management **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Technology Change Management should be predefined and recongnized in the knowledge base.

S. KPA 18. Process Change Management

The purpose of Process Change Management is to continually improve the software processes used in the organization with the intent of improving software quality, increasing productivity, and decreasing the cycle time for product development [28].

KPA 18.1 Does the organization follow a documented procedure for developing and maintaining plans for software process improvement?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***set, ***Description***set ,***Type***set(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	PCM Group
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***set			PCM Tool
Activity	**InitialMilestone,**			PCM Activity

	FinalMilestone, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PCM Document

2) The heuristics used to make an assessment based on that data

The facts *PCM Group, PCM Tool, PCM Activity,* and *PCM Document* are used in the JESS rule. *PCM Group* counts the number of responsible group or **Individual**s assigned to developing and maintaining plans for software process improvement. *PCM Activity* counts the number of **Activit(y)**ies related to developing and maintaining plans for software process improvement. *PCM Tool* counts the number of tools to use for developing and maintaining plans for software process improvement. *PCM Document* counts the number of documents related to developing and maintaining plans for software process improvement. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PCM Group, PCM Tool, PCM Activity,* and *PCM Document* is more than 1, which means developing and maintaining plans for software process improvement related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA18-1Yes
  (PCM Group, ?w) (PCM Tool, ?x) (PCM Activity, ?y) (PCM Document, ?z)
   (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA18-1 Yes))
  (printout Result.txt " KPA 18.1 Does the organization follow a documented procedure for developing and
maintaining plans for software process improvement? Yes." crlf))

(defrule KPA18-1No
  (PCM Group, ?w) (PCM Tool, ?x) (PCM Activity, ?y) (PCM Document, ?z)
   (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA18-1 No))
  (printout Result.txt " KPA 18.1 Does the organization follow a documented procedure for developing and
maintaining plans for software process improvement? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for developing and maintaining plans for software process improvement.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Process Change Management should be predefined and recongnized in the knowledge base.

KPA 18.2 Do people throughout your organization participate in software process improvement activities (e.g., on teams to develop software process improvements)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***set***, Description***set ,***Type***set(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	PCM Individual
Individual	***FirstName***set, ***LastName***set, ***Title***set	**Individual** owns **Activit(y)**ies.		
Activity	**InitialMilestone, FinalMilestone,**			PCM Activity

	Name$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PCM Document

2) The heuristics used to make an assessment based on that data

The fact *PCM Individual, PCM Activity,* and *PCM Document* are used in the JESS rule. *PCM Individual* counts the number of responsible **Individual**s assigned to software process improvements. *PCM Activity* counts the number of **Activit(y)**ies related to software process improvements. *PCM Document* counts the number of documents related to software process improvements. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PCM Individual, PCM Activity,* and *PCM Document* is more than 1, which means people throughout the organization participate in software process improvement activities.

```
(defrule KPA18-2Yes
  (PCM Individual, ?x) (PCM Activity, ?y) (PCM Document, ?z)
   (test (and (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA18-2 Yes))
  (printout Result.txt " KPA 18.2 Do people throughout your organization participate in software process improvement activities (e.g., on teams to develop software process improvements)?  Yes." crlf))

(defrule KPA18-2No
  (PCM Individual, ?x) (PCM Activity, ?y) (PCM Document, ?z)
   (test (or (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
   =>
  (assert (KPA18-2 No))
  (printout Result.txt " KPA 18.2 Do people throughout your organization participate in software process improvement activities (e.g., on teams to develop software process improvements)?  No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The heuristics checks the existence of software process improvement **Activit(y)**ies and the assigned group and Individuals to these **Activit(y)**ies.

4) The limitations of the data and heuristics

The **Activit(y)**ies related to Process Change Management should be predefined and recongnized in the knowledge base.

KPA 18.3 Are improvements continually made to the organization's standard software process and the projects' defined software processes?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone,** **FinalMilestone,** ***Name***$_{set}$**,** ***Description***$_{set}$**,** **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set,}$ ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			PCM Activity

Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PCM Document

2) The heuristics used to make an assessment based on that data

The facts *PCM Activity,* and *PCM Document* are used in the JESS rule. *PCM Activity* counts the number of **Activit(y)**ies, which are performed in accordance with the software process improvement plan. *PCM Document* counts the number of documents related to software process improvement proposal. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PCM Activity,* and *PCM Document* is more than 1, which means improvements continually made to the organization's standard software process and the projects' defined software processes.

```
(defrule KPA18-3Yes
  (PCM Activity, ?y) (PCM Document, ?z)
   (test (and (< 0 ?y ) (< 0 ?z )))
   =>
  (assert (KPA18-3 Yes))
  (printout Result.txt " KPA 18.3 Are improvements continually made to the organization's standard
software process and the projects' defined software processes? Yes." crlf))
```

```
(defrule KPA18-3No
 (PCM Activity, ?y) (PCM Document, ?z)
  (test (or (= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA18-3 No))
 (printout Result.txt " KPA 18.3 Are improvements continually made to the organization's standard
software process and the projects' defined software processes? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that first, the software process improvement activities are performed in accordance with the software process improvement plan. Second, there is a software process improvement proposal. Third, the software process improvements are installed on a pilot basis. Forth, the improvement is implemented according to a documentted procedure [28]. The heuristics checks the existence of software process improvement activities, which are performed in accordance with the software process improvement plan and the software process improvement proposal.

4) The limitations of the data and heuristics

We can not change the subjective activity, the software process improvements are installed on a pilot basis and the implementation according to a documented procedure.

KPA 18.4 Does the organization follow a written policy for implementing software process improvements?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Organization	***Name***$_{set}$, ***Description***$_{set}$,***Type***$_{set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)		Project Planning Tool (MS Project)	PCM Group
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.		
Organization	***SoftwareTools***$_{set}$			PCM Tool
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC,			PCM Activity

	PR, QPM, SQM, DP, TCM, PCM)			
Artifact	***Name***$_{set}$ **Type**(RequirementsFile, DesignFile, DocumentFile, SourceFile, Other)	**Artifact**s are related to **Feature**s	Configuration Management Tool (ClearCase)	PCM Document

2) The heuristics used to make an assessment based on that data

The fact *PCM Group, PCM Tool, PCM Activity,* and *PCM Document* are used in the JESS rule. *PCM Group* counts the number of responsible group or **Individual**s assigned to software process improvements. *PCM Activity* counts the number of **Activit(y)**ies related to software process improvements. *PCM Tool* counts the number of tools to use for software process improvements. *PCM Document* counts the number of documents related to software process improvements. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks each of *PCM Group, PCM Tool, PCM Activity,* and *PCM Document* is more than 1, which means software process improvements related group or **Individual**s, **Activit(y)**ies, Tools and document are exist.

```
(defrule KPA18-4Yes
```

```
 (PCM Group, ?w) (PCM Tool, ?x) (PCM Activity, ?y) (PCM Document, ?z)
  (test (and (< 0 ?w ) (< 0 ?x )(< 0 ?y ) (< 0 ?z )))
  =>
 (assert (KPA18-4 Yes))
 (printout Result.txt " KPA 18.4 Does the organization follow a written policy for implementing software
process improvements? Yes." crlf))

(defrule KPA18-4No
 (PCM Group, ?w) (PCM Tool, ?x) (PCM Activity, ?y) (PCM Document, ?z)
  (test (or (= 0 ?w ) (= 0 ?x )(= 0 ?y ) (= 0 ?z )))
  =>
 (assert (KPA18-4 No))
 (printout Result.txt " KPA 18.4 Does the organization follow a written policy for implementing software
process improvements? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the policy specifies individuals, document, **Activit(y)**ies, and support tools [28]. The heuristics above checks the existence of group, **Individual**s, **Activit(y)**ies, tools and documents for managing the system requirements allocated to software.

4) The limitations of the data and heuristics

The **Activit(y)**ies and documents related to Process Change Management should be predefined and recongnized in the knowledge base.

KPA 18.5 Is training in software process improvement required for both management and technical staff?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***Experience***$_{set}$(software development years) ***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$		Project Planning Tool (MS Project)	Process Change Management Education

2) The heuristics used to make an assessment based on that data

The fact *Process Change Management Education* is used in the JESS rule. The *Process Change Management Education* checks whether the **Individual**s, who participates in software process improvement, have sufficient experiences. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the *Process Change Management Education* is 1, which means the **Individual**s, who participate in software process improvement, have sufficient experiences.

```
(defrule KPA18-5Yes
(Process Change Management Education, ?x)
```

```
(test (= 1  ?x ))
 =>
(assert (KPA18-5 Yes))
(printout Result.txt " KPA 18.5 Is training in software process improvement required for both management and technical staff? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

The Capability Maturity Model states that the managers and technical staff on the software engineering group and other software-related groups receive required training in software process improvement [28]. By checking the training experience of **Individual**s, who participates in software process improvement, we can understand whether the **Individual**s, who participates in software process improvement, receive the required training to perform these **Activit(y)**ies.

4) The limitations of the data and heuristics

None

KPA 18.6 Are measurements made to determine the status of the activities for software process improvement (e.g., the effect of implementing each process improvement compared to its defined goals)?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Activity	**InitialMilestone**, **FinalMilestone**, ***Name***$_{set}$, ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM), ***Description***$_{set}$, **Complete**$_{set}$(Yes=1, No=blank or 0), **ActualCost**$_{set}$, **PlannedCost**$_{set}$		Project Planning Tool (MS Project)	Process Change Management, Bad Process Change Management
Initial-Milestone	***PlannedStartDate***$_{set}$, ***ActualStartDate***$_{set}$	**Initial-Milestone** is an attribute of **Activity**.		
Final-Milstone	***PlannedEndDate***$_{set}$, ***ActualEndDate***$_{set}$	**Final-Milestone** is an attribute of **Activity**.		

2) The heuristics used to make an assessment based on that data

The facts *Process Change Management* and *Bad Process Change Management* are used in the JESS rule. *Process Change Management* counts the number of **Activit(y)**ies related to process change management. *Bad Process Change Management* counts the number of quantitative **Activit(y)**ies, which do not have proper attributes of cost and schedule. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks the equation result below. If $\frac{(\text{Process Change Management} - \textit{Bad}\text{Process Change Management})}{\text{Process Change Management}}$ is more than 0.9, it means 90 % of the project is measured.

```
(defrule KPA18-6AlmostAlways
 (Process Change Management, ?x)
 ( Bad Process Change Management , ?y)
 (test (<= 0.9 (/ (- ?x ?y) ?x)))
  =>
 (assert (KPA18-6 AlmostAlways))
 (printout Result.txt "KPA 18.6 Are measurements made to determine the status of the activities for software process improvement (e.g., the effect of implementing each process improvement compared to its defined goals)? AlmostAlways." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of **Activit(y)**ies related to process change management **Activit(y)**ies and the attributes of cost and schedule to determine the number, type of process change **Activit(y)**ies.

4) The limitations of the data and heuristics

It is not easy to measure the size and the effect of implementing the process change objetively.

KPA 18.7 Are software process improvement efforts reviewed with senior management on a periodic basis?

1) The data available related to the CMM analysis

PAMPA 2.0			Source	Facts elicited
Object	Attributes	Relationships		
Individual	***FirstName***$_{set}$, ***LastName***$_{set}$, ***Title***$_{set}$	**Individual** owns **Activit(y)**ies.	Project Planning Tool (MS Project)	PCM Reviewer
Organization	***Name***$_{set}$, ***Description***$_{set}$	**Organization** contains **Individual**s		
Activity	**InitialMilestone, FinalMilestone,** ***Name***$_{set}$, ***Description***$_{set}$, **Complete**$_{set}$(Yes=1,			PCM Review

	No=blank or 0), ***Type***$_{set}$(RM, SPP, SPTO, SSM, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, SQM, DP, TCM, PCM)			

2) The heuristics used to make an assessment based on that data

The fact *PCM Review,* and *PCM Reviewer* are used in the JESS rule. *PCM Review* counts the number of **Activit(y)**ies related to software process improvement efforts. *PCM Reviewer* counts the number of **Individual**s who are assigned to review software process improvement **Activit(y)**ies. The fact-related attributes are gathered automatically by querying attributes from the PAMPA 2.0 Knowledge base instead of searching by human efforts for each attribute in the project environment.

Below is the JESS rule to measure the questionnaire. It checks whether there are any **Activit(y)**ies related to software process improvement efforts review and **Individual**s to review software process improvement **Activit(y)**ies. If this condition is satisfied, then we know it satisfies this questionnaire.

```
(defrule KPA18-7Yes
 (PCM Review, ?x) (PCM Reviewer, ?y)
 (test (and(<= 1 ?x) (<= 1 ?y)  ))
  =>
```

```
 (assert (KPA18-7 Yes))
 (printout Result.txt " KPA 18.7 Are software process improvement efforts reviewed with senior management on a periodic basis? Yes." crlf))

(defrule KPA18-7No
 (PCM Review, ?x) (PCM Reviewer, ?y)
 (test (or(= 0 ?x) (= 0 ?y)  ))
  =>
 (assert (KPA18-7 No))
 (printout Result.txt "KPA 18.7 Are software process improvement efforts reviewed with senior management on a periodic basis? No." crlf))
```

3) An argument as to why these heuristics match those of human evaluators or the intent of the CMM evaluation

It checks the existence of software process improvement efforts review **Activit(y)**ies and the assigned **Individual**s (senior manager) to review software process improvement **Activit(y)**ies.

4) The limitations of the data and heuristics

The review **Activit(y)**ies related to Process Change Management should be predefined and recongnized in the knowledge base.

APPENDIX B

PAMPA II OBJECT CLASSES

A software process can be divided into the following objects, each of which have relationships to other objects and attributes:

- **ProjectList**
  - **Project**
    - **ProjectVersion**
      - **Plan**
        - **Process**
          - **Activity**
            - **InitialMilestone**
            - **Risk**
            - **FinalMilestone**
              - **Criteria**
      - **Supplier**
        - **ReusableSourceFile**
        - **COTSRunFile**
      - **Organization**
        - **Individual**
          - **Salary**
          - **WorkBreakdownStructure**
      - **SoftwareProduct**
        - **Feature**
        - **Change**
        - **Problem**
        - **Defect**
        - **Version**
          - **Subsystem**
            - **Deliverable**
              - **Chunk**
                - **Volume**
                - **Structure**
              - **Rework**
          - **VAndVTest**
          - **UsabilityTest**
            - **Usability**
      - **Customer**

Activity
Artifact
Chunk
COTSRunFile
Criteria
Customer
Defect
Deliverable
Feature
FinalMilestone
Individual
InitialMilestone
Organization
Plan
Process
Project
ProjectVersion
ReusableSourceFile
Rework
Risk
Salary
SoftwareProduct
Structure
Subsystem
Supplier
Usability
UsabilityTest
VAndVTest
Version
Volume
WorkBreakdownStructure

Activity

Relationships

Activit(**y**)ies are contained in a **Process**.
Activit(**y**)ies are related to **Activit**(**y**)ies.
Activity is owned by an **Induvudual**.

Attributes

InitialMilestone
Risk
FinalMilestone

Strings

Name ***Set***
Description ***Set***
SecurityVector ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])
PointerToPredicessorActivities
PointerToSuccessorActivities
PointerToPredicessorProcesses
PointerToSuccessorProcesses
Complete ***Set***(Yes=1, No=blank or 0)
PointerToSignoffIndividual
ActualCost ***Set***
PlannedCost ***Set***
Type ***Set***(RM, SPP, SPPO, SQA, SCM, OPF, OPD, TP, ISM, SPE, IC, PR, QPM, DP, TCM, PCM)

Artifacts

$\boxed{\text{Relationships}}$

Artifact contains **Chunk**s.
Artifact has attribute **Rework**.
Artifact is authored by an **Individual**.
Artifact is authored by an **Organization**.
Artifact is owned by an **Individual**.
Artifacts are contained in a **Subsystem**.
Artifacts contain **Defect**s.
Artifacts are related to **Feature**s.

Chunk

Relationships

Chunks are contained in **Deliverable**.
Chunks contain **Chunk**s.

Attributes

Structure
Volume
Strings:
`Name`

COTSRunFile

Relationships

COTSRunFiles are provided by a **Supplier**.
COTSRunFiles are related to **Feature**s.
COTSRunFile is owned by **Individual**.

Attributes

Strings:

Name$_{\textbf{Set}}$
Description$_{\textbf{Set}}$
SecurityVector$_{\textbf{Set}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Criteria

Relationships

Criteria is an attribute of **InitialMilestone**
Criteria is an attribute of **Risk**
Criteria is an attribute of **FinalMilestone**

Attributes

<u>Strings:</u>

Knowledge$_{\textit{Set}}$(Describing when criteria 1 is met)
Knowledge$_{\textit{Set}}$(Describing when criteria 2 is met)

- 1
-
-

Knowledge$_{\textit{Set}}$(Describing when criteria **n** is met)

Customer

Relationships

Customers are related to **ProjectVersion**.

Attributes

<u>Strings:</u>

`Name` ***Set***
`Description` ***Set***

<u>Gather:</u>

`Performance`

<u>Set:</u>

`ExperienceLevel` ***Set***
`Satisfaction` ***Set***

Defect

$\boxed{\text{Relationships}}$

SoftwareProduct contains **Defect**s.
Defects are located in **Deliverable**s.
Defect is owned by an **Individual**.

$\boxed{\text{Attributes}}$

Strings:

Name$_{\textbf{\textit{Set}}}$
Description$_{\textbf{\textit{Set}}}$
Identification$_{\textbf{\textit{Set}}}$(number)

Status (These attributres should be Gnat columns):

Open$_{\textbf{\textit{Set}}}$(date)
AllocateResources$_{\textbf{\textit{Set}}}$(date)
V&VTest$_{\textbf{\textit{Set}}}$(date)
Close$_{\textbf{\textit{Set}}}$(date)

Rework

Strings:

Name ***Set***
Language ***Set***(Natural Language(English, Chinese, etc.), Program Language(Ada, C++, Fortran, Java, etc.))
Type(RequementsFile, DesignFile, DocumentFile, SourceFile, Other)
Complete ***Set***(Yes=1, No=blank or 0)
PointerToSignoffIndividual
Description ***Set***
SecurityVector ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Gather:

Chunks
BranchesTested
FanIn
FanOut
GlobalVariables
TotalBranches
UniqueNCSLOC
UniqueNCSS
UniqueReferenceNCSLOC(Unique compared to a unique library based NCSLOC list)
UniqueReferenceSLOC(Unique compared to a unique library based SLOC list)
UniqueSLOC
UniquesSS

Calculate:

InformationFlow1 ***Calculate***
InformationFlow2 ***Calculate***
InformationFlow3 **Calculate**

Accumulate:

BlankSS **Calculate**
BytesObjectCode **Calculate**
BytesSourceCode **Calculate**
CommentSLOC **Calculate**
CommentSS **Calculate**
CompilerDirectiveSS **Calculate**
DataDeclarationSS **Calculate**
Decisions **Calculate**
ExecutableSS **Calculate**
FunctionPoints **Calculate**
GlobalVariables **Calculate**
ObjectPoints **Calculate**
SLOC **Calculate**
SS **Calculate**
Variables **Calculate**
VolumeSoftSci ***Calculate***

Average per chunk:

EssentialComplexity ***Calculate***
InheritanceDepth ***Calculate***
Knots ***Calculate***

NestingDepth
SourceLiveVariables$_{\textit{\textbf{Calculate}}}$
SourceLiveVariablesPerExecutableSS$_{\textit{\textbf{Calculate}}}$
SpanLiveVariables$_{\textit{\textbf{Calculate}}}$
SpanLiveVariablesPerExecutableSS$_{\textit{\textbf{Calculate}}}$
Spans$_{\textit{\textbf{Calculate}}}$
TresholdLiveVariables$_{\textit{\textbf{Calculate}}}$
TresholdLiveVariablesPerExecutableSS$_{\textit{\textbf{Calculate}}}$
Variables$_{\textit{\textbf{Calculate}}}$

Feature

Relationships

Features are contained in a **SoftwareProduct**.
Feature is owned by an **Individual**.
Features are related to **Deliverable**s.
Features are related to **Subsystem**s.
Features are related to **VAndVTest**s.
Features are related to **UsabilityTest**s.
Features are related to **Version**s.
Features are related to **COTSRunFile**s.
Features are related to **ReusableSourceFile**s.

Attributes

Strings:

Name $_{\textbf{Set}}$
Description $_{\textbf{Set}}$
SecurityVector $_{\textbf{Set}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])
Status $_{\textbf{Set}}$(Proposed, Approved, Incorporated, Validated)

FinalMilestone

Relationships

FinalMilestone is an attribute of **Process**.
FinalMilestone is an attribute of **Activity**.

Attributes

Criteria

Strings:

PlannedEndDate$_{\textbf{Set}}$
ActualEndDate$_{\textbf{Set}}$
SecurityVector$_{\textbf{Set}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Individual

Relationships

Individual authors **Subsystem**s.
Individual authors **Deliverable**s.
Individual authors **VAndVTest**s.
Individual authors **UsabilityTest**s.
Individual performs **WorkBreakdownStructure**.
Individual is related to **Salar(y)**ies.
Individual manages an **Organization**.
Individual is a member of an **Organization**.
Individual owns **Activit(y)**ies.
Individual owns **COTSRunFile**s.
Individual owns **Defect**s.
Individual owns **Deliverable**s.
Individual owns **Feature**s.
Individual owns **Plan**s.
Individual owns **Process**es.
Individual owns **Project**.
Individual owns **ReusableSourceFile**s.
Individual owns **SoftwareProduct**.
Individual owns **Subsystem**s.
Individual owns **UsabilityTest**s.
Individual owns **VAndVTest**s.
Individual owns **Version**s.
Individual owns **WorkBreakdownStructure**s.
Individual runs **VAndVTest**s.
Individual runs **UsabilityTest**s.

Attributes

Strings:

- *EmployeeNumber*$_{\textbf{Set}}$
- *EmploymentDate*$_{\textbf{Set}}$
- *Experience*$_{\textbf{Set}}$(software development years)
- *FirstName*$_{\textbf{Set}}$
- *LastName*$_{\textbf{Set}}$
- *MiddleName*$_{\textbf{Set}}$
- *OverheadFactor*$_{\textbf{Set}}$(≥ 1)
- *Title*$_{\textbf{Set}}$
- *TestResult*$_{\textbf{Set}}$

Calculate:

- *DefectRate*$_{\textbf{Calculate}}$(Defects per 1,000 SLOC).
- *EffortToDate*$_{\textbf{Calculate}}$(in person months).
- *Productivity*$_{\textbf{Calculate}}$(average source lines of code per month).
- *Time*$_{\textbf{Calculate}}$(time spent on project in person months).
- *TurmoilRate*$_{\textbf{Calculate}}$(Turmoil per 1,000 SLOC).

InitialMilestone

Relationships

InitialMilestone is an attribute of **Process**.
InitialMilestone is an attribute of **Activity**.

Attributes

Criteria

Strings:

PlannedStartDate ***Set***
ActualStartDate ***Set***
SecurityVector ***Set*** (Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Organization

Relationships

Organizations contain **Organization**s .
Organization perform **WorkBreakdownStructure**.
Organization contains **Individual**s.
Organizations are managed by an **Individual**.
Organizations are parts of **ProjectVersion**.

Attributes

Strings:

- *Name*$_{Set}$
- *Description*$_{Set}$
- *SecurityVector*$_{Set}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])
- *Funding*$_{Set}$(Training Progran)
- *SoftwareTools*$_{Set}$(Word Processing system, CM, Test, Requirement, Plan, DB, Activity Tracking)
- *ComputerResource*$_{Set}$
- *ToolCompatability*$_{Set}$
- *TrainingFacilities*$_{Set}$
- *CourseReview*$_{Set}$(Individual, software managers)
- *Training*$_{Set}$(Audience, Objectives, Length, Lesson, Plans)

Calculate:

- *AverageIndividualProductivity*$_{Calculate}$(average NCSS person per month).
- *DefectRate*$_{Calculate}$(Defects per 1,000 SLOC).
- *Efficiency*$_{Calculate}$
- *EffortToDate*$_{Calculate}$(in person months).
- *Productivity*$_{Calculate}$(average NCSS per month).
- *Speedup*$_{Calculate}$

Gather:

- n
- Salary$_{average}$

Plan

Relationships

Plan is part of a **ProjectVersion**.
Plan is associated with **Plan**s**.**
Plan is owned by an **Individual**.
Plan contains **Process**es**.**

Attributes

Strings:

Name$_{\textbf{Set}}$
Description$_{\textbf{Set}}$
SecurityVector$_{\textbf{Set}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Process

Relationships

Processes are contained in a **Plan**.
Processes are contained in a **WorkBreakdownStructure**.
Process is associated with **Process**es.
Process contains **Process**es.
Process contains **Activit**(**y**)ies.
Process is owned by an **Individual.**

Attributes

InitialMilestone
Risk
FinalMilestone

Strings:

Name$_{Set}$
Description$_{Set}$
SecurityVector$_{Set}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Project

Relationships

Projects are contained in a **ProjectList**.
Project contains **ProjectVersion**s.
Project is owned by an **Individual**.

Attributes

Strings:

Name ***Set***
Description ***Set***
SecurityVector ***Set*** (Administor, Manager, Individual[Create, Modify, Delete, Read, Write])
Overhead ***Set***

Calculate:

Cost ***Calculate***
EffortToDate ***Calculate*** (person months).
HeadCount ***Calculate*** (persons)
FullTimeEquivalent ***Calculate*** (persons)
TimeToDate ***Calculate*** (months).

ProjectList

Relationships

A **ProjectList** contains **Project**s.

Attributes

Strings:

Name$_{\textbf{\textit{Set}}}$

Description$_{\textbf{\textit{Set}}}$

SecurityVector$_{\textbf{\textit{Set}}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

ProjectVersion

Relationships

ProjectVersions are contained in a **Project**.
ProjectVersion contains **Plan**s.
ProjectVersion contains **Supplier**s.
ProjectVersion contains **Organization**s.
ProjectVersion contains **SoftwareProduct**s.
ProjectVersion contains **Customer**s.

Attributes

Strings:

Name$_{\textbf{Set}}$
Description$_{\textbf{Set}}$
SecurityVector$_{\textbf{Set}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Gather:

Cost
Time

ReusableSourceFile

Relationships

ReusableSourceFiles are provided by a **Supplier**.
ReusableSourceFile is owned by a **Supplier**.
ReusableSourceFiles are related to **Feature**s.

Attributes

Strings:

Name$_{\textbf{\textit{Set}}}$
Description$_{\textbf{\textit{Set}}}$
SecurityVector$_{\textbf{\textit{Set}}}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Rework

Relationships

Rework contains attributes of a **Deliverable.**

Attributes

Gather:

Adds

Changes

Deletes

Calculate:

Turmoil $_{\textbf{\textit{Calculate}}}$

Risk

Relationships

Risk is an attribute of **Process**.
Risk is an attribute of **Activity**.

Attributes

Criteria

Strings:

EstimatedRisk ***Set***
Description ***Set***
SecurityVector ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Salary

Relationships

Salar(**y**)ies are related to an **Individual**.

Attributes

Strings:

Amount ***Set***
EffectiveDate ***Set***

SoftwareProduct

Relationships

A **SoftwareProduct** is contained in a **ProjectVersion**.
A **SoftwareProduct** is owned by an **Individual**.
A **SoftwareProduct** contains **Feature**s.
A **SoftwareProduct** contains **Defect**s.
A **SoftwareProduct** contains **Version**s.

Attributes

Strings:

Name ***Set***
Description ***Set***
SecurityVector ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Gather:

`Reliability`
`Usability`
`Volume`

Structure

Relationships

Structure contains attributes of a **Chunk**.

Attributes

Gather:

```
EssentialComplexity
InheritanceDepth
Knots
NestingDepth
Pairs
RelativePercentageUsagePairs
SpanLiveVariables
Spans
TresholdLiveVariables
Variables
```

Strings:

n1 ***Set*** (Threshold for threshold live variables)

Calculate

CyclomaticNumber ***Calculate***
SourceLiveVariables ***Calculate***
SourceLiveVariablesPerExecutableSS ***Calculate***
SpanLiveVariablesPerExecutableSS ***Calculate***
TresholdLiveVariablesPerExecutableSS ***Calculate***

Subsystem

Relationships

Subsystems are contained in a **Version**.
Subsystems contains **Subsystem**s.
Subsystem contains **Deliverable**s.
Subsystem is owned by an **Individual**.
Subsystem is authored by an **Individual**.
Subsystem is authored by an **Organization**.
Subsystems are related to **Feature**s.

Attributes

Strings:

Name ***Set***
Type(RequementsFile, DesignFile, DocumentFile, SourceFile, Other)
Complete ***Set***(Yes=1, No=blank or 0)
PointerToSignoffIndividual
Description ***Set***
SecurityVector ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Gather:

UniqueNCSLOC
UniqueNCSS
UniqueReferenceNCSLOC(Unique compared to a unique library based NCSLOC list)
UniqueReferenceSLOC(Unique compared to a unique library based SLOC list)
UniqueSLOC
UniqueSS

Accumulation:

Adds **Calculate**
BlankSS **Calculate**
BranchesTested **Calculate**
BytesObjectCode **Calculate**
BytesSourceCode **Calculate**
Changes **Calculate**
Chunks **Calculate**
CommentSLOC **Calculate**
CommentSS **Calculate**
CompilerDirectiveSS **Calculate**
DataDeclarationSS **Calculate**
Decisions **Calculate**
Deletes **Calculate**
ExecutableSS **Calculate**
FunctionPoints **Calculate**
GlobalVariables **Calculate**
ObjectPoints **Calculate**
SLOC **Calculate**
SS **Calculate**
TotalBranches **Calculate**
Turmoil ***Calculate***
Variables **Calculate**

VolumeSoftSci ***Calculate***

Supplier

Relationships

Suppliers are contained in a **ProjectVersion**.
Supplier provides **ReusableSourceFile**s.
Supplier provides **COTSRunFile**s.

Attributes

<u>Strings:</u>

Name$_{\textbf{Set}}$
Description$_{\textbf{Set}}$

Usability

Relationships

Usability contains attributes of a **UsabilityTest**.

Attributes

<u>Gather:</u>

`Responses`
`HelpRequests`
`Efficiency`

<u>Set:</u>

`Confidence`$_{\textit{Set}}$
`Difficulty`$_{\textit{Set}}$

<u>Forecast:</u>

<u>Infer:</u>

`SolutionCorrectness`$_{\textit{Infer}}$
`Solution`$_{\textit{Infer}}$(YES/NO)

UsabilityTest

Relationships

UsabilityTests are contained in a **Version**.
UsabilityTest are authored by an **Individual**.
UsabilityTest are authored by an **Organization**.
UsabilityTest is owned by an **Individual**.
UsabilityTest is run by an **Individual**.
UsabilityTests are related to **Feature**s.

Attributes

Usability

Strings:

`Name`
`Date`
`Description`
`Configuration`
`InputFiles`
`TestStartDateTime`
`TestEndDateTime`
`EngineerTime`
`TechnicianTime`
`Status`(Failed/Passed, i. e. Usability Test is ready or not ready to be run.)
`Duration`

VAndVTest

Relationships

VAndVTest is contained in a **Version**
VAndVTest is authored by an **Individual**.
VAndVTest is authored by an **Organization**.
VAndVTest is owned by an **Individual**.
VAndVTest is run by an **Individual**.
VAndVTests are related to **Feature**s.

Attributes

Strings:

`Configuration`
`Name`
`Description`
`Status`(Failed/Passed, i. e. Test is ready or not ready to be run.)
`Failure`(YES/NO)

Gather:

`Date`
`InputFiles`
`TestStartDateTime`
`TestEndDateTime`
`EngineerTime`
`TechnicianTime`
`Duration`
`CoverageVector`(% by source)

Version

Relationships

Versions are contained in a **SoftwareProduct**.
Version contains **Subsystem**s.
Version contains **VAndVTest**s.
Version contains **UsabilityTest**s.
Version is owned by an **Individual**.
Version is related to **Feature**s.

Attributes

Strings:

`PreviousVersionIdentification`
`SourceDir`
`VersionIdentification`
`Complete` ***Set***(Yes=1, No=blank or 0)
`PointerToSignoffIndividual`
`Description` ***Set***
`SecurityVector` ***Set***(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

Gather:

`Defects`
`VersionCreated`(date)
`UniqueNCSLOC`
`UniqueNCSS`
`UniqueReferenceNCSLOC`(Unique compared to a unique library based NCSLOC list)
`UniqueReferenceSLOC`(Unique compared to a unique library based SLOC list)
`UniqueSLOC`
`UniqueSS`

Accumulation:

`Adds` **Calculate**
`BlankSS` **Calculate**
`BranchesTested` **Calculate**
`BytesObjectCode` **Calculate**
`BytesSourceCode` **Calculate**
`Changes` **Calculate**
`Chunks` **Calculate**
`CommentSLOC` **Calculate**
`CommentSS` **Calculate**
`CompilerDirectiveSS` **Calculate**
`DataDeclarationSS` **Calculate**
`Decisions` **Calculate**
`Deletes` **Calculate**
`ExecutableSS` **Calculate**
`FunctionPoints` **Calculate**
`GlobalVariables` **Calculate**
`ObjectPoints` **Calculate**
`SLOC` **Calculate**
`SS` **Calculate**
`TotalBranches` **Calculate**
`Turmoil` ***Calculate***

Variables **Calculate**
VolumeSoftSci ***Calculate***

Volume

Relationships

Volume contains attributes of a **Chunk**.

Attributes

<u>Gather:</u>

`BlankSS`
`BytesSourceCode`
`BytesObjectCode`
`CommentSLOC`
`CommentSS`
`CompilerDirectiveSS`
`DataDeclarationSS`
`Decisions`
`ExecutableSS`
`FunctionPoints`
`PointorToUniqueSSReference`
`PointerToUniqueSLOCReference`
`ObjectPoints`
`Operands`
`Operators`
`SLOC`
`UniqueNCSLOC`
`UniqueOperands`
`UniqueOperators`
`UniqueReferenceNCSLOC`(Unique compared to a unique library based NCSLOC list)
`UniqueReferenceSLOC`(Unique compared to a unique library based SLOC list)
`UniqueSLOC`

<u>Volume Reuse Constants:</u>

CodeModifiedCode$_{\textbf{Set}}$(% of total effort to code modified code)
DesignModifiedCode$_{\textbf{Set}}$(% of total effort to design modified code)
IntegrationModifiedCode$_{\textbf{Set}}$(% of total effort to integration modified code)
k1$_{\textbf{Set}}$(Bailey and Basili used a value of 0.2)
k2$_{\textbf{Set}}$(Thebaut used a value of 0.857)
ModifiedCode$_{\textbf{Set}}$(% of code modified)

<u>Calculate:</u>

CommentBytesOC$_{\textbf{Calculate}}$
Length$_{\textbf{Calculate}}$
NCSS$_{\textbf{Calculate}}$
NCSLOC$_{\textbf{Calculate}}$
SS$_{\textbf{Calculate}}$
Vocabulary$_{\textbf{Calculate}}$
VolumeSoftSci$_{\textbf{Calculate}}$

WorkBreakdownStructure

Relationships

WorkBreakdownStructure is associated with an **Organization**.
WorkBreakdownStructure is associated with an **Individual.**
WorkBreakdownStructure is associated with **WorkBreakdownStructure**s**.**
WorkBreakdownStructure is owned by an **Individual**.
WorkBreakdownStructure contains **Process**es**.**

Attributes

Strings:

Name$_{Set}$
Description$_{Set}$
SecurityVector$_{Set}$(Administor, Manager, Individual[Create, Modify, Delete, Read, Write])

APPENDIX C

MATURITY QUESTIONNAIRE

KPA 1. Requirements Management

1 Are system requirements allocated to software used to establish a baseline for software engineering and management use?

2 As the systems requirements allocated to software change, are the necessary adjustments to software plans, work products, and activities made?

3 Does the project follow a written organizational policy for managing the system requirements allocated to software?

4 Are the people in the project who are charged with managing the allocated requirements trained in the procedures for managing allocated requirements?

5 Are measurements used to determine the status of the activities performed for managing the allocated requirements (e.g., total number of requirements changes that are proposed, open, approved, and incorporated into the baseline)?

6 Are the activities for managing allocated requirements on the project subjected to SQA review?

KPA 2. Software Project Planning

1 Are estimates (e.g., size, cost, and schedule) documented for use in planning and tracking the software project?

2 Do the software plans document the activities to be performed and the commitments made for the software project?

3 Do all affected groups and individuals agree to their commitments related to the software project?

4 Does the project follow a written organizational policy for planning a software project?

5 Are adequate resources provided for planning the software project (e.g., funding and experienced individuals)?

6 Are measurements used to determine the status of the activities for planning the software project (e.g., completion of milestones for the project planning activities as compared to the plan)?

7 Does the project manager review the activities for planning the software project on both a periodic and event-driven basis?

KPA 3. Software Project Tracking and Oversight

1 Are the project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans?

2 Is corrective action taken when actual results deviate significantly from the project's software plans?

3 Are changes in the software commitments agreed to by all affected groups and individuals?

4 Does the project follow a written organizational policy for both tracking and controlling its software development activities?

5 Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)?

6 Are measurements used to determine the status of the activities for software tracking and oversight (e.g., total effort expended in performing tracking and oversight activities)?

7 Are the activities for software project tracking and oversight reviewed with senior management on a periodic basis (e.g.,project performance, open issues, risks, and action items)?

KPA 4. Software Subcontract Management

1 Is a documented procedure used for selecting subcontractors based on their ability to perform the work?

2 Are changes to subcontracts made with the agreement of both the prime contractor and the subcontractor?

3 Are periodic technical interchanges held with subcontractors?

4 Are the results and performance of the software subcontractor tracked against their commitments?

5 Does the project follow a written organizational policy for managing software subcontracts?

6 Are the people responsible for managing software subcontracts trained in managing software subcontracts?

7 Are measurements used to determine the status of the activities for managing software subcontracts (e.g., schedule status with respect to planned delivery dates and effort expended for managing the subcontract)?

8 Are the software subcontract activities reviewed with the project manager on both a periodic and event-driven basis?

KPA 5. Software Quality Assurance (SQA)

1 Are SQA activities planned?

2 Does SQA provide objective verification that software products and activities adhere to applicable standards, procedures, and requirements?

3 Are the results of SQA reviews and audits provided to affected groups and individuals (e.g., those who performed the work and those who are responsible for the work)?

4 Are issues of noncompliance that are not resolved within the software project addressed by senior management (e.g., deviations from applicable standards)?

5 Does the project follow a written organizational policy for implementing SQA?

6 Are adequate resources provided for performing SQA activities (e.g., funding and a designated manager who will receive and act on software noncompliance items)?

7 Are measurements used to determine the cost and schedule status of the activities performed for SQA (e.g., work completed, effort and funds expended compared to the plan)?

8 Are activities for SQA reviewed with senior management on a periodic basis?

KPA 6. Software Configuration Management (SCM)

1 Are software configuration management activities planned for the project? Comments:

2 Has the project identified, controlled, and made available the software work products through the use of configuration management?

3 Does the project follow a documented procedure to control changes to configuration items/units?

4 Are standard reports on software baselines (e.g., software configuration control board minutes and change request summary and status reports) distributed to affected groups and individuals?

5 Does the project follow a written organizational policy for implementing software configuration management activities?

6 Are project personnel trained to perform the software configuration management activities for which they are responsible?

7 Are measurements used to determine the status of activities for software configuration management (e.g., effort and funds expended for software configuration management activities)?

8 Are periodic audits performed to verify that software baselines conform to the documentation that defines them (e.g., by the SCM group)?

KPA 7. Organization Process Focus

1 Are the activities for developing and improving the organization's and project's software processes coordinated across the organization (e.g., via a software engineering process group)?

2 Is your organization's software process assessed periodically?

3 Does your organization follow a documented plan for developing and improving its software process?

4 Does senior management sponsor the organization's activities for software process development and improvements (e.g., by establishing long-term plans, and by committing resources and funding)?

5 Do one or more individuals have full-time or part-time responsibility for the organization's software process activities (e.g., a software engineering process group)?

6 Are measurements used to determine the status of the activities performed to develop and improve the organization's software process (e.g., effort expended for software process assessment and improvement)?

7 Are the activities performed for developing and improving software processes reviewed periodically with senior management?

KPA 8. Organization Process Definition

1 Has your organization developed, and does it maintain, a standard software process?

2 Does the organization collect, review, and make available information related to the use of the organization's standard software process (e.g., estimates and actual data on software size, effort, and cost; productivity data; and quality measurements)?

3 Does the organization follow a written policy for both developing and maintaining its standard software process and related process assets (e.g., descriptions of approved software life cycles)?

4 Do individuals who develop and maintain the organization's standard software process receive the required training to perform these activities?

5 Are measurements used to determine the status of the activities performed to define and maintain the organization's standard software process (e.g., status of schedule milestones and the cost of process definition activities)?

6 Are the activities and work products for developing and maintaining the organization's standard software process subjected to SQA review and audit?

KPA 9. Training Program

1 Are training activities planned?

2 Is training provided for developing the skills and knowledge needed to perform software managerial and technical roles?

3 Do members of the software engineering group and other software-related groups receive the training necessary to perform their roles?

4 Does your organization follow a written organizational policy to meet its training needs?

5 Are adequate resources provided to implement the organization’s training program (e.g., funding, software tools, appropriate training facilities)?

6 Are measurements used to determine the quality of the training program?

7 Are training program activities reviewed with senior management on a periodic basis?

KPA 10. Integrated Software Management

1 Was the project's defined software process developed by tailoring the organization's standard software process?

2 Is the project planned and managed in accordance with the project’s defined software process?

3 Does the project follow a written organizational policy requiring that the software project be planned and managed using the organization’s standard software process?

4 Is training required for individuals tasked to tailor the organization’s standard software process to define a software process for a new project?

5 Are measurements used to determine the effectiveness of the integrated software management activities (e.g., frequency, causes and magnitude of replanning efforts)?

6 Are the activities and work products used to manage the software project subjected to SQA review and audit?

KPA 11. Software Product Engineering

1 Are the software work products produced according to the project’s defined software process?

2 Is consistency maintained across software work products (e.g., is the documentation tracing allocated requirements through software requirements, design, code, and test cases maintained)?

3 Does the project follow a written organizational policy for performing the software engineering activities (e.g., a policy which requires the use of appropriate methods and tools for building and maintaining software products)?

4 Are adequate resources provided for performing the software engineering tasks (e.g., funding, skilled individuals, and appropriate tools)?

5 Are measurements used to determine the functionality and quality of the software products (e.g., numbers, types, and severity of defects identified)?

6 Are the activities and work products for engineering software subjected to SQA reviews and audits (e.g., is required testing performed, are allocated requirements traced through the software requirements, design, code and test cases)?

KPA 12. Intergroup Coordination

1 On the project, do the software engineering group and other engineering groups collaborate with the customer to establish the system requirements?

2 Do the engineering groups agree to the commitments as represented in the overall project plan?

3 Do the engineering groups identify, track, and resolve intergroup issues (e.g., incompatible schedules, technical risks, or system-level problems)?

4 Is there a written organizational policy that guides the establishment of interdisciplinary engineering teams?

5 Do the support tools used by different engineering groups enable effective communication and coordination (e.g., compatible word processing systems, database systems, and problem tracking systems)?

6 Are measures used to determine the status of the intergroup coordination activities (e.g., effort expended by the software engineering group to support other groups)?

7 Are the activities for intergroup coordination reviewed with the project manager on both a periodic and event-driven basis?

KPA 13. Peer Reviews

1 Are peer reviews planned?

2 Are actions associated with defects that are identified during peer reviews tracked until they are resolved?

3 Does the project follow a written organizational policy for performing peer reviews?

4 Do participants of peer reviews receive the training required to perform their roles?

5 Are measurements used to determine the status of peer review activities (e.g., number of peer reviews performed, effort expended on peer reviews, and number of work products reviewed compared to the plan)?

6 Are peer review activities and work products subjected to SQA review and audit (e.g., planned reviews are conducted and follow-up actions are tracked)?

KPA 14. Quantitative Process Management

1 Does the project follow a documented plan for conducting quantitative process management?

2 Is the performance of the project's defined software process controlled quantitatively (e.g., through the use of quantitative analytic methods)?

3 Is the process capability of the organization's standard software process known in quantitative terms?

4 Does the project follow a written organizational policy for measuring and controlling the performance of the project's defined software process (e.g., projects plan for how to identify, analyze, and control special causes of variations)?

5 Are adequate resources provided for quantitative process management activities (e.g., funding, software support tools, and organizational measurement program)?

6 Are measurements used to determine the status of the quantitative process management activities (e.g., cost of quantitative process management activities and accomplishment of milestones for quantitative process management activities)?

7 Are the activities for quantitative process management reviewed with the project manager on both a periodic and event-driven basis?

KPA 15. Software Quality Management

1 Are the activities for managing software quality planned for the project?

2 Does the project use measurable and prioritized goals for managing the quality of its software products (e.g., functionality, reliability, maintainability and usability)?

3 Are measurements of quality compared to goals for software product quality to determine if the quality goals are satisfied?

4 Does the project follow a written organizational policy for managing software quality?

5 Do members of the software engineering group and other software-related groups receive required training in software quality management (e.g., training in collecting measurement data and benefits of quantitatively managing product quality)?

6 Are measurements used to determine the status of the activities for managing software quality (e.g., the cost of poor quality)?

7 Are the activities performed for software quality management reviewed with senior management on a periodic basis?

KPA 16. Defect Prevention

1 Are defect prevention activities planned?

2 Does the project conduct causal analysis meetings to identify common causes of defects?

3 Once identified, are common causes of defects prioritized and systematically eliminated?

4 Does the project follow a written organizational policy for defect prevention activities?

5 Do members of the software engineering group and other software-related groups receive required training to perform their defect prevention activities (e.g., training in defect prevention methods and the conduct of task kick-off or causal analysis meetings)?

6 Are measurements used to determine the status of defect prevention activities (e.g., the time and cost for identifying and correcting defects and the number of action items proposed, open, and completed)?

7 Are the activities and work products for defect prevention subjected to SQA review and audit?

KPA 17. Technology Change Management

1 Does the organization follow a plan for managing technology changes?

2 Are new technologies evaluated to determine their effect on quality and productivity?

3 Does the organization follow a documented procedure for incorporating new technologies into the organization's standard software process?

4 Does senior management sponsor the organization's activities for managing technology change (e.g., by establishing long-term plans and commitments for funding, staffing, and other resources)?

5 Do process data exist to assist in the selection of new technology?

6 Are measurements used to determine the status of the organization's activities for managing technology change (e.g.,the effect of implementing technology changes)?

7 Are the organization's activities for managing technology change reviewed with senior management on a periodic basis?

KPA 18. Process Change Management

1 Does the organization follow a documented procedure for developing and maintaining plans for software process improvement?

2 Do people throughout your organization participate in software process improvement activities (e.g., on teams to develop software process improvements)?

3 Are improvements continually made to the organization's standard software process and the projects' defined software processes?

4 Does the organization follow a written policy for implementing software process improvements?

5 Is training in software process improvement required for both management and technical staff?

6 Are measurements made to determine the status of the activities for software process improvement (e.g., the effect of implementing each process improvement compared to its defined goals)?

7 Are software process improvement efforts reviewed with senior management on a periodic basis?

VITA

Jinhwan Jung was born in Seoul, Korea in 1966. He received his Bachelor of Art (B.A.) in International Relationships at Korea Naval Academy in 1989. He received his Bachelor of Art (B.A.) in English language and literature at Yonsei University, Seoul, Korea in 1992. He received his Master of Science (M.S.) in computer science at Yonsei University, Seoul, Korea in 1997 and Ph.D. in computer science at Texas A&M University in 2004.

His research interests and areas of expertise are in project management using the Knowledge Based System. For further inquires about his work, he can be reached at the following address:

Seoul Kangnamgu Apgujungdong Hyundai APT 119-1304, Republic of Korea