UNIVERSIDAD POLITÉCNICA
DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INFORMÁTICOS

MÁSTER UNIVERSITARIO EN INGENIERÍA DEL SOFTWARE –
EUROPEAN MASTER ON SOFTWARE ENGINEERING

An Analysis Of The Recommended Knowledge For The Software Project Management Discipline

Master Thesis

Jonathan Araujo Duarte

Madrid, June 2014

This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfillment of the requirements for the degree of Master of Science on Software Engineering.

Master Thesis
Master Universitario en Ingeniería del Software – European Master on Software Engineering
Thesis Title: An Analysis Of The Recommended Knowledge For The Software Project Management Discipline
Thesis no: EMSE-2014-03
June 2014

Author: Jonathan Araujo Duarte
Systems and Computing Engineering
Universidad del Norte, Colombia

Supervisor:

Ana Moreno
Ph.D. Computer Science
Universidad Politécnica de Madrid

School of Computing
Universidad Politécnica de Madrid

Co-supervisor:

Lawrence Peters
Ph.D. Engineering Management
California Coast University

Software Project Manager and Consultant
Software Consulting International Ltd

ETSI Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo, s/n
28660 Boadilla del Monte (Madrid)
Spain

Table of Contents

Index of Tables

Index of Figures

1 Introduction

1.1 Objective of the Master Thesis

According to the PMBOK (Project Management Body of Knowledge), project management is “the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements” [1]. Project Management has proven to be one of the most important disciplines at the moment of determining the success of any project [2][3][4]. Given that many of the activities covered by this discipline can be said that are “horizontal” for any kind of domain, the importance of acknowledge the concepts and practices becomes even more obvious. The specific case of the projects that fall in the domain of Software Engineering are not the exception about the great influence of Project Management for their success.

The critical role that this discipline plays in the industry has come to numbers. A report by McKinsey & Co [4] shows that the establishment of programs for the teaching of critical skills of project management can improve the performance of the project in time and costs. As an example of the above, the reports exposes: “One defense organization used these programs to train several waves of project managers and leaders who together administered a portfolio of more than 1,000 capital projects ranging in Project management size from $100,000 to $500 million. Managers who successfully completed the training were able to cut costs on most projects by between 20 and 35 percent. Over time, the organization expects savings of about 15 percent of its entire baseline spending”.

In a white paper by the PMI (Project Management Institute) about the value of project management [5], it is stated that: “Leading organizations across sectors and geographic borders have been steadily embracing project management as a way to control spending and improve project results”. According to the research made by the PMI for the paper, after the economical crisis “Executives discovered that adhering to project management methods and strategies reduced risks, cut costs and improved success rates—all vital to surviving the economic crisis”. In every elite company, a proper execution of the project management discipline has become a must.

Several members of the software industry have putted effort into achieving ways of assuring high quality results from projects; many standards, best practices, methodologies and other resources have been produced by experts from different fields of expertise. In the industry and the academic community, there is a continuous research on how to teach better software engineering together with project management [4][6].

For the general practices of Project Management the PMI produced a guide of the required knowledge that any project manager should have in their toolbox to lead any kind of project, this guide is called the PMBOK. On the side of best practices and required knowledge for the Software Engineering discipline, the IEEE (Institute of Electrical and Electronics Engineers) developed the SWEBOK (Software Engineering Body of Knowledge) in collaboration with software industry experts and academic researchers, introducing into the guide many of the needed knowledge for a 5-year expertise software engineer [7]. The SWEBOK also covers management from the perspective of a software project.

This thesis is developed to provide guidance to practitioners and members of the academic community about project management applied to software engineering.

The way used in this thesis to get useful information for practitioners is to take an industry-approved guide for software engineering professionals such as the SWEBOK, and compare the content to what is found in the PMBOK. After comparing the contents of the SWEBOK and the PMBOK, what is found missing in the SWEBOK is used to give recommendations on how to enrich project management skills for a software engineering professional.

Recommendations for members of the academic community on the other hand, are given taking into account the GSwE2009 (Graduated Software Engineering 2009) standard [8]. GSwE2009 is often used as a main reference for software engineering master programs [9]. The standard is mostly based on the content of the SWEBOK, plus some contents that are considered to reinforce the education of software engineering. Given the similarities between the SWEBOK and the GSwE2009, the results of comparing SWEBOK and PMBOK are also considered valid to enrich what the GSwE2009 proposes. So in the end the recommendations for practitioners end up being also useful for the academic community and their strategies to teach project management in the context of software engineering.

1.2 Structure of the Thesis

The strategy used for the thesis was to do an overall recognition of what is contained in the primary sources of information (SWEBOK, PMBOK and GSwE2009), which is why a summary with the most relevant information of each source was made and presented as an independent section of the thesis. After the summaries of the sources, an information junction section is shown with the results of the comparison between contents. The comparison is made throw a table that makes visible the junction of information between the SWEBOK and the PMBOK, followed by an extended explanation of everything contained in the table. Finally, the recommendations for practitioners and for the academic community are made based on the results of the information junction section.

The following are brief descriptions of the sections:

SWEBOK Description: A summary of the relevant information provided in the SWEBOK for the teaching of Software Engineering together with Project Management. Answers questions like; what is the purpose of the guide, who is expected to be the reader and the description of the topics in the Project Management KA (Knowledge Area) of the SWEBOK are presented.

PMBOK Description: In this section is presented the summary of all what is contained inside of the PMBOK guide. The structure and intended audience are explained, and an overview for each one of the Knowledge Areas and the topics that compose them is provided.

GSwE2009 Description: Outcomes, expected background and the CBOK (Core Body of Knowledge) that the standard suggests, are covered in this summary.

SPM Knowledge Mapping: Here is presented an analysis of what is missing in the SWEBOK taking as reference the PMBOK. A table with colors indicating level of coverage by the SWEBOK of the PMBOK topics is shown and explained.

Recommendations: Based on the results from the "**SPM Knowledge Mapping**", recommendations are made for practitioners and academic community of the SPM in the context of software engineering. The recommendations are given based on the topics that were found missing in the SWEBOK and in the GSwE2009, taking into account that the GSwE2009 is mostly based in the SWEBOK.

Conclusion: In this section is exposed the conclusion about the suggestions and results from the research carried out for the thesis.

References: The references of all the resources that were used to carry out the research.

2 SWEBOK Description

2.1 SWEBOK Purpose and Intended Use

The SWEBOK is all about defining the bounds and required knowledge that is *"necessary but not sufficient"* to apply Software Engineering discipline. This characterization of the required knowledge areas is exposed based on a consensus among industry and academic experts, which by experience and proven studies have built the presented list of concepts, processes, tools and methods and their corresponding description. As a consequence of the previously said, the guide looks to promote a consistent view of the Software Engineering discipline and try to expose the boundary and relations with other disciplines such as computer science, project management, computer engineering, and mathematics. A professional software engineer could also get guidance on building its professional curriculum by approaching the defined KAs.

Because this is not intended to be an exhaustive list of concepts or areas that a Software Engineer should know about, the reader should approach it having in mind that the concepts here have been chosen because they are horizontally useful for building a software artifact, meaning that can be used as a toolkit that an informed expert should have for potential use.

The guide is organized in KAs and each one of them have subsections in which processes, tools and methods are explained in not so great detail, but enough to have an overall idea. Support material like links to literature about the respective topics are also given, so the reader could get more detailed information if needed. In this way the guide purpose a simple well-defined structure that anyone can use to track a topic of interest, review them and find resources for further information.

2.2 SWEBOK Intended Reader

The recommended practices in the SWEBOK and the precise manner in which is defined, makes it eligible for certain uses by certain readers. People in charge of doing processes like producing criteria for accreditation of academic programs, development of education and training programs, certification of specialists, or professional licensing. Generally, a professional society or related body maintains custody of such a formal definition. In cases where no such formality exists, the body of knowledge and recommended practices are "generally recognized" by practitioners and may be codified in a variety of ways for different uses.

It is hoped that readers will find this book useful in guiding them toward the knowledge and resources they need in their lifelong career development as software engineering professionals.

2.3 SWEBOK Structure of the Guide

The SWEBOK guide presents 11 chapters, 10 of them related to the KAs and the last one about the related disciplines Software Engineering has tight relation with. The first five KAs are presented in the same order as the waterfall lifecycle.

The breakdown of topics constitutes the core of each KA description, describing the decomposition of the KA into subareas, topics, and sub-topics. For each topic or sub-topic, a short description is given, along with one or more references.

The KAs are the following (*Figure 1 and Figure 2)*:

2.3.1 Software Requirements

A requirement is defined as a property that must be exhibited in order to solve some real-world problem.

The subareas that are treated in this KA are:

- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations

2.3.2 Software Design

According to the IEEE definition [IEEE 610.12-90], design is both "the process of defining the architecture, components, interfaces, and other characteristics of a system or component" and "the result of [that] process." This KA is divided into the following subareas:

- Software Design Fundamentals
- Key Issues in Software Design
- Software Structure and Architecture
- Design Quality Analysis
- Software Design Notations
- Software Design Strategies and Methods

2.3.3 Software Construction

This one refers to the combination of coding, verification, unit testing, integration testing, and debugging. The KA includes three subareas.

- Software Construction Fundamentals
- Managing Construction
- Practical Considerations

2.3.4 Software Testing

Testing is the dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite executions domain, against the expected behavior. It includes five subareas.

- Software Testing Fundamentals
- Test Levels
- Test Techniques
- Test-Related Measures
- Test Process

2.3.5 Software Maintenance

The maintenance phase of the life cycle starts upon delivery, but maintenance activities occur much earlier. The subareas are:

- Software Maintenance Fundamentals
- Key Issues in Software Maintenance
- Maintenance Process
- Techniques for Maintenance constitute

2.3.6 Software Configuration Management

Software Configuration Management (SCM) is the discipline of identifying the configuration of software at distinct points in time for the purpose of systematically controlling changes to the configuration and of maintaining the integrity and traceability of the configuration throughout the system life cycle. This KA includes six subareas.

- Management of the SCM Process
- Software Configuration Identification
- Software Configuration Control
- Software Configuration Status Accounting
- Software Configuration Auditing
- Software Release Management and Delivery

2.3.7 Software Engineering Management

This KA addresses the management and measurement of software engineering.

- Initiation and Scope Definition
- Software Project Planning
- Software Project Enactment
- Review and Evaluation
- Closure
- Measurement

2.3.8 Software Engineering Process

Here the reader can find the definition, implementation, assessment, measurement, management, change, and improvement of the software engineering process itself. The subareas are:

- Process Implementation and Change
- Process Definition
- Process Assessment
- Process and Product Measurements

2.3.9 Software Engineering Tools and Methods

This KA includes subareas for tools and methods for Software Engineering.

- Software Engineering Tools, this has nine subareas, one for each one of the other KAs, plus one for miscellaneous tools.
- Software Engineering Methods, it also has heuristic methods, formal methods, prototyping methods.

2.3.10 Software Quality

Software Quality deals with the considerations for quality in the software lyfe cycle process. Even though quality is taken into account in other KAs, given its importance it also has a dedicated section. The subareas are:

- Software Quality Fundamentals
- Software Quality Management Processes
- Practical Considerations

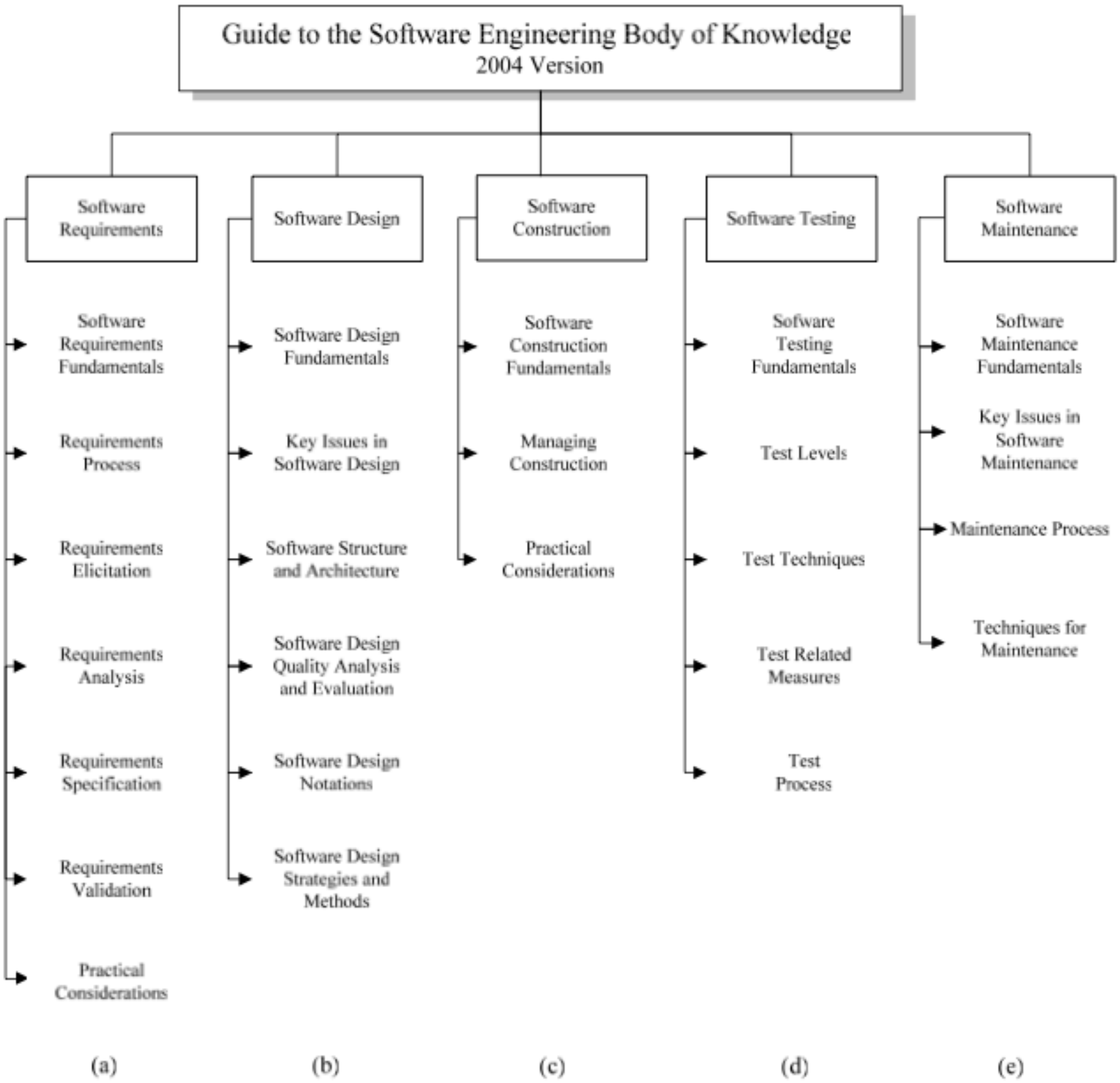

Guide to the Software Engineering Body of Knowledge
2004 Version
Software Requirements
Software Requirements Fundamentals
Requirements Process
Requirements Elicitation
Requirements Analysis
Requirements Specification
Requirements Validation
Practical Considerations
Software Design
Software Design Fundamentals
Key Issues in Software Design
Software Structure and Architecture
Software Design Quality Analysis and Evaluation
Software Design Notations
Software Design Strategies and Methods
Software Construction
Software Construction Fundamentals
Managing Construction
Practical Considerations
Software Testing
Sofware Testing Fundamentals
Test Levels
Test Techniques
Test Related Measures
Test Process
Software Maintenance
Software Maintenance Fundamentals
Key Issues in Software Maintenance
Maintenance Process
Techniques for Maintenance
(a)
(b)
(c)
(d)
(e)

Figure 1. First five SWEBOK KAs

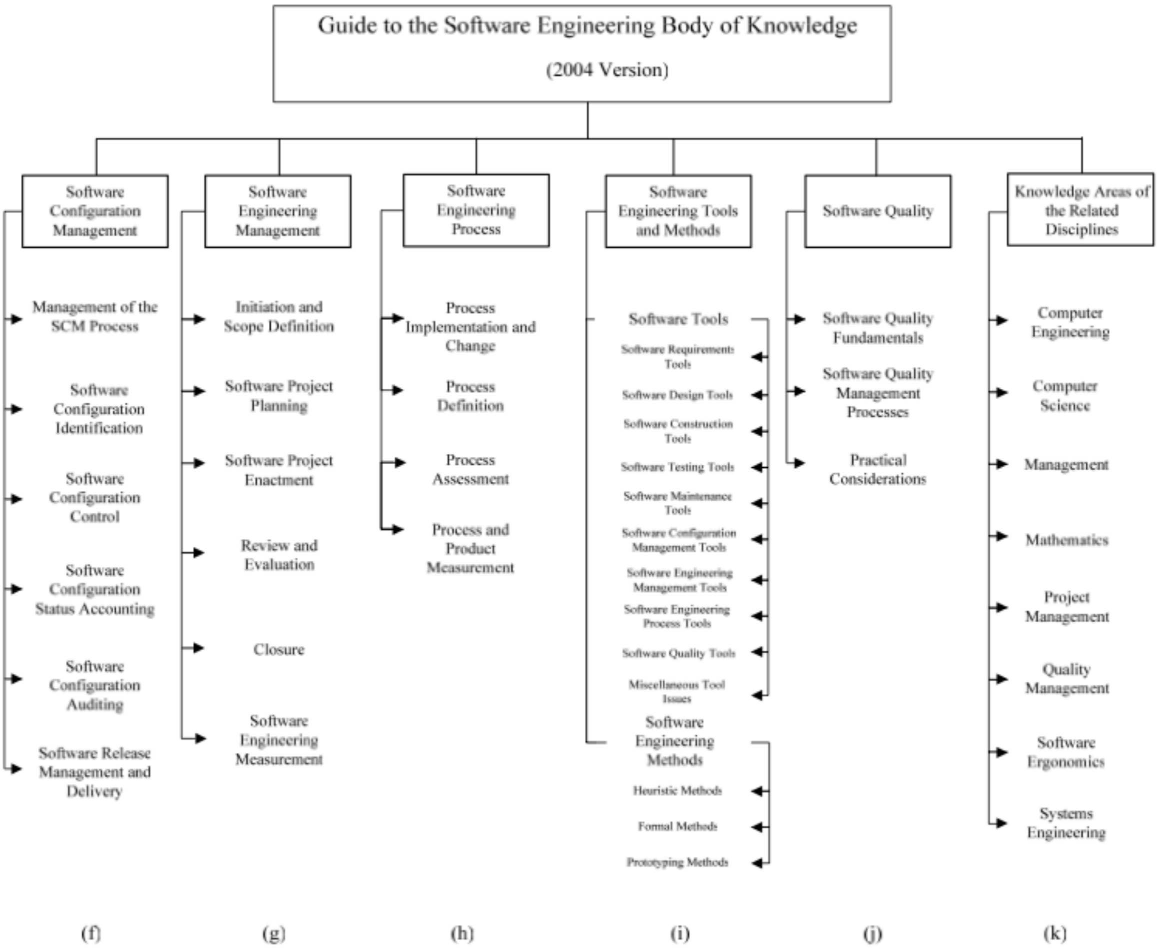

Guide to the Software Engineering Body of Knowledge
(2004 Version)
Software Configuration Management
Management of the SCM Process
Software Configuration Identification
Software Configuration Control
Software Configuration Status Accounting
Software Configuration Auditing
Software Release Management and Delivery
Software Engineering Management
Initiation and Scope Definition
Software Project Planning
Software Project Enactment
Review and Evaluation
Closure
Software Engineering Measurement
Software Engineering Process
Process Implementation and Change
Process Definition
Process Assessment
Process and Product Measurement
Software Engineering Tools and Methods
Software Tools
Software Requirements Tools
Software Design Tools
Software Construction Tools
Software Testing Tools
Software Maintenance Tools
Software Configuration Management Tools
Software Engineering Management Tools
Software Engineering Process Tools
Software Quality Tools
Miscellaneous Tool Issues
Software Engineering Methods
Heuristic Methods
Formal Methods
Prototyping Methods
Software Quality
Software Quality Fundamentals
Software Quality Management Processes
Practical Considerations
Knowledge Areas of the Related Disciplines
Computer Engineering
Computer Science
Management
Mathematics
Project Management
Quality Management
Software Ergonomics
Systems Engineering
(f)
(g)
(h)
(i)
(j)
(k)

Figure 2. Last six SWEBOK KAs

On the other hand the related disciplines are:

- Computer engineering
- Computer science
- Management
- Mathematics
- Project management
- Quality management
- Software ergonomics
- Systems engineering

2.4 Project Management KA

Software Engineering Management can be defined as the application of management activities—planning, coordinating, measuring, monitoring, controlling, and reporting—to ensure that the development and maintenance of software is systematic, disciplined, and quantified (IEEE610.12-90). The overall structure of the content of the Project Management KA can be seen in *Figure 3*.

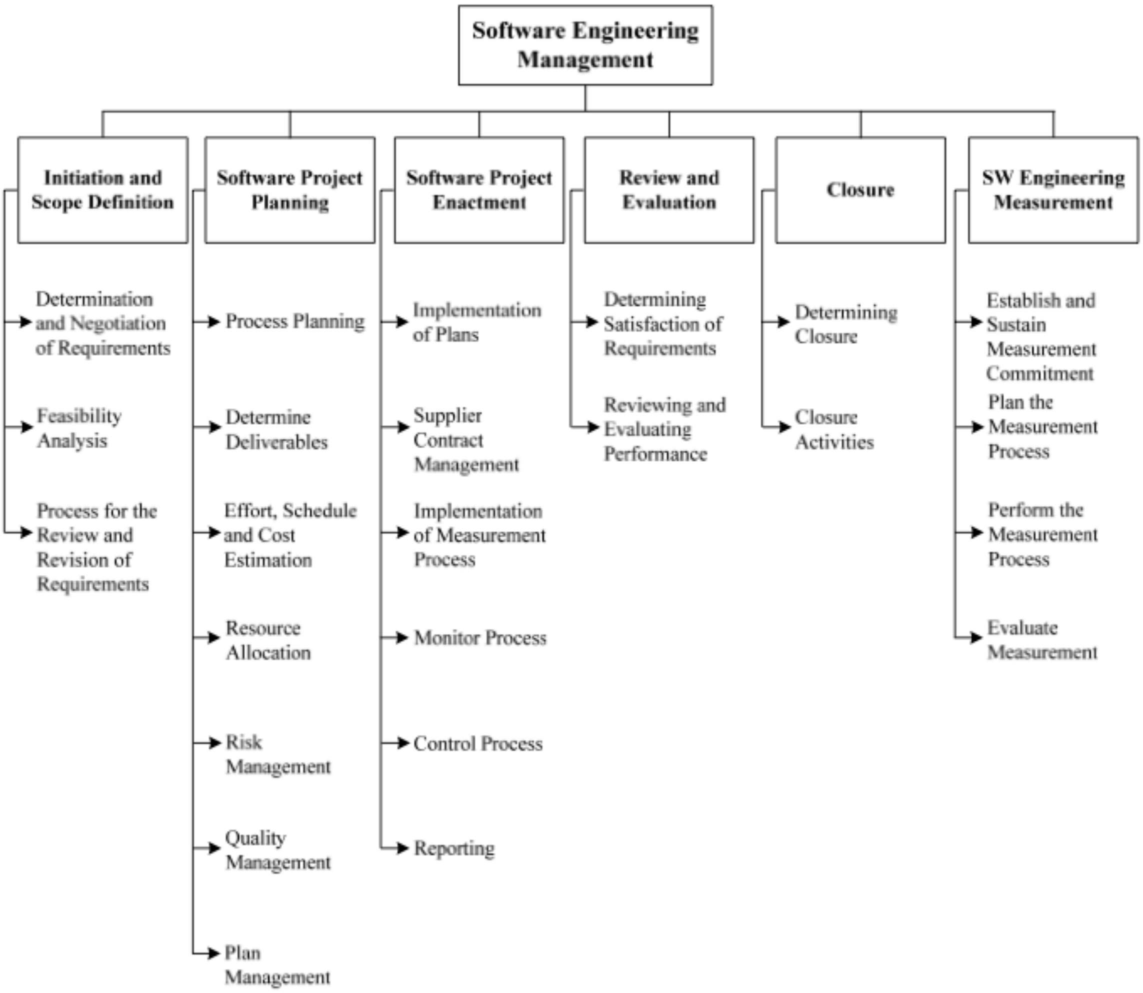

Software Engineering Management
Initiation and Scope Definition
Determination and Negotiation of Requirements
Feasibility Analysis
Process for the Review and Revision of Requirements
Software Project Planning
Process Planning
Determine Deliverables
Effort, Schedule and Cost Estimation
Resource Allocation
Risk Management
Quality Management
Plan Management
Software Project Enactment
Implementation of Plans
Supplier Contract Management
Implementation of Measurement Process
Monitor Process
Control Process
Reporting
Review and Evaluation
Determining Satisfaction of Requirements
Reviewing and Evaluating Performance
Closure
Determining Closure
Closure Activities
SW Engineering Measurement
Establish and Sustain Measurement Commitment
Plan the Measurement Process
Perform the Measurement Process
Evaluate Measurement

Figure 3. Breakdown of topics for Software Engineering Management

2.4.1 Initiation and scope definition

The activities that are done for initiation and scope definition have as goal to determine the software requirements and assess the feasibility of the project itself. Once feasibility is established comes the validation of requirements and the establishment of change procedures.

The activities are:

2.4.1.1 Determination and Negotiation of Requirements

This refers to the activities that serve for the objective of gathering, defining and validating the requirements of the software system. The different phases and activities are:

Requirements Elicitation. This phase is concerned with where software requirements come from and how the software engineer can collect them. It is the first stage in building an understanding of the problem the software is required to solve. It is fundamentally a human activity, and is where the stakeholders are identified and relationships established between the development team and the customer. It is variously termed “requirements capture,” “requirements discovery,” and “requirements acquisition”. Before development begins, requirements specialists may form the conduit for this communication. They must mediate between the domain of the software users (and other stakeholders) and the technical world of the software engineer.

Useful concepts for handling elicitation are:

- **Requirements sources:** Goals, domain knowledge, stakeholders, the operational environment, and the organizational environment.
- **Elicitation techniques:** Interviews, scenarios, prototypes, facilitated meetings, and observation.

Requirements Analysis. This topic is concerned with the process of analyzing requirements to:

- Detect and resolve conflicts between requirements
- Discover the bounds of the software and how it must interact with its environment
- Elaborate system requirements to derive software requirements

The traditional view of requirements analysis has been that it be reduced to conceptual modeling using one of a number of analysis methods such as the Structured Analysis and Design Technique (SADT). While conceptual modeling is important, we include the classification of requirements to help inform trade-offs between requirements (requirements classification) and the process of establishing these trade-offs (requirements negotiation).

Care must be taken to describe requirements precisely enough to enable the requirements to be validated, their implementation to be verified, and their costs to be estimated.

Concepts related to this are:

- **Requirements classification:** Functional or nonfunctional, derived from high-level requirements or an emergent property or is being imposed directly on the software by a stakeholder or some other source, requirement is on the product or the process, priority, scope and volatility/stability.
- **Conceptual modeling:** Data and control flows, state models, event traces, user interactions, object models, data models, and many others. This should be chosen based on several factors, nature of the problem, expertise of the software engineer and customer’s preferences.

Architectural Design and Requirements Allocation. At some point, the architecture of the solution must be derived. Architectural design is the point at which the requirements process overlaps with software or systems design and illustrates how impossible it is to cleanly decouple the two tasks.

Requirements Negotiation. Another term commonly used for this sub-topic is "conflict resolution." This concerns resolving problems with requirements where conflicts occur between two stakeholders requiring mutually incompatible features, between requirements and resources, or between functional and non-functional requirements, for example.

Software Specification. In software engineering jargon, "software requirements specification" typically refers to the production of a document, or its electronic equivalent, which can be systematically reviewed, evaluated, and approved. For complex systems, particularly those involving substantial non-software components, as many as three different types of documents are produced: system definition, system requirements, and software requirements. For simple software products, only the third of these is required.

Requirements Validation. The requirements documents may be subject to validation and verification procedures. The requirements may be validated to ensure that the software engineer has understood the requirements, and it is also important to verify that a requirements document conforms to company standards, and that it is understandable, consistent, and complete. Formal notations offer the important advantage of permitting the last two properties to be proven (in a restricted sense, at least). Different stakeholders, including representatives of the customer and developer, should review the document(s). Requirements documents are subject to the same software configuration management practices as the other deliverables of the software life cycle processes.

It is normal to explicitly schedule one or more points in the requirements process where the requirements are validated.

Concepts for these activities:

- **Requirements Review**
- **Prototyping**
- **Model Validation**
- **Acceptance Tests**

2.4.1.2 Feasibility Analysis (Technical, Financial, Social/Political)

Software engineers must be assured that adequate capability and resources are available in the form of people, expertise, facilities, infrastructure, and support (either internally or externally) to ensure that the project can be successfully completed in a timely and cost-effective manner (using, for example, a requirement-capability matrix). This often requires some "ballpark" estimation of effort and cost based on appropriate methods.

2.4.1.3 Process for the Review and Revision of Requirements

Given the inevitability of change, it is vital that agreement among stakeholders is reached at this early point as to the means by which scope and requirements are to be reviewed and revised (for example, via agreed change management procedures).

Software configuration control is concerned with managing changes during the software life cycle. It covers the process for determining what changes to make, the authority for approving certain changes, support for the implementation of those changes, and the concept of formal deviations from project requirements, as well as waivers of them.

- **Requesting, Evaluating, and Approving Software Changes.** The first step in managing changes to controlled items is determining what changes to make. The software change request process (see *Figure 3*) provides formal procedures for submitting and recording change requests, evaluating the potential cost and impact of a proposed change, and accepting, modifying, or rejecting the proposed change. Requests for changes to software configuration items may be originated by anyone at any point in the software life cycle and may include a suggested solution and requested priority. One source of change requests is the initiation of corrective action in response to problem reports. Regardless of the source, the type of change (for example, defect or enhancement) is usually recorded on the SCR (Software Change Request).

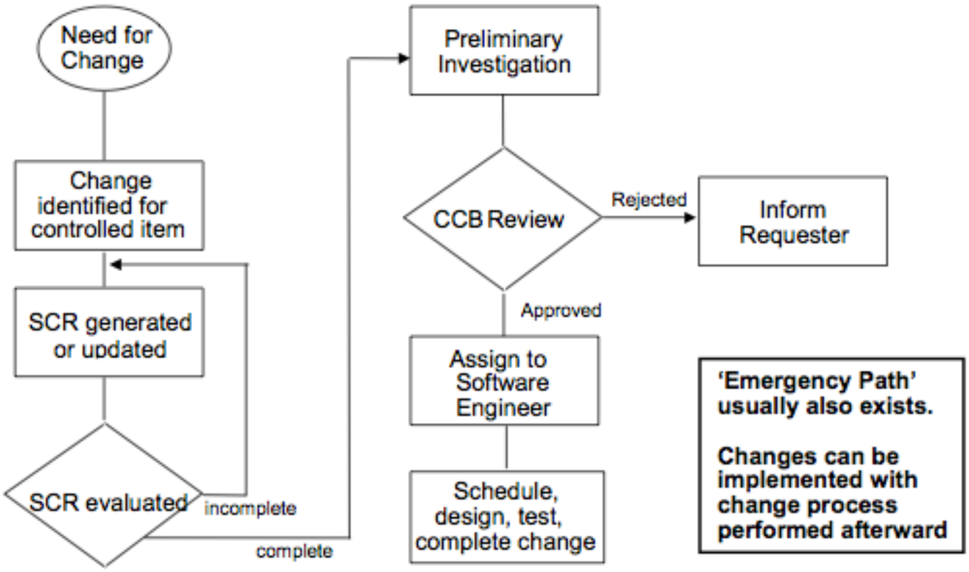

Need for Change
Change identified for controlled item
SCR generated or updated
SCR evaluated
incomplete
complete
Preliminary Investigation
CCB Review
Rejected
Inform Requester
Approved
Assign to Software Engineer
Schedule, design, test, complete change
'Emergency Path' usually also exists.
Changes can be implemented with change process performed afterward

Figure 4. Flow of change control process

2.4.2 Software Project Planning

The input for the planning is the set of requirements and defined scope, and also the feasibility. The next step is choosing a software life cycle based on the characteristics of the project, including complexity, quality requirements among others. Where relevant the project is decomposed into a hierarchical set of tasks, and the deliverable for each task is defined.

Based on the produced tasks to be done, a detailed effort, schedule, and cost estimation is undertaken. Resources are then allocated to tasks so as to optimize personnel productivity (at individual, team, and organizational levels). Risk management is planned and presented to all the relevant stakeholders for their acceptance. Processes for software quality assurance are also defined and agreed. As an iterative process, it is vital that the processes and responsibilities for ongoing plan management, review, and revision are also clearly stated and agreed.

The activities are:

2.4.2.1 Process Planning

Selection of the appropriate software life cycle and the adaptation and deployment of appropriate software life cycle processes are done according to the particular scope and requirements of the project. At the project level, appropriate methods and tools are used to decompose the project into tasks, with associated inputs, outputs, and completion conditions (for example, work breakdown structure). The output of this activity serves as information for cost and schedule calculations.

2.4.2.2 Determine Deliverables

The output/products of each task are specified, the decision of re-using previously developed software is made, and also the third party providers are selected if necessary.

2.4.2.3 Effort, Schedule, and Cost Estimation

Based on the breakdown of tasks, inputs, and outputs, the expected effort range required for each task is determined using a calibrated estimation model based on historical size-effort data where available and relevant, or other methods like expert judgment. Task dependencies are established and potential bottlenecks are identified. Bottlenecks are resolved where possible, and the expected schedule of tasks with projected start times, durations, and end times is produced. Resource requirements (people, tools) are translated into cost estimates. Consensus is an important part of this activity, which is why is suppose to be iterative.

2.4.2.4 Resource Allocation

Equipment, facilities, and people are associated with the scheduled tasks, including the allocation of responsibilities for completion (using, for example, a Gantt chart). This activity takes into account the characteristics of the resources for their optimal use.

2.4.2.5 Risk Management

Risk identification and analysis (what can go wrong, how and why, and what are the likely consequences), critical risk assessment (which are the most significant risks in terms of exposure, which can we do something about in terms of leverage), risk mitigation and contingency planning (formulating a strategy to deal with risks and to manage the risk profile) are all undertaken. Risk assessment methods (for example, decision trees and process simulations) should be used in order to highlight and evaluate risks. Project abandonment policies should also be determined at this point in discussion with all other stakeholders. Software-unique aspects of risk must influence the project's risk management.

2.4.2.6 Quality Management

Quality is defined in terms of pertinent attributes of the specific project and any associated product(s), perhaps in both quantitative and qualitative terms. These quality characteristics will have been determined in the specification of detailed software requirements.

Thresholds for adherence to quality are set for each indicator as appropriate to stakeholder expectations for the software at hand. Procedures relating to ongoing SQA throughout the process and for product (deliverable) verification and validation are also specified at this stage (for example, technical reviews and inspections).

2.4.2.7 Plan Management

How the project will be managed and how the plan will be managed must also be planned. Reporting, monitoring, and control of the project must fit the selected software engineering process and the realities of the project. All of this is required in an environment where change is expected. This requires that adherence to plans be systematically directed, monitored, reviewed, reported, and, where appropriate, revised. Plans associated with other management-oriented support processes (for example, documentation, software configuration management, and problem resolution) also need to be managed in the same manner.

2.4.3 Software Project Enactment

The plans are then implemented, and the processes embodied in the plans are enacted. Throughout, there is a focus on adherence to the plans, with an overriding expectation that such adherence will lead to the successful satisfaction of stakeholder requirements and achievement of the project objectives. Fundamental to enactment are the ongoing management activities of measuring, monitoring, controlling, and reporting.

2.4.3.1 Implementation of Plans

The project is initiated and the project activities are undertaken according to the schedule. In the process, resources are utilized (for example, personnel effort, funding) and deliverables are produced (for example, architectural design documents, test cases).

2.4.3.2 Supplier Contract Management

Prepare and execute agreements with suppliers, monitor supplier performance, and accept supplier products, incorporating them as appropriate.

2.4.3.3 Implementation of Measurement Process

The measurement process is enacted alongside the software project, ensuring that relevant and useful data are collected (see also topics 6.2 Plan the Measurement Process and 6.3 Perform the Measurement Process).

2.4.3.4 Monitor Process

Adherence to the various plans is assessed continually and at predetermined intervals. Outputs and completion conditions for each task are analyzed. Deliverables are evaluated in terms of their required characteristics (for example, via reviews and audits). Effort expenditure, schedule adherence, and costs to date are investigated, and resource usage is examined. The project risk profile is revisited, and adherence to quality requirements is evaluated.
Measurement data are modeled and analyzed. Variance analysis based on the deviation of actual from expected outcomes and values is undertaken. This may be in the form of cost overruns, schedule slippage, and the like. Outlier identification and analysis of quality and other measurement data are performed (for example, defect density analysis). Risk exposure and leverage are recalculated. Activities for enable problem detection and exception identification are run again.

2.4.3.5 Control Process

The outcomes of the process monitoring activities provide the basis on which action decisions are taken. Where appropriate, and where the impact and associated risks are modeled and managed, changes can be made to the project. This may take the form of corrective action (for example, retesting certain components), it may involve the incorporation of contingencies so that similar occurrences are avoided (for example, the decision to use prototyping to assist in software requirements validation), and/or it may entail the revision of the various plans and other project documents (for example, requirements specification) to accommodate the unexpected outcomes and their implications.

2.4.3.6 Reporting

At specified and agreed periods, adherence to the plans is reported, both within the organization (for example to the project portfolio steering committee) and to external stakeholders (for example, clients, users). Reports of this nature should focus on overall adherence as opposed to the detailed reporting required frequently within the project team.

2.4.4 Review and Evaluation

At critical points in the project, overall progress towards achievement of the stated objectives and satisfaction of stakeholder requirements are evaluated. Similarly, assessments of the effectiveness of the overall process to date, the personnel involved, and the tools and methods employed are also undertaken at particular milestones.

2.4.4.1 Determining Satisfaction of Requirements

Since attaining stakeholder (user and customer) satisfaction is one of our principal aims, it is important that progress towards this aim be formally and periodically assessed. This occurs on achievement of major project milestones (for example, confirmation of software design architecture, software integration technical review).

2.4.4.2 Reviewing and Evaluating Performance

Periodic performance reviews for project personnel provide insights as to the likelihood of adherence to plans as well as possible areas of difficulty (for example, team member conflicts). The various methods, tools, and techniques employed are evaluated for their effectiveness and appropriateness, and the process itself is systematically and periodically assessed for its relevance, utility, and efficacy in the project context. Where appropriate, changes are made and managed.

2.4.5 Closure

The project reaches closure when all the plans and embodied processes have been enacted and completed. At this stage, the criteria for project success are revisited. Once closure is established, archival, post mortem, and process improvement activities are performed.

2.4.5.1 Determining Closure

The tasks as specified in the plans are complete, and satisfactory achievement of completion criteria is confirmed. All planned products have been delivered with acceptable characteristics. Requirements are checked off and confirmed as satisfied, and the objectives of the project have been achieved. These processes generally involve all stakeholders and result in the documentation of client acceptance and any remaining known problem reports.

2.4.5.2 Closure Activities

After closure has been confirmed, archival of project materials takes place in line with stakeholder-agreed methods, location, and duration. The organization’s measurement database is updated with final project data and post-project analyses are undertaken. A project post mortem is undertaken so that issues, problems, and opportunities encountered during the process (particularly via review and evaluation) are analyzed, and lessons are drawn from the process and fed into organizational learning and improvement endeavors.

2.4.6 Software Engineering Measurement

The importance of measurement and its role in better management practices is widely acknowledged, and so its importance can only increase in the coming years. Effective measurement has become one of the cornerstones of organizational maturity.

Key terms on software measures and measurement methods have been defined in [ISO15939-02] on the basis of the ISO international vocabulary of metrology [ISO93].

This topic follows the international standard ISO/IEC 15939, which describes a process that defines the activities and tasks necessary to implement a software measurement process and includes, as well, a measurement information model.

2.4.6.1 Establish and Sustain Measurement Commitment

Accept requirements for measurement. Each measurement endeavor should be guided by organizational objectives and driven by a set of measurement requirements established by the organization and the project. For example, an organizational objective might be "first-to-market with new products." So the factors that contribute to this objective should be measured. 27

- Define scope of measurement. The organizational unit to which each measurement requirement is to be applied must be established. This may consist of a functional area, a single project, a single site, or even the whole enterprise. The 27 stakeholders should also be identified.
- Commitment of management and staff to measurement. The commitment must be formally established, communicated, and supported by resources.

Commit resources for measurement. The organization's commitment to measurement is an essential factor for success, as evidenced by assignment of resources for implementing the measurement process. Assigning resources includes allocation of responsibility for the various tasks of the measurement process (such as user, analyst, and librarian) and providing adequate funding, training, tools, and support to conduct the process in an enduring fashion.

2.4.6.2 Plan the Measurement Process

Characterize the organizational unit. The organizational unit provides the context for measurement, so it is important to make this context explicit and to articulate the assumptions that it embodies and the constraints that it imposes. Characterization can be in terms of organizational processes, application domains, technology, and organizational interfaces. An organizational process model is also typically an element of the organizational unit characterization.

Identify information needs. Information needs are based on the goals, constraints, risks, and problems of the organizational unit. They may be derived from business, organizational, regulatory, and/or product objectives. They must be identified and prioritized. Then, a subset to be addressed must be selected and the results documented, communicated, and reviewed by stakeholders.

Select measures. Candidate measures must be selected, with clear links to the information needs. Measures must then be selected based on the priorities of the information needs and other criteria such as cost of collection, degree of process disruption during collection, ease of analysis, ease of obtaining accurate, consistent data, and so on.

Define data collection, analysis, and reporting procedures. This encompasses collection procedures and schedules, storage, verification, analysis, reporting, and configuration management of data.

Define criteria for evaluating the information products. Criteria for evaluation are influenced by the technical and business objectives of the organizational unit. Information products include those associated with the product being produced, as well as those associated with the processes being used to manage and measure the project.

Review, approve, and provide resources for measurement tasks.

- The measurement plan must be reviewed and approved by the appropriate stakeholders. This includes all data collection procedures, storage, analysis, and reporting procedures; evaluation criteria; schedules; and responsibilities. Criteria for reviewing these artifacts should have been established at the organizational unit level or higher and should be used as the basis for these reviews. Such criteria should take into consideration previous experience, availability of resources, and potential disruptions to projects when changes from current practices are proposed. Approval demonstrates commitment to the measurement process.

- Resources should be made available for implementing the planned and approved measurement tasks. Resource availability may be staged in cases where changes are to be piloted before widespread deployment. Consideration should be paid to the resources necessary for successful deployment of new procedures or measures.

Acquire and deploy supporting technologies. This includes evaluation of available supporting technologies, selection of the most appropriate technologies, acquisition of those technologies, and deployment of those technologies.

2.4.6.3 Perform the Measurement Process

Integrate measurement procedures with processes. The measurement procedures, such as data collection, must be integrated into the processes they are measuring. This may involve changing current processes to accommodate data collection or generation activities. It may also involve analysis of current processes to minimize additional effort and evaluation of the effect on employees to ensure that the measurement procedures will be accepted. Morale issues and other human factors need to be considered. In addition, the measurement procedures must be communicated to those providing the data, training may need to be provided, and support must typically be provided. Data analysis and reporting procedures must typically be integrated into organizational and/or project processes in a similar manner.

Collect data. The data must be collected, verified, and stored.

Analyze data and develop information products. Data may be aggregated, transformed, or recoded as part of the analysis process, using a degree of rigor appropriate to the nature of the data and the information needs. The results of this analysis are typically indicators such as graphs, numbers, or other indications that must be interpreted, resulting in initial conclusions to be presented to stakeholders. The results and conclusions must be reviewed, using a process defined by the organization (which may be formal or informal). Data providers and measurement users should participate in reviewing the data to ensure that they are meaningful and accurate, and that they can result in reasonable actions.

2.4.6.4 Evaluate Measurement

Evaluate information products. Evaluate information products against specified evaluation criteria and determine strengths and weaknesses of the information products. This may be performed by an internal process or an external audit and should include feedback from measurement users. Record lessons learned in an appropriate database.

Evaluate the measurement process. Evaluate the measurement process against specified evaluation criteria and determine the strengths and weaknesses of the process. This may be performed by an internal process or an external audit and should include feedback from measurement users. Record lessons learned in an appropriate database.

Identify potential improvements. Such improvements may be changes in the format of indicators, changes in units measured, or reclassification of categories. Determine the costs and benefits of potential improvements and select appropriate improvement actions. Communicate proposed improvements to the measurement process owner and stakeholders for review and approval. Also communicate lack of potential improvements if the analysis fails to identify improvements.

3 PMBOK Description

3.1 Audience and Structure

In the PMBOK (Project Management Body of Knowledge) contains the globally recognized standard and guide for the project management profession. A standard is a formal document that describes established norms, methods, processes, and practices. As with other professions, the knowledge contained in this standard has evolved from the recognized good practices of project management practitioners who have contributed to the development of this standard.

As another important purpose of the PMBOK it can be pointed out the aim of having a common vocabulary for the Project Management profession for using and applying project management concepts. *Figure 5* and *6* show the overall structure of the PMBOK and its areas.

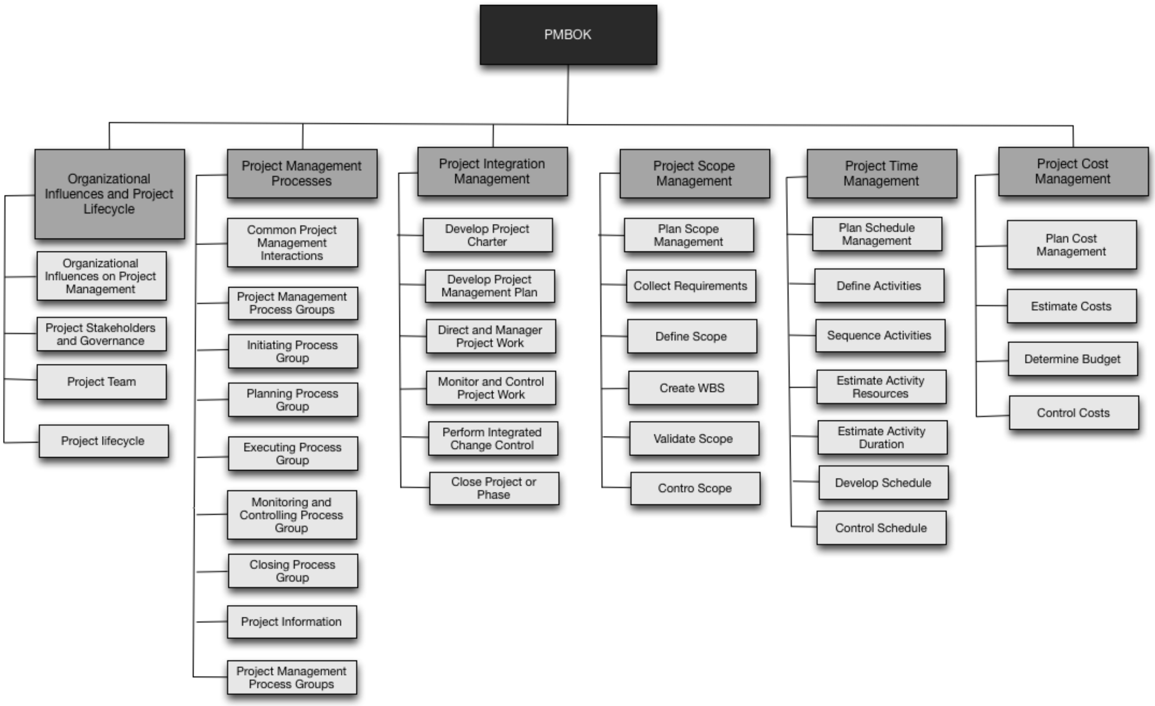

PMBOK
Organizational Influences and Project Lifecycle
Organizational Influences on Project Management
Project Stakeholders and Governance
Project Team
Project lifecycle
Project Management Processes
Common Project Management Interactions
Project Management Process Groups
Initiating Process Group
Planning Process Group
Executing Process Group
Monitoring and Controlling Process Group
Closing Process Group
Project Information
Project Management Process Groups
Project Integration Management
Develop Project Charter
Develop Project Management Plan
Direct and Manager Project Work
Monitor and Control Project Work
Perform Integrated Change Control
Close Project or Phase
Project Scope Management
Plan Scope Management
Collect Requirements
Define Scope
Create WBS
Validate Scope
Contro Scope
Project Time Management
Plan Schedule Management
Define Activities
Sequence Activities
Estimate Activity Resources
Estimate Activity Duration
Develop Schedule
Control Schedule
Project Cost Management
Plan Cost Management
Estimate Costs
Determine Budget
Control Costs

Figure 5. First six PMBOK Areas

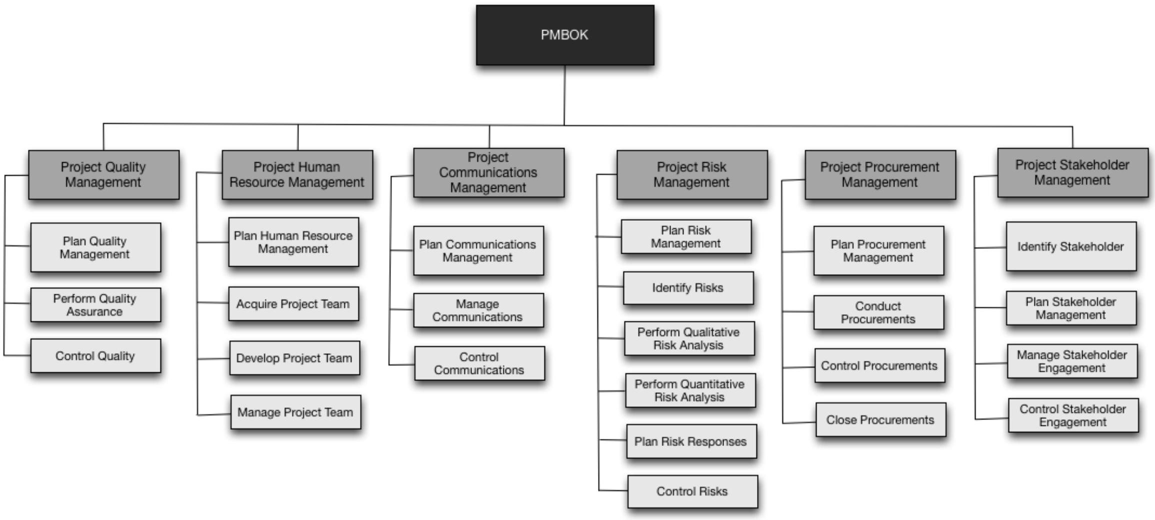

PMBOK
Project Quality Management
Plan Quality Management
Perform Quality Assurance
Control Quality
Project Human Resource Management
Plan Human Resource Management
Acquire Project Team
Develop Project Team
Manage Project Team
Project Communications Management
Plan Communications Management
Manage Communications
Control Communications
Project Risk Management
Plan Risk Management
Identify Risks
Perform Qualitative Risk Analysis
Perform Quantitative Risk Analysis
Plan Risk Responses
Control Risks
Project Procurement Management
Plan Procurement Management
Conduct Procurements
Control Procurements
Close Procurements
Project Stakeholder Management
Identify Stakeholder
Plan Stakeholder Management
Manage Stakeholder Engagement
Control Stakeholder Engagement

Figure 6. Last six PMBOK Areas

3.2 PMBOK Areas Description

3.2.1 Organizational influences and project lifecycle

The context of the project management activities is much wider than the project itself, for that reason is important to realize and understand how organizational influence, stakeholders influence and relationship between activities affect a project.

3.2.1.1 Organizational Influences on Project Management

An organization's culture, style, and structure influence how its projects are performed. The organization's level of project management maturity and its project management systems can also influence the project. When a project involves external entities such as those that are part of a joint venture or partnering agreement, the project will be influenced by more than one organization. The following are the organizational characteristics, factors, and assets within an enterprise that are likely to influence the project. The following topics are covered:

- **Organizational cultures and Styles**
- **Organizational communications**
- **Organizational Structures**
- **Organizational Process Assets**
    - Processes and Procedures
    - Corporate Knowledge Base

3.2.1.2 Project Stakeholders and Governance

A stakeholder is an individual, group, or organization who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project. Stakeholders may be actively involved in the project or have interests that may be positively or negatively affected by the performance or completion of the project. Project governance enables organizations to consistently manage projects and maximize the value of project outcomes and align the projects with business strategy. The following topics are covered:

- **Project Governance**
- **Project Success**

3.2.1.3 Project team

The project team includes the project manager and the group of individuals who act together in performing the work of the project to achieve its objectives. This team is comprised of individuals from different groups with specific subject matter knowledge or with a specific skill set to carry out the work of the project. The structure and characteristics of a project team can vary widely, but one constant is the project manager's role as the leader of the team, regardless of what authority the project manager may have over its members.

Project teams include roles such as:

- **Project management staff.** The members of the team who perform project management activities such as scheduling, budgeting, reporting and control, communications, risk management and administrative support. This role may be performed or supported by a project management office (PMO).

- **Project staff.** The members of the team who carry out the work of creating the project deliverables.

- **Supporting experts**. Supporting experts perform activities required to develop or execute the project management plan.

- **User or customer representatives**. Members of the organization who will accept the deliverables or products of the project may be assigned to act as representatives or liaisons to ensure proper coordination, advise on requirements, or validate the acceptability of the project's results.

- **Sellers**. Sellers, also called vendors, suppliers, or contractors, are external companies that enter into a contractual agreement to provide components or services necessary for the project. The project team is often assigned the responsibility to oversee the performance and acceptance of sellers' deliverables or services. If the sellers bear a large share of the risk for delivering the project's results, they may play a significant role on the project team.

- **Business partner members**. Members of business partners' organizations may be assigned as members of the project team to ensure proper coordination.

- **Business partners**. Business partners are also external companies, but they have a special relationship with the enterprise, sometimes attained through a certification process. Business partners provide specialized expertise or fill a specified role such as installation, customization, training, or support.

The **Composition of Project teams** is the main topic covered.

3.2.1.4 Project Life cycle

A project life cycle is the series of phases that a project passes through from its initiation to its closure. The phases are generally sequential, and their names and numbers are determined by the management and control needs of the organization or organizations involved in the project, the nature of the project itself, and its area of application. The phases can be broken down by functional or partial objectives, intermediate results or deliverables, specific milestones within the overall scope of work, or financial availability. Phases are generally time bounded, with a start and ending or control point. The life cycle provides the basic framework for managing the project, regardless of the specific work involved.

Project life cycles can range along a continuum from predictive or plan-driven approaches at one end to adaptive or change-driven approaches at the other. In a predictive life cycle, the product and deliverables are defined at the beginning of the project and any changes to scope are carefully managed. In an adaptive life cycle, the product is developed over multiple iterations and detailed scope is defined for each iteration only as the iteration begins.

The following topics related to project lifecycle are presented:

- **Characteristics of the Project Life cycle**
- **Project Phases**
    - **Phase-to-Phase relationships**
    - **Predictive Life cycles**
    - **Iterative and Incremental Life cycles**
    - **Adaptive Life cycles**

3.2.2 Project Management Processes

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. This application of knowledge requires the effective management of the project management processes. A process is a set of interrelated actions and activities performed to create a pre-specified product, service, or result. Each process is characterized by its inputs, the tools and techniques that can be applied, and the resulting outputs.

3.2.2.1 Common Project Management Process Interactions

The project management processes are presented as discrete elements with well-defined interfaces. However, in practice they overlap and interact in ways that are not completely detailed in this document. Most experienced project management practitioners recognize there is more than one way to manage a project.

3.2.2.2 Project Management Process Groups

The following sections identify and describe the five Project Management Process Groups required for any project. These five Process Groups have clear dependencies and are typically performed in each project and highly interact with one another. These five Process Groups are independent of application areas or industry focus. The process groups are:

- **Initiating Process Group**
- **Planning Process Group**
- **Execute Process Group**
- **Closing Process Group**
- **Monitoring and Controlling Process Group**

3.2.2.3 Initiating Process Group

The Initiating Process Group consists of those processes performed to define a new project or a new phase of an existing project by obtaining authorization to start the project or phase. Within the Initiating processes, the initial scope is defined and initial financial resources are committed.

3.2.2.4 Planning Process Group

The Planning Process Group consists of those processes performed to establish the total scope of the effort, define and refine the objectives, and develop the course of action required to attain those objectives.

3.2.2.5 Executing Process Group

The Executing Process Group consists of those processes performed to complete the work defined in the project management plan to satisfy the project specifications.

3.2.2.6 Monitoring and Controlling Process Group

The Monitoring and Controlling Process Group consists of those processes required to track, review, and orchestrate the progress and performance of the project; identify any areas in which changes to the plan are required; and initiate the corresponding changes.

3.2.2.7 Closing Process Group

The Closing Process Group consists of those processes performed to conclude all activities across all Project Management Process Groups to formally complete the project, phase, or contractual obligations.

3.2.2.8 Project Information

Throughout the life cycle of the project, a significant amount of data and information is collected, analyzed, transformed, and distributed in various formats to project team members and other stakeholders.

3.2.2.9 Role of the Knowledge Areas

The 47 project management processes identified in the *PMBOK® Guide* are further grouped into ten separate Knowledge Areas. A Knowledge Area represents a complete set of concepts, terms, and activities that make up a professional field, project management field, or area of specialization. These ten Knowledge Areas are used on most projects most of the time.

3.2.3 Project Integration Management

Project Integration Management includes the processes and activities to identify, define, combine, unify, and coordinate the various processes and project management activities within the Project Management Process Groups. Project Integration Management includes making choices about resource allocation, making trade-offs among competing objectives and alternatives, and managing the interdependencies among the project management Knowledge Areas.

3.2.3.1 Develop Project charter

Develop Project Charter is the process of developing a document that formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities. The key benefit of this process is a well-defined project start and project boundaries, creation of a formal record of the project, and a direct way for senior management to formally accept and commit to the project. *Figure 7* shows the inputs, tools and techniques and outputs expected from this process.

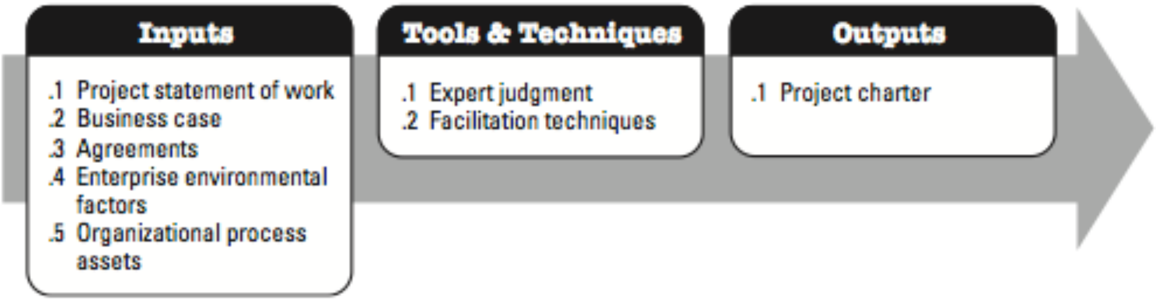

Inputs
.1 Project statement of work
.2 Business case
.3 Agreements
.4 Enterprise environmental factors
.5 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Facilitation techniques
Outputs
.1 Project charter

Figure 7. Develop Project Charter: Inputs, tools and techniques and outputs

3.2.3.2 Develop Project Management Plan

Develop Project Management Plan is the process of defining, preparing, and coordinating all subsidiary plans and integrating them into a comprehensive project management plan. *Figure 8* shows the inputs, tools and techniques and, outputs expected from this process.

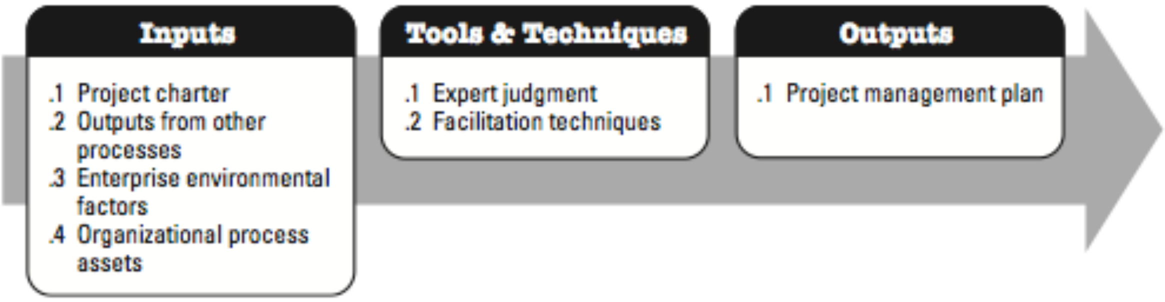

Inputs
.1 Project charter
.2 Outputs from other processes
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Facilitation techniques
Outputs
.1 Project management plan

Figure 8. Develop Project Management Plan: Inputs, tools and techniques, and outputs.

3.2.3.3 Direct and Manage Project Work

Direct and Manage Project Work is the process of leading and performing the work defined in the project management plan and implementing approved changes to achieve the project's objectives. *Figure 9* shows the inputs, tools and techniques, and outputs expected from this process.

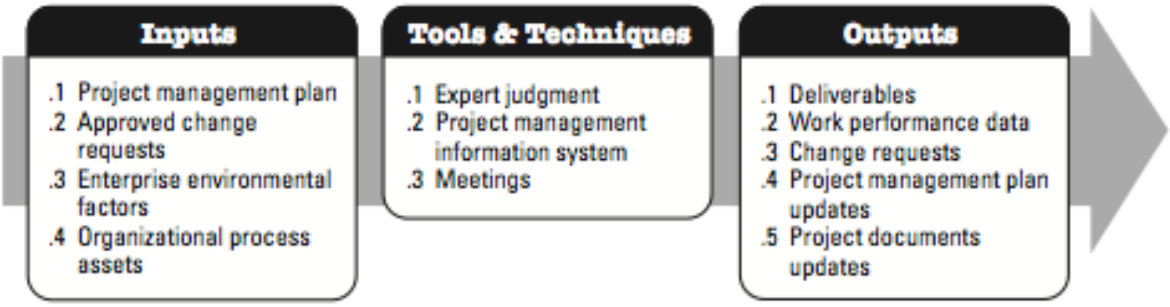
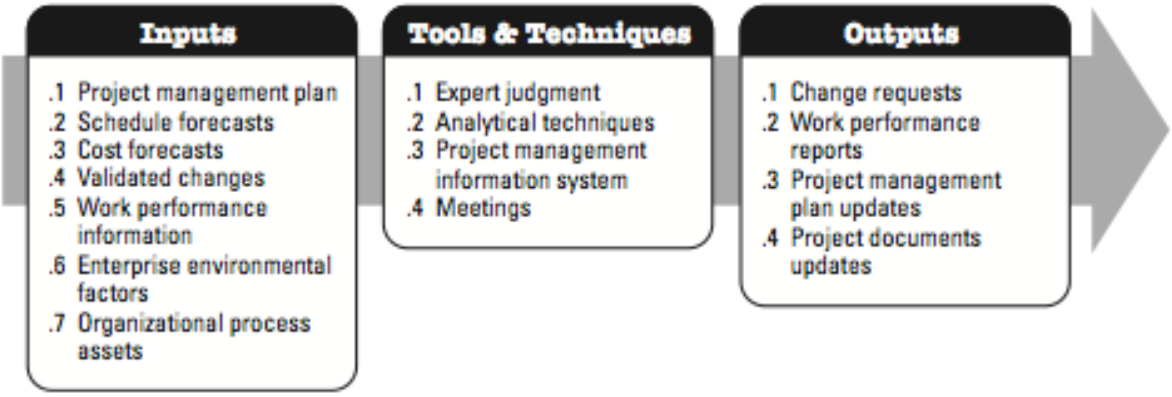

Inputs
.1 Project management plan
.2 Approved change requests
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Project management information system
.3 Meetings
Outputs
.1 Deliverables
.2 Work performance data
.3 Change requests
.4 Project management plan updates
.5 Project documents updates

Figure 9. Direct and Manage Project Work: Inputs, tools and techniques, and outputs

3.2.3.4 Monitor and Control Project Work

Monitor and Control Project Work is the process of tracking, reviewing, and reporting the progress to meet the performance objectives defined in the project management plan. *Figure 10* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Schedule forecasts
.3 Cost forecasts
.4 Validated changes
.5 Work performance information
.6 Enterprise environmental factors
.7 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Analytical techniques
.3 Project management information system
.4 Meetings

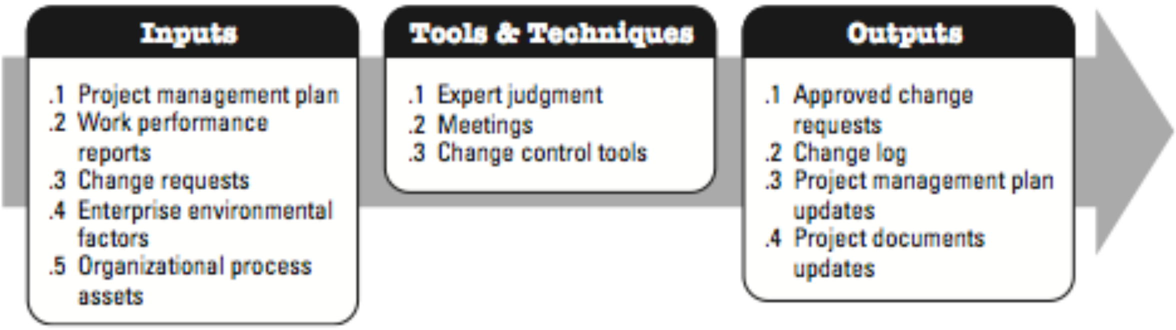
Outputs

.1 Change requests
.2 Work performance reports
.3 Project management plan updates
.4 Project documents updates

Figure 10. Monitor and Control Project Work: Inputs, tools and techniques, and outputs

3.2.3.5 Perform Integrated Change Control

Perform Integrated Change Control is the process of reviewing all change requests; approving changes and managing changes to deliverables, organizational process assets, project documents, and the project management plan; and communicating their disposition. *Figure 11* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Work performance reports
.3 Change requests
.4 Enterprise environmental factors
.5 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Meetings
.3 Change control tools

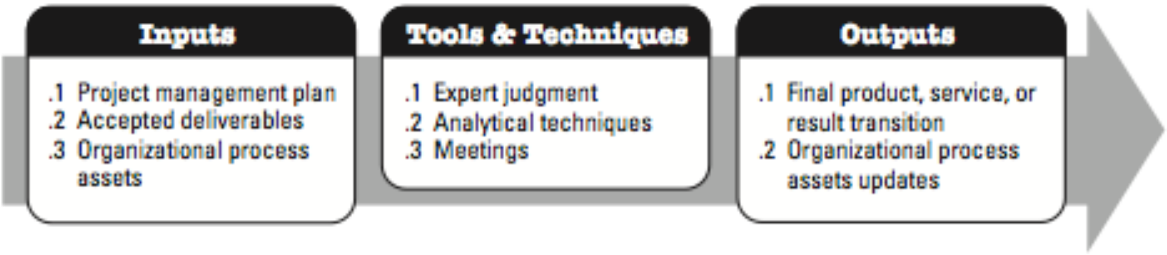
Outputs

.1 Approved change requests
.2 Change log
.3 Project management plan updates
.4 Project documents updates

Figure 11. Perform Integrated Change Control: Inputs, tools and techniques, and outputs

3.2.3.6 Close Project or Phase

Close Project or Phase is the process of finalizing all activities across all of the Project Management Process Groups to formally complete the project or phase. *Figure 12* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Accepted deliverables
.3 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Analytical techniques
.3 Meetings

Outputs

.1 Final product, service, or result transition
.2 Organizational process assets updates

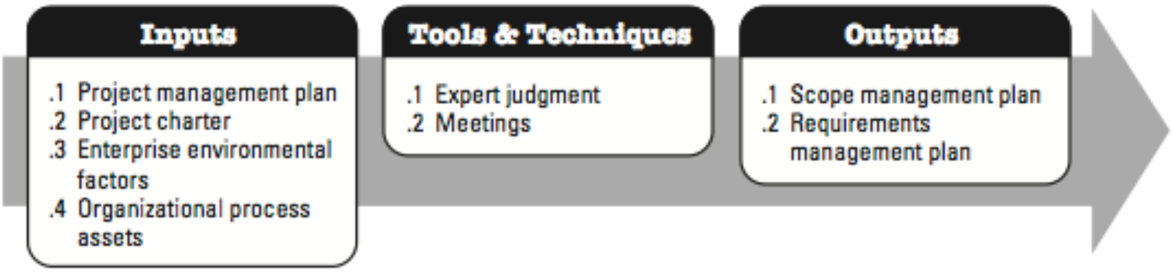
Figure 12. Close Project or Phase: Inputs, tools and techniques, and output

3.2.4 Project scope management

Project Scope Management includes the processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully. Managing the project scope is primarily concerned with defining and controlling what is and is not included in the project.

3.2.4.1 Plan Scope Management

Plan Scope Management is the process of creating a scope management plan that documents how the project scope will be defined, validated, and controlled. *Figure 13* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Project charter
.3 Enterprise environmental factors
.4 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Meetings

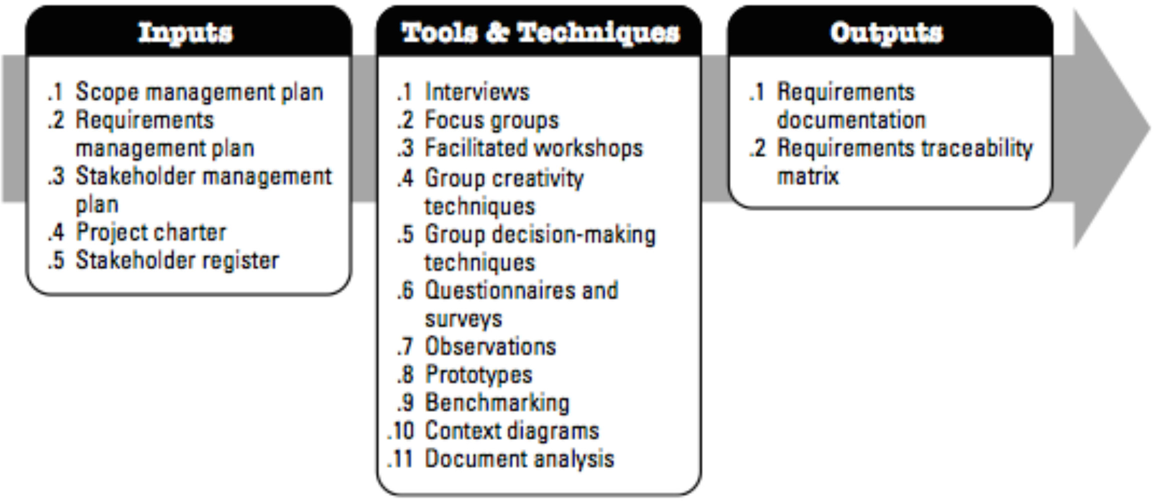
Outputs

.1 Scope management plan
.2 Requirements management plan

Figure 13. Plan Scope Management: Inputs, tools and techniques, and outputs

3.2.4.2 Collect requirements

The requirements management plan is a component of the project management plan that describes how requirements will be analyzed, documented, and managed.



Inputs

.1 Scope management plan
.2 Requirements management plan
.3 Stakeholder management plan
.4 Project charter
.5 Stakeholder register

Tools & Techniques

.1 Interviews
.2 Focus groups
.3 Facilitated workshops
.4 Group creativity techniques
.5 Group decision-making techniques
.6 Questionnaires and surveys
.7 Observations
.8 Prototypes
.9 Benchmarking
.10 Context diagrams
.11 Document analysis

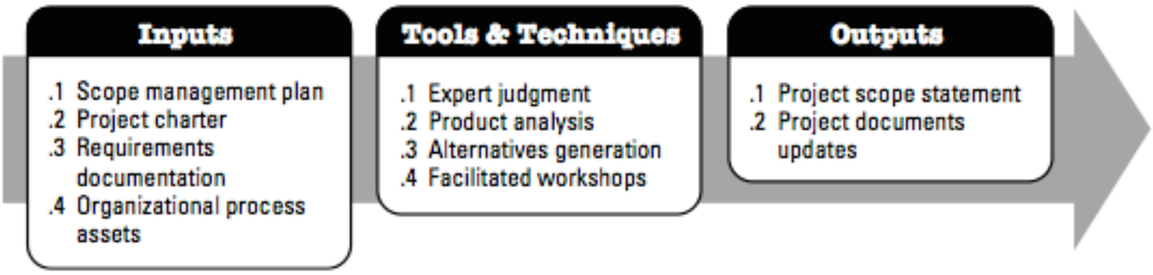
Outputs

.1 Requirements documentation
.2 Requirements traceability matrix

Figure 14. Collect Requirements: Inputs, tools and techniques, and outputs

3.2.4.3 Define Scope

Define Scope is the process of developing a detailed description of the project and product. The key benefit of this process is that it describes the project, service, or result boundaries by defining which of the requirements collected will be included in and excluded from the project scope. *Figure 15* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Scope management plan
.2 Project charter
.3 Requirements documentation
.4 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Product analysis
.3 Alternatives generation
.4 Facilitated workshops

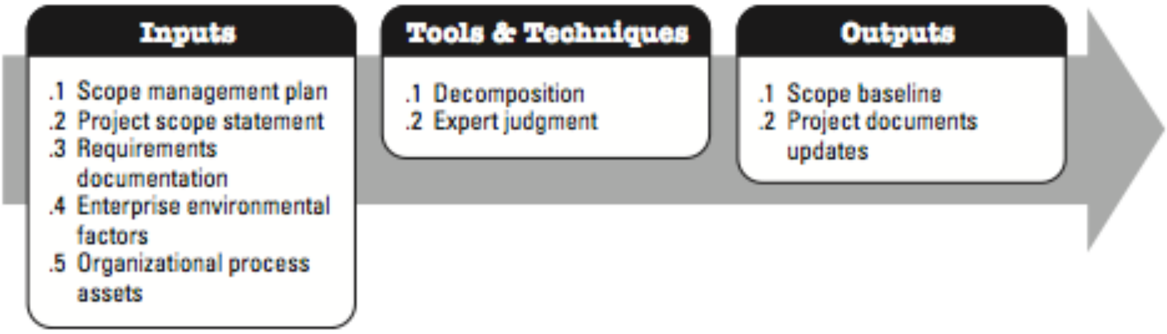
Outputs

.1 Project scope statement
.2 Project documents updates

Figure 15. Define Scope: Inputs, tools and techniques, and outputs

3.2.4.4 Create WBS

Create WBS is the process of subdividing project deliverables and project work into smaller, more manageable components. The key benefit of this process is that it provides a structured vision of what has to be delivered. *Figure 16* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Scope management plan
.2 Project scope statement
.3 Requirements documentation
.4 Enterprise environmental factors
.5 Organizational process assets

Tools & Techniques

.1 Decomposition
.2 Expert judgment

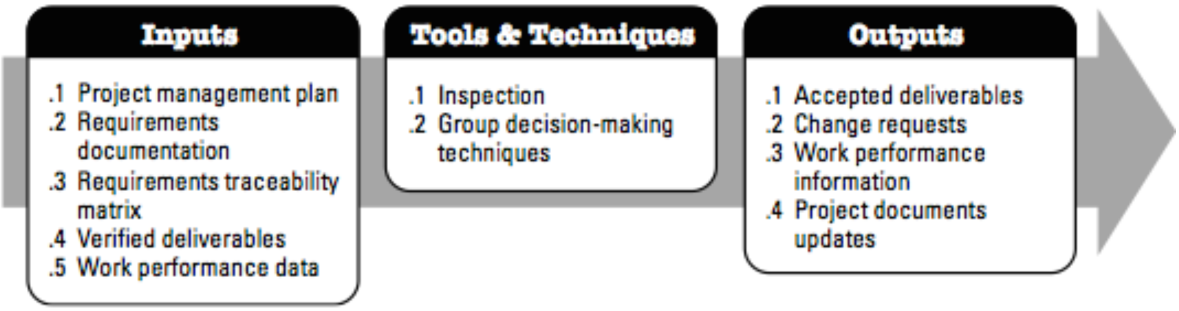
Outputs

.1 Scope baseline
.2 Project documents updates

Figure 16. Create WBS: Inputs, tools & techniques, and outputs

3.2.4.5 Validate Scope

Validate Scope is the process of formalizing acceptance of the completed project deliverables. The key benefit of this process is that it brings objectivity to the acceptance process and increases the chance of final product, service, or result acceptance by validating each deliverable. *Figure 17* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Requirements documentation
.3 Requirements traceability matrix
.4 Verified deliverables
.5 Work performance data

Tools & Techniques

.1 Inspection
.2 Group decision-making techniques

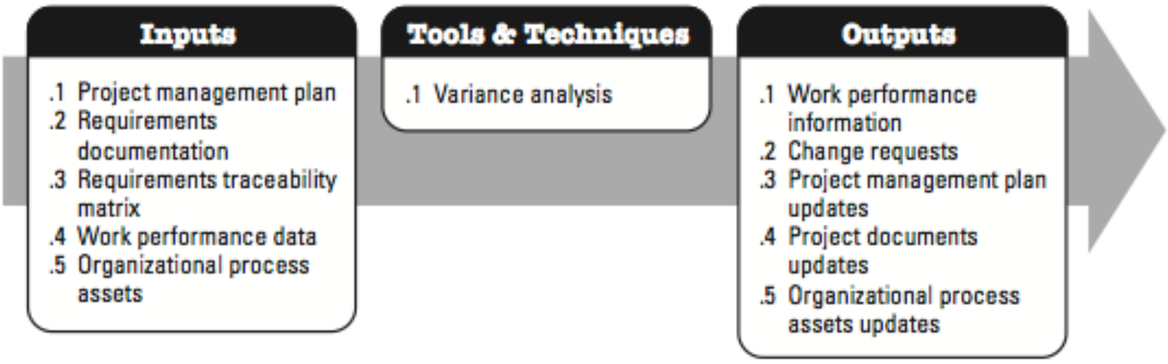
Outputs

.1 Accepted deliverables
.2 Change requests
.3 Work performance information
.4 Project documents updates

Figure 17. Validate Scope: Inputs, tools and techniques, and outputs

3.2.4.6 Control Scope

The completed deliverables that have not been formally accepted are documented, along with the reasons for nonacceptance of those deliverables. Those deliverables may require a change request for defect repair. The change requests are processed for review and disposition through the Perform Integrated Change Control process. *Figure 18* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Requirements documentation
.3 Requirements traceability matrix
.4 Work performance data
.5 Organizational process assets

Tools & Techniques

.1 Variance analysis

Outputs

.1 Work performance information
.2 Change requests
.3 Project management plan updates
.4 Project documents updates
.5 Organizational process assets updates

Figure 18. Control Scope: Inputs, tools and techniques, and outputs

3.2.5 Project Time Management

Project Time Management includes the processes required to manage the timely completion of the project.

3.2.5.1 Plan Schedule Management

Plan Schedule Management is the process of establishing the policies, procedures, and documentation for planning, developing, managing, executing, and controlling the project schedule. The key benefit of this process is that it provides guidance and direction on how the project schedule will be managed throughout the project. *Figure 19* shows the inputs, tools and techniques and, outputs expected from this process.

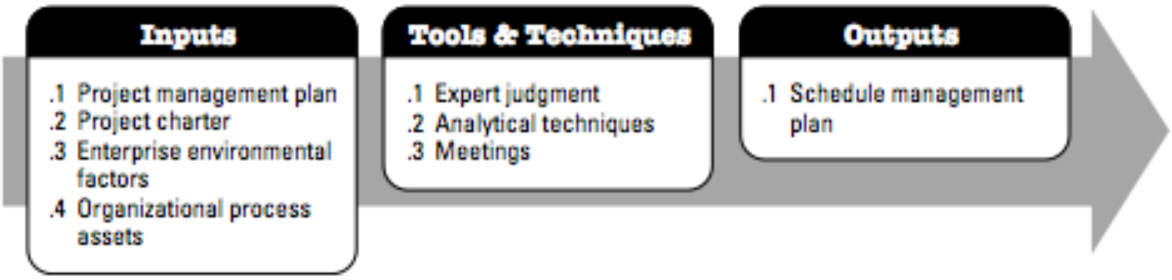

Inputs
.1 Project management plan
.2 Project charter
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Analytical techniques
.3 Meetings
Outputs
.1 Schedule management plan

Figure 19. Plan Schedule Management: Inputs, tools & techniques, and outputs

3.2.5.2 Define Activities

Define Activities is the process of identifying and documenting the specific actions to be performed to produce the project deliverables. The key benefit of this process is to break down work packages into activities that provide a basis for estimating, scheduling, executing, monitoring, and controlling the project work. *Figure 20* shows the inputs, tools and techniques and, outputs expected from this process.

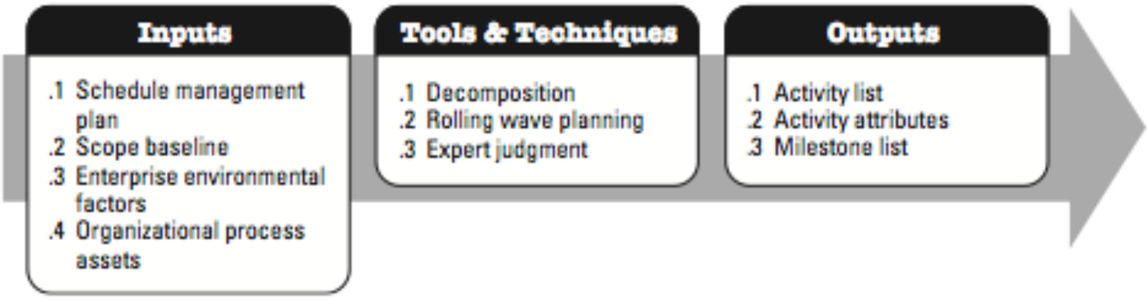


Figure 20. Define Activities: Inputs, tools & techniques, and outputs

3.2.5.3 Sequence Activities

Sequence Activities is the process of identifying and documenting relationships among the project activities. The key benefit of this process is that it defines the logical sequence of work to obtain the greatest efficiency given all project constraints. *Figure 21* shows the inputs, tools and techniques and, outputs expected from this process.

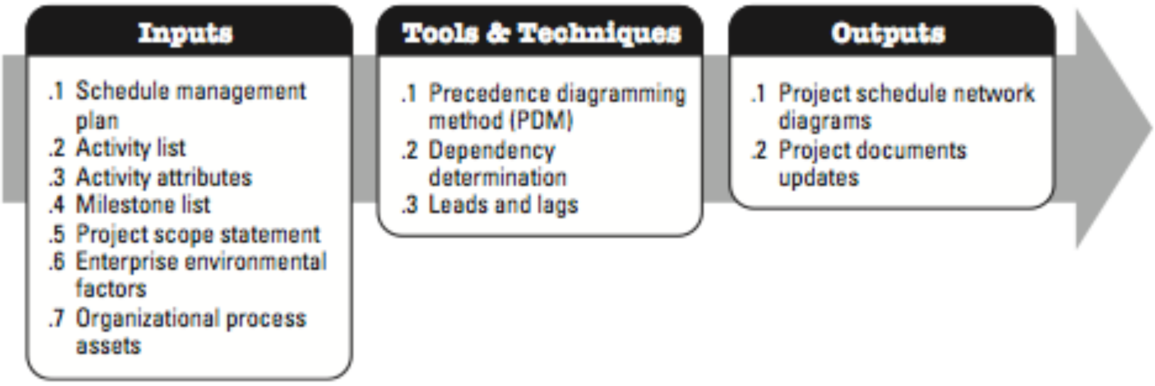

Inputs
.1 Schedule management plan
.2 Activity list
.3 Activity attributes
.4 Milestone list
.5 Project scope statement
.6 Enterprise environmental factors
.7 Organizational process assets
Tools & Techniques
.1 Precedence diagramming method (PDM)
.2 Dependency determination
.3 Leads and lags
Outputs
.1 Project schedule network diagrams
.2 Project documents updates

Figure 21. Sequence Activities: Inputs, tools & techniques, and outputs

3.2.5.4 *Estimate Activity Resources*

Estimate Activity Resources is the process of estimating the type and quantities of material, human resources, equipment, or supplies required to perform each activity. *Figure 22* shows the inputs, tools and techniques and, outputs expected from this process.

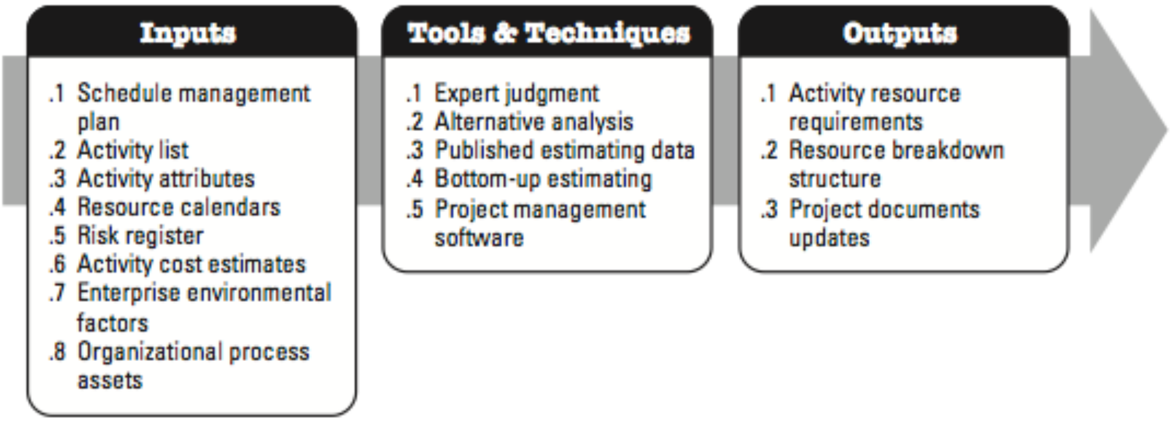
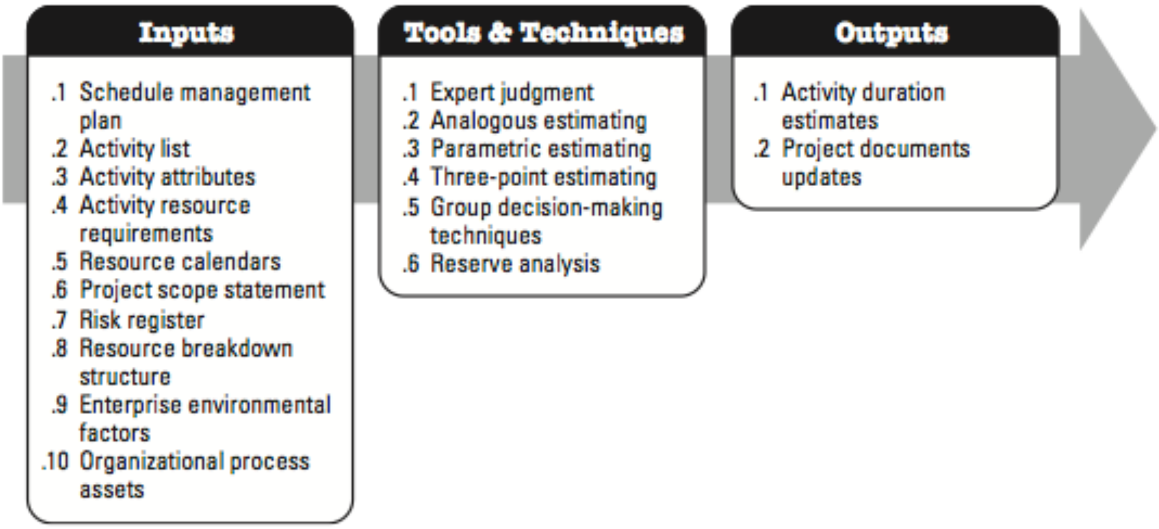

Inputs
.1 Schedule management plan
.2 Activity list
.3 Activity attributes
.4 Resource calendars
.5 Risk register
.6 Activity cost estimates
.7 Enterprise environmental factors
.8 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Alternative analysis
.3 Published estimating data
.4 Bottom-up estimating
.5 Project management software
Outputs
.1 Activity resource requirements
.2 Resource breakdown structure
.3 Project documents updates

Figure 22. Estimate Activity Resources: Inputs, tools and techniques, and outputs

3.2.5.5 *Estimate Activity Durations*

Estimate Activity Durations is the process of estimating the number of work periods needed to complete individual activities with estimated resources. *Figure 23* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

- .1 Schedule management plan
- .2 Activity list
- .3 Activity attributes
- .4 Activity resource requirements
- .5 Resource calendars
- .6 Project scope statement
- .7 Risk register
- .8 Resource breakdown structure
- .9 Enterprise environmental factors
- .10 Organizational process assets

Tools & Techniques

- .1 Expert judgment
- .2 Analogous estimating
- .3 Parametric estimating
- .4 Three-point estimating
- .5 Group decision-making techniques
- .6 Reserve analysis

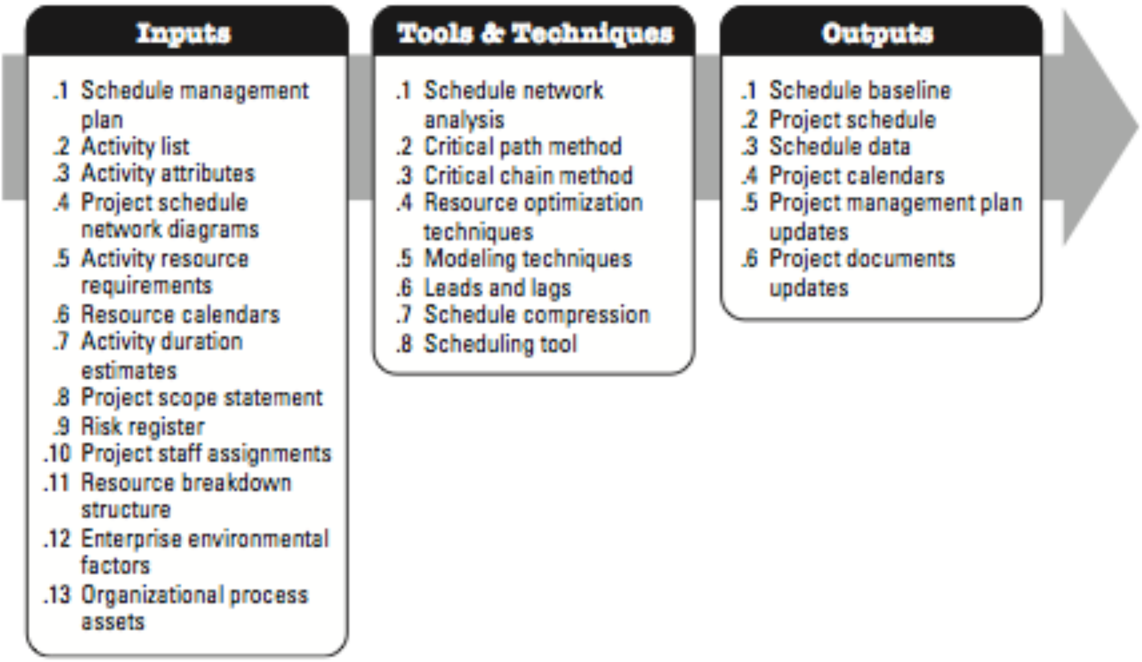
Outputs

- .1 Activity duration estimates
- .2 Project documents updates

Figure 23. Estimate Activity Durations: Inputs, tools and techniques, and outputs

3.2.5.6 Develop Schedule

Develop Schedule is the process of analyzing activity sequences, durations, resource requirements, and schedule constraints to create the project schedule model. *Figure 24* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

- .1 Schedule management plan
- .2 Activity list
- .3 Activity attributes
- .4 Project schedule network diagrams
- .5 Activity resource requirements
- .6 Resource calendars
- .7 Activity duration estimates
- .8 Project scope statement
- .9 Risk register
- .10 Project staff assignments
- .11 Resource breakdown structure
- .12 Enterprise environmental factors
- .13 Organizational process assets

Tools & Techniques

- .1 Schedule network analysis
- .2 Critical path method
- .3 Critical chain method
- .4 Resource optimization techniques
- .5 Modeling techniques
- .6 Leads and lags
- .7 Schedule compression
- .8 Scheduling tool

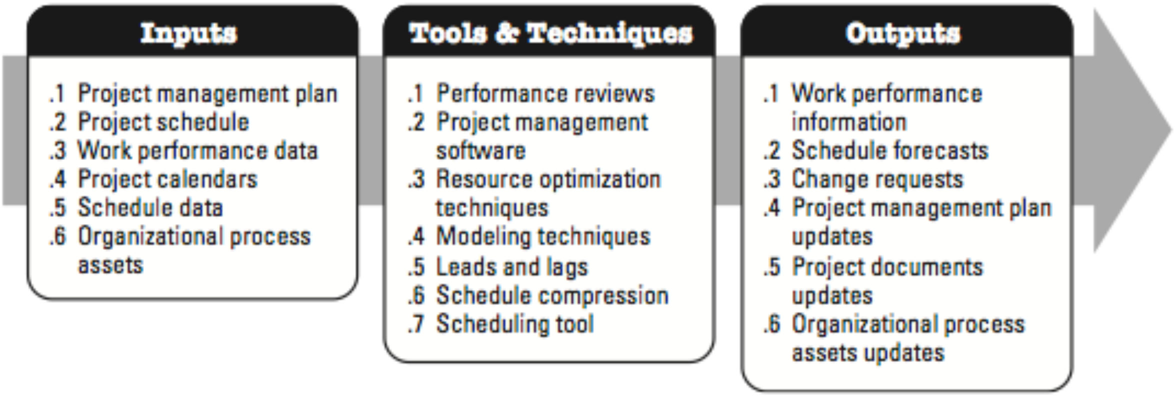
Outputs

- .1 Schedule baseline
- .2 Project schedule
- .3 Schedule data
- .4 Project calendars
- .5 Project management plan updates
- .6 Project documents updates

Figure 24. Develop Schedule: Inputs, tools and techniques, and outputs

3.2.5.7 Control Schedule

Control Schedule is the process of monitoring the status of project activities to update project progress and manage changes to the schedule baseline to achieve the plan. *Figure 25* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Project schedule
.3 Work performance data
.4 Project calendars
.5 Schedule data
.6 Organizational process assets

Tools & Techniques

.1 Performance reviews
.2 Project management software
.3 Resource optimization techniques
.4 Modeling techniques
.5 Leads and lags
.6 Schedule compression
.7 Scheduling tool

Outputs

.1 Work performance information
.2 Schedule forecasts
.3 Change requests
.4 Project management plan updates
.5 Project documents updates
.6 Organizational process assets updates

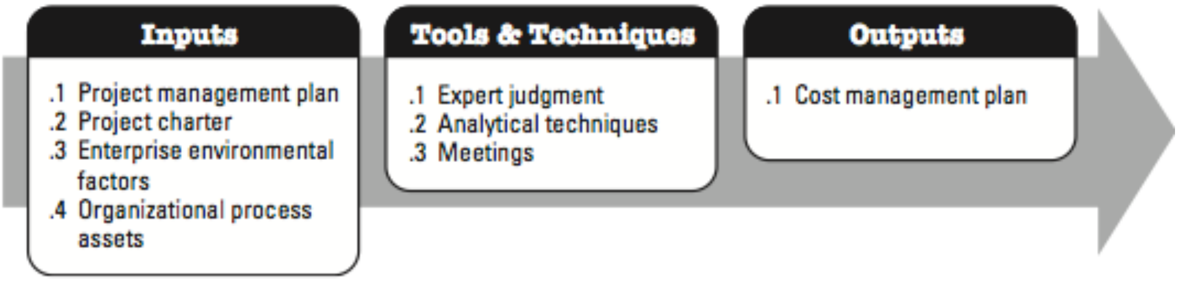
Figure 25. Control Schedule: Inputs, tools and techniques, and outputs

3.2.6 Project Cost Management

Project Cost Management includes the processes involved in planning, estimating, budgeting, financing, funding, managing, and controlling costs so that the project can be completed within the approved budget.

3.2.6.1 Plan Cost Management

Plan Cost Management is the process that establishes the policies, procedures, and documentation for planning, managing, expending, and controlling project costs. *Figure 26* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Project charter
.3 Enterprise environmental factors
.4 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Analytical techniques
.3 Meetings

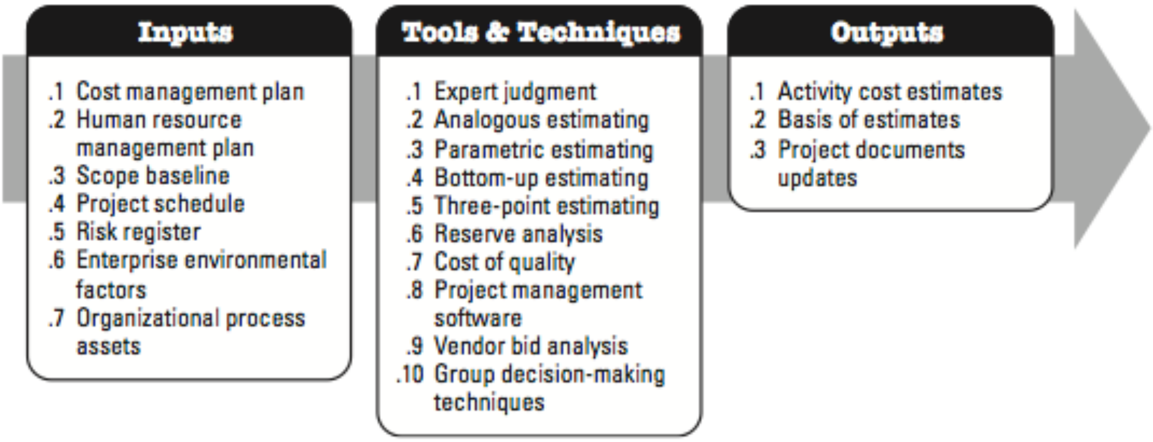
Outputs

.1 Cost management plan

Figure 26. Plan Cost Management: Inputs, tools and techniques, and outputs

3.2.6.2 Estimate Costs

Estimate Costs is the process of developing an approximation of the monetary resources needed to complete project activities. *Figure 27* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Cost management plan
.2 Human resource management plan
.3 Scope baseline
.4 Project schedule
.5 Risk register
.6 Enterprise environmental factors
.7 Organizational process assets

Tools & Techniques

.1 Expert judgment
.2 Analogous estimating
.3 Parametric estimating
.4 Bottom-up estimating
.5 Three-point estimating
.6 Reserve analysis
.7 Cost of quality
.8 Project management software
.9 Vendor bid analysis
.10 Group decision-making techniques

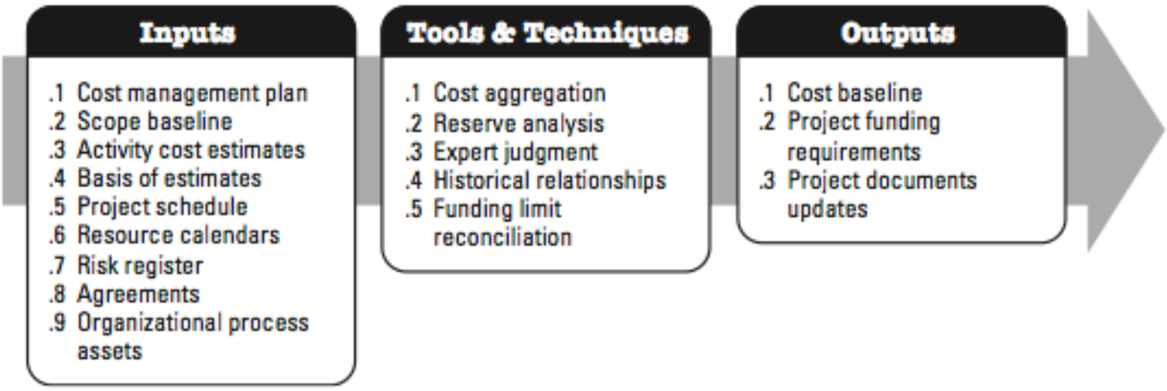
Outputs

.1 Activity cost estimates
.2 Basis of estimates
.3 Project documents updates

Figure 27. Estimate Costs: Inputs, tools and techniques, and outputs

3.2.6.3 Determine Budget

Determine Budget is the process of aggregating the estimated costs of individual activities or work packages to establish an authorized cost baseline. *Figure 28* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Cost management plan
.2 Scope baseline
.3 Activity cost estimates
.4 Basis of estimates
.5 Project schedule
.6 Resource calendars
.7 Risk register
.8 Agreements
.9 Organizational process assets

Tools & Techniques

.1 Cost aggregation
.2 Reserve analysis
.3 Expert judgment
.4 Historical relationships
.5 Funding limit reconciliation

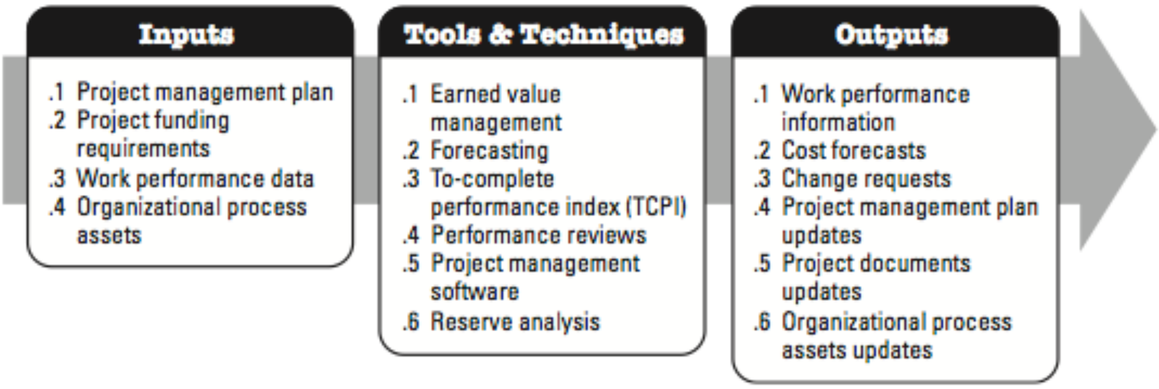
Outputs

.1 Cost baseline
.2 Project funding requirements
.3 Project documents updates

Figure 28. Determine Budget: Inputs, tools and techniques, and outputs

3.2.6.4 Control Costs

Control Costs is the process of monitoring the status of the project to update the project costs and managing changes to the cost baseline. *Figure 29* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Project funding requirements
.3 Work performance data
.4 Organizational process assets

Tools & Techniques

.1 Earned value management
.2 Forecasting
.3 To-complete performance index (TCPI)
.4 Performance reviews
.5 Project management software
.6 Reserve analysis

Outputs

.1 Work performance information
.2 Cost forecasts
.3 Change requests
.4 Project management plan updates
.5 Project documents updates
.6 Organizational process assets updates

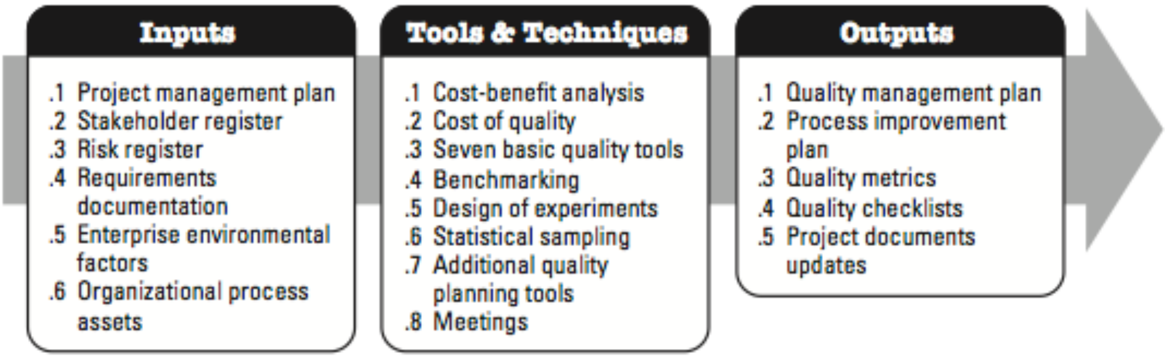
Figure 29. Control Costs: Inputs, tools and techniques, and outputs

3.2.7 Project Quality Management

Project Quality Management includes the processes and activities of the performing organization that determine quality policies, objectives, and responsibilities so that the project will satisfy the needs for which it was undertaken. Project Quality Management uses policies and procedures to implement, within the project's context, the organization's quality management system and, as appropriate, it supports continuous process improvement activities as undertaken on behalf of the performing organization.

3.2.7.1 Plan Quality Management

Plan Quality Management is the process of identifying quality requirements and/or standards for the project and its deliverables, and documenting how the project will demonstrate compliance with relevant quality requirements. *Figure 30* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Stakeholder register
.3 Risk register
.4 Requirements documentation
.5 Enterprise environmental factors
.6 Organizational process assets

Tools & Techniques

.1 Cost-benefit analysis
.2 Cost of quality
.3 Seven basic quality tools
.4 Benchmarking
.5 Design of experiments
.6 Statistical sampling
.7 Additional quality planning tools
.8 Meetings

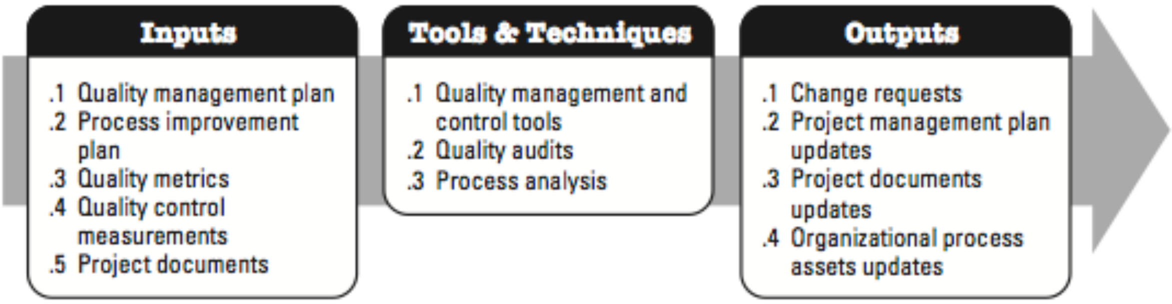
Outputs

.1 Quality management plan
.2 Process improvement plan
.3 Quality metrics
.4 Quality checklists
.5 Project documents updates

Figure 30. Plan Quality Management: Inputs, tools and techniques, and outputs

3.2.7.2 Perform Quality Assurance

Perform Quality Assurance is the process of auditing the quality requirements and the results from quality control measurements to ensure that appropriate quality standards and operational definitions are used. *Figure 31* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Quality management plan
.2 Process improvement plan
.3 Quality metrics
.4 Quality control measurements
.5 Project documents

Tools & Techniques

.1 Quality management and control tools
.2 Quality audits
.3 Process analysis

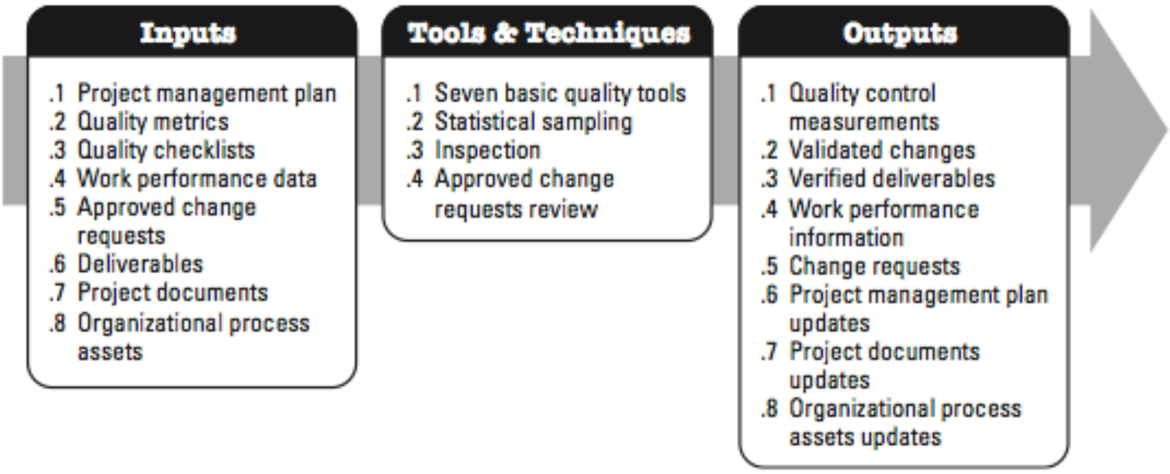
Outputs

.1 Change requests
.2 Project management plan updates
.3 Project documents updates
.4 Organizational process assets updates

Figure 31. Perform Quality Assurance: Inputs, tools and techniques, and outputs

3.2.7.3 Control Quality

Control Quality is the process of monitoring and recording results of executing the quality activities to assess performance and recommend necessary changes. *Figure 32* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Quality metrics
.3 Quality checklists
.4 Work performance data
.5 Approved change requests
.6 Deliverables
.7 Project documents
.8 Organizational process assets

Tools & Techniques

.1 Seven basic quality tools
.2 Statistical sampling
.3 Inspection
.4 Approved change requests review

Outputs

.1 Quality control measurements
.2 Validated changes
.3 Verified deliverables
.4 Work performance information
.5 Change requests
.6 Project management plan updates
.7 Project documents updates
.8 Organizational process assets updates

Figure 32. Control Quality: Inputs, tools and techniques, and outputs

3.2.8 Project Human Resource Management

Project Human Resource Management includes the processes that organize, manage, and lead the project team. The project team is comprised of the people with assigned roles and responsibilities for completing the project.

3.2.8.1 Plan Human Resource Management

Plan Human Resource Management is the process of identifying and documenting project roles, responsibilities, required skills, reporting relationships, and creating a staffing management plan. *Figure 33* shows the inputs, tools and techniques and, outputs expected from this process.

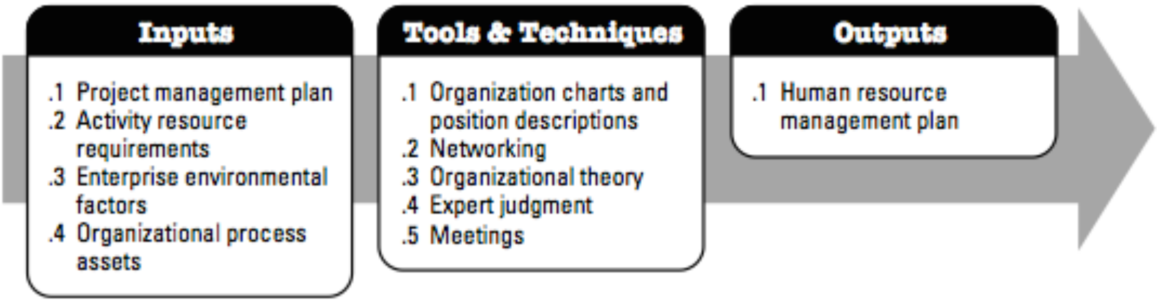

Inputs
.1 Project management plan
.2 Activity resource requirements
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Organization charts and position descriptions
.2 Networking
.3 Organizational theory
.4 Expert judgment
.5 Meetings
Outputs
.1 Human resource management plan

Figure 33. Plan Human Resource Management: Inputs, tools and techniques, and outputs

3.2.8.2 Acquire Project Team

Acquire Project Team is the process of confirming human resource availability and obtaining the team necessary to complete project activities. *Figure 34* shows the inputs, tools and techniques and, outputs expected from this process.

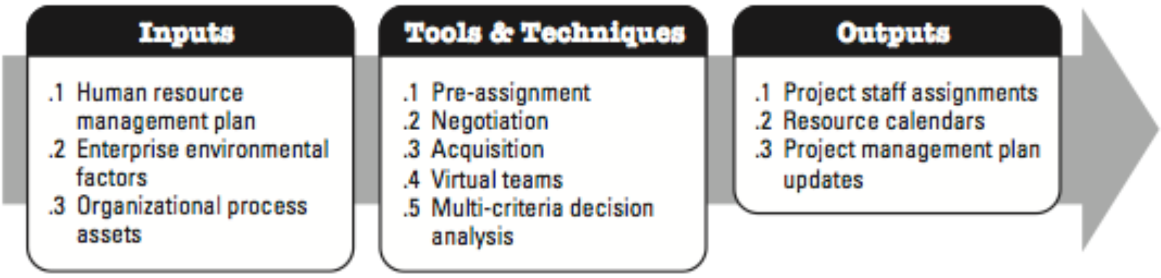

Inputs
.1 Human resource management plan
.2 Enterprise environmental factors
.3 Organizational process assets
Tools & Techniques
.1 Pre-assignment
.2 Negotiation
.3 Acquisition
.4 Virtual teams
.5 Multi-criteria decision analysis
Outputs
.1 Project staff assignments
.2 Resource calendars
.3 Project management plan updates

Figure 34. Acquire Project Team: Inputs, tools and techniques, and outputs

3.2.8.3 Develop Project Team

Develop Project Team is the process of improving competencies, team member interaction, and overall team environment to enhance project performance. *Figure 35* shows the inputs, tools and techniques and, outputs expected from this process.

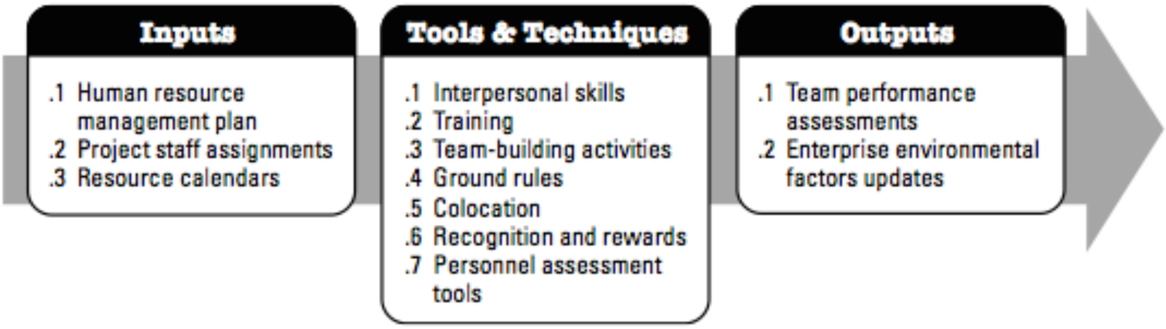

Inputs
.1 Human resource management plan
.2 Project staff assignments
.3 Resource calendars
Tools & Techniques
.1 Interpersonal skills
.2 Training
.3 Team-building activities
.4 Ground rules
.5 Colocation
.6 Recognition and rewards
.7 Personnel assessment tools
Outputs
.1 Team performance assessments
.2 Enterprise environmental factors updates

Figure 35. Develop Project Team: Inputs, tools and techniques, and outputs

3.2.8.4 Manage Project Team

Manage Project Team is the process of tracking team member performance, providing feedback, resolving issues, and managing team changes to optimize project performance. *Figure 36* shows the inputs, tools and techniques and, outputs expected from this process.

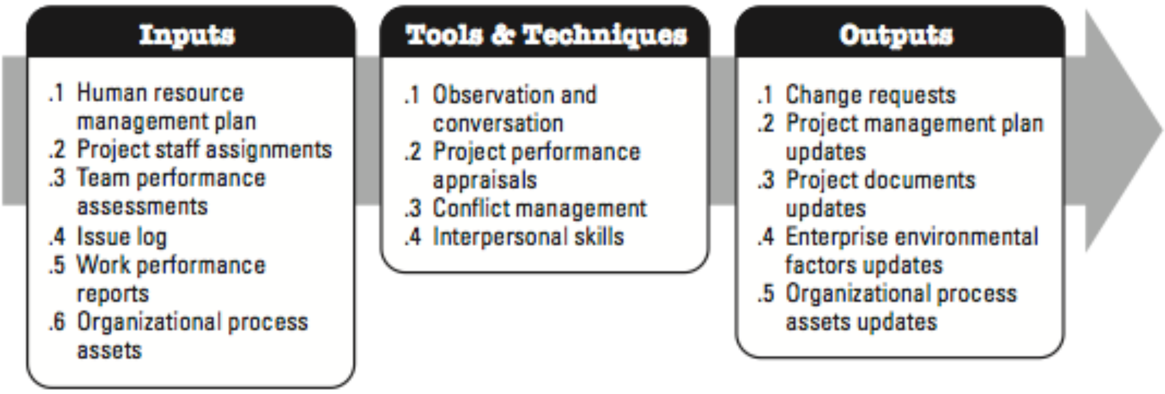

Inputs
.1 Human resource management plan
.2 Project staff assignments
.3 Team performance assessments
.4 Issue log
.5 Work performance reports
.6 Organizational process assets
Tools & Techniques
.1 Observation and conversation
.2 Project performance appraisals
.3 Conflict management
.4 Interpersonal skills
Outputs
.1 Change requests
.2 Project management plan updates
.3 Project documents updates
.4 Enterprise environmental factors updates
.5 Organizational process assets updates

Figure 36. Manage Project Team: Inputs, tools and techniques, and outputs

3.2.9 Project Communications Management

Project Communications Management includes the processes that are required to ensure timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and the ultimate disposition of project information.

3.2.9.1 Plan Communications Management

Plan Communications Management is the process of developing an appropriate approach and plan for project communications based on stakeholder's information needs and requirements, and available organizational assets. *Figure 37* shows the inputs, tools and techniques and, outputs expected from this process.

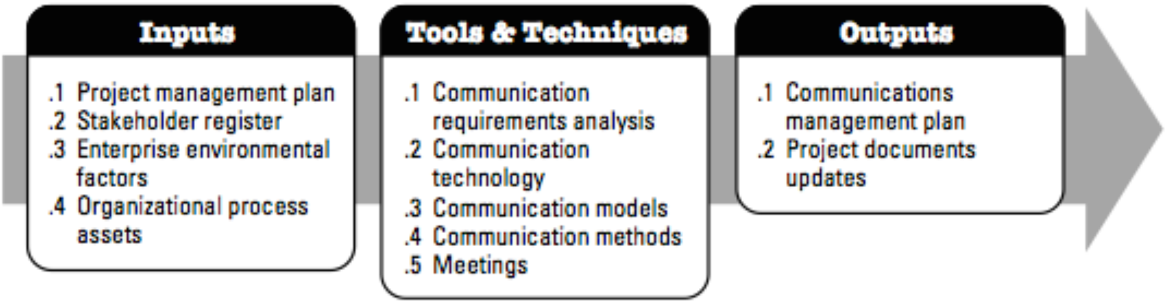
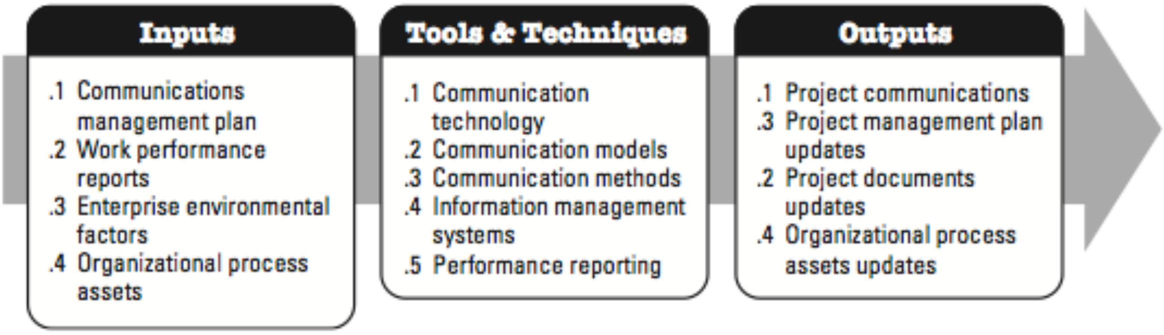

Inputs
.1 Project management plan
.2 Stakeholder register
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Communication requirements analysis
.2 Communication technology
.3 Communication models
.4 Communication methods
.5 Meetings
Outputs
.1 Communications management plan
.2 Project documents updates

Figure 37. Plan Communications Management: Inputs, tools and techniques, and outputs

3.2.9.2 Manage Communications

Manage Communications is the process of creating, collecting, distributing, storing, retrieving, and the ultimate disposition of project information in accordance to the communications management plan. *Figure 38* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Communications management plan
.2 Work performance reports
.3 Enterprise environmental factors
.4 Organizational process assets

Tools & Techniques

.1 Communication technology
.2 Communication models
.3 Communication methods
.4 Information management systems
.5 Performance reporting

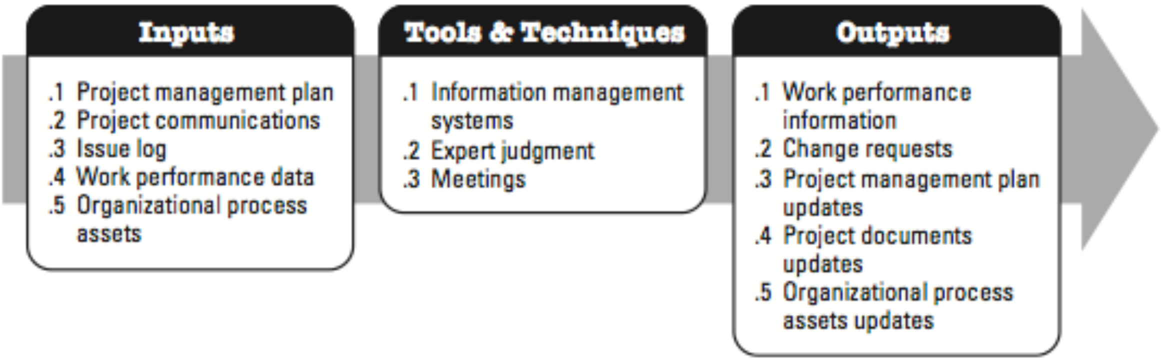
Outputs

.1 Project communications
.3 Project management plan updates
.2 Project documents updates
.4 Organizational process assets updates

Figure 38. Manage Communications: Inputs, tools and techniques, and outputs

3.2.9.3 Control Communications

Control Communications is the process of monitoring and controlling communications throughout the entire project life cycle to ensure the information needs of the project stakeholders are met. *Figure 39* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Project communications
.3 Issue log
.4 Work performance data
.5 Organizational process assets

Tools & Techniques

.1 Information management systems
.2 Expert judgment
.3 Meetings

Outputs

.1 Work performance information
.2 Change requests
.3 Project management plan updates
.4 Project documents updates
.5 Organizational process assets updates

Figure 39. Control Communications: Inputs, tools and techniques, and outputs

3.2.10 Project Risk Management

Project Risk Management includes the processes of conducting risk management planning, identification, analysis, response planning, and controlling risk on a project. The objectives of project risk management are to increase the likelihood and impact of positive events, and decrease the likelihood and impact of negative events in the project.

3.2.10.1 Plan Risk Management

Plan Risk Management is the process of defining how to conduct risk management activities for a project. The key benefit of this process is it ensures that the degree, type, and visibility of risk management are commensurate with both the risks and the importance of the project to the organization. *Figure 40* shows the inputs, tools and techniques and, outputs expected from this process.

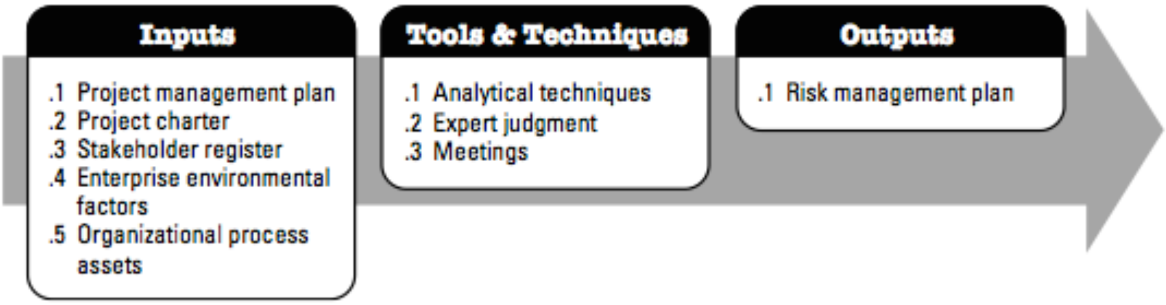


Figure 40. Plan Risk Management: Inputs, tools and techniques, and outputs

3.2.10.2 Identify Risks

Identify Risks is the process of determining which risks may affect the project and documenting their characteristics. *Figure 41* shows the inputs, tools and techniques and, outputs expected from this process.

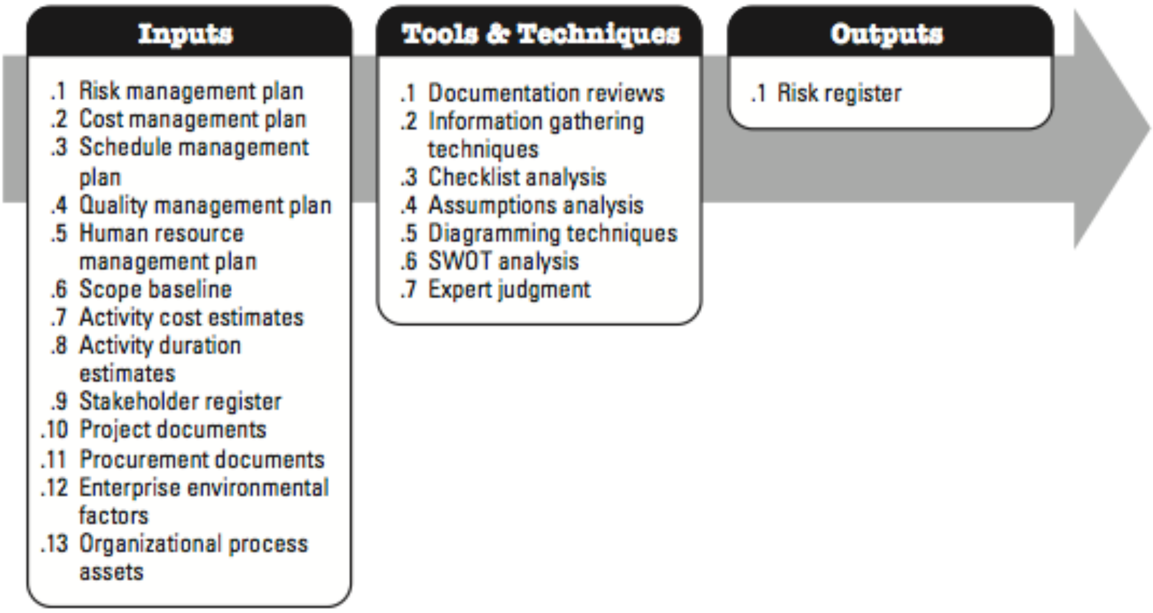
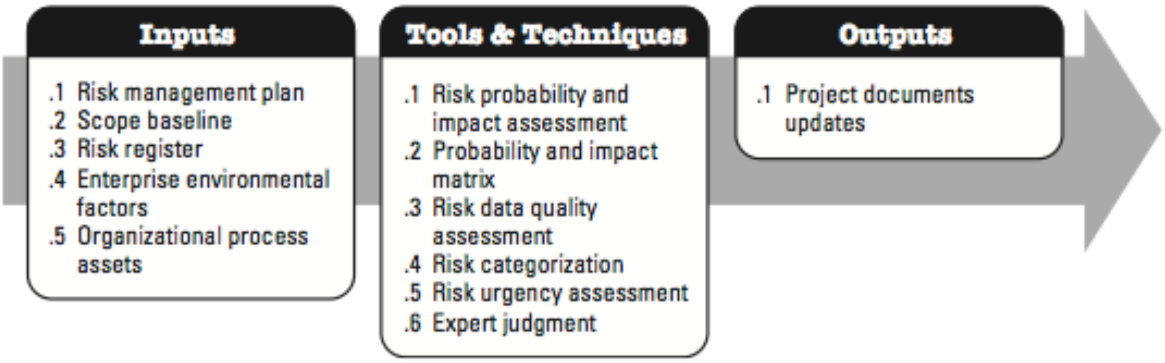

Inputs
.1 Risk management plan
.2 Cost management plan
.3 Schedule management plan
.4 Quality management plan
.5 Human resource management plan
.6 Scope baseline
.7 Activity cost estimates
.8 Activity duration estimates
.9 Stakeholder register
.10 Project documents
.11 Procurement documents
.12 Enterprise environmental factors
.13 Organizational process assets
Tools & Techniques
.1 Documentation reviews
.2 Information gathering techniques
.3 Checklist analysis
.4 Assumptions analysis
.5 Diagramming techniques
.6 SWOT analysis
.7 Expert judgment
Outputs
.1 Risk register

Figure 41. Identify Risks: Inputs, tools and techniques, and outputs

3.2.10.3 Perform Qualitative Risk Analysis

Perform Qualitative Risk Analysis is the process of prioritizing risks for further analysis or action by assessing and combining their probability of occurrence and impact. *Figure 42* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Risk management plan
.2 Scope baseline
.3 Risk register
.4 Enterprise environmental factors
.5 Organizational process assets

Tools & Techniques

.1 Risk probability and impact assessment
.2 Probability and impact matrix
.3 Risk data quality assessment
.4 Risk categorization
.5 Risk urgency assessment
.6 Expert judgment

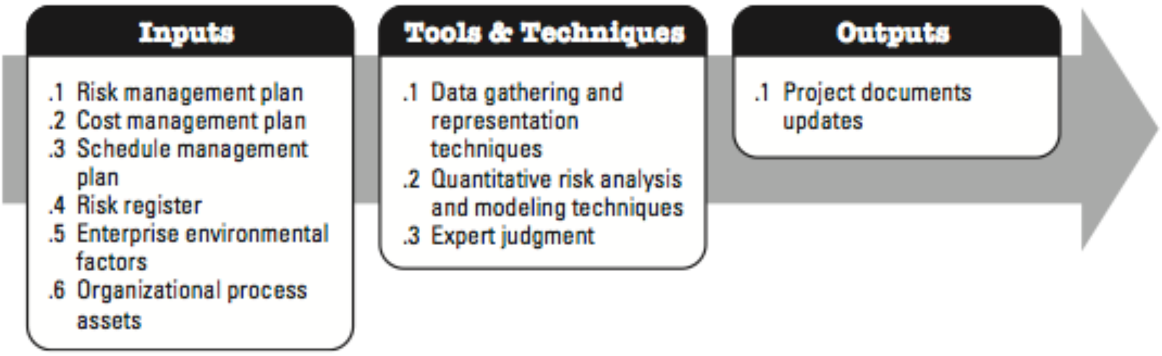
Outputs

.1 Project documents updates

Figure 42. Perform Quality Risk Analysis: Inputs, tools and techniques, and outputs

3.2.10.4 Perform Quantitative Risk Analysis

Perform Quantitative Risk Analysis is the process of numerically analyzing the effect of identified risks on overall project objectives. *Figure 43* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Risk management plan
.2 Cost management plan
.3 Schedule management plan
.4 Risk register
.5 Enterprise environmental factors
.6 Organizational process assets

Tools & Techniques

.1 Data gathering and representation techniques
.2 Quantitative risk analysis and modeling techniques
.3 Expert judgment

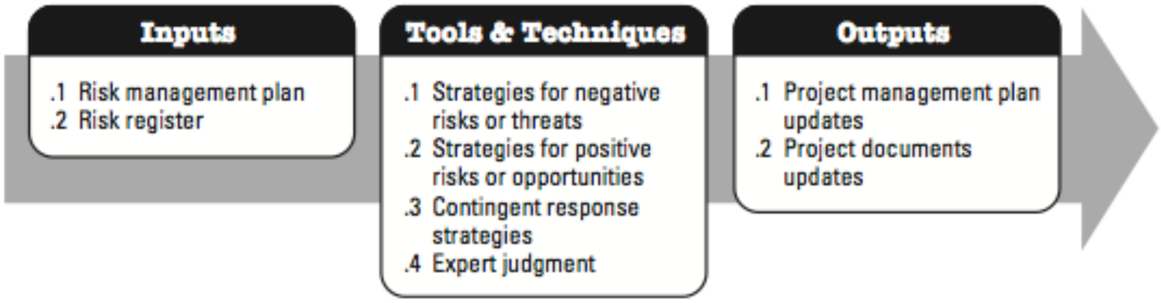
Outputs

.1 Project documents updates

Figure 43. Perform Quantitative Risk Analysis: Inputs, tools and techniques, and outputs

3.2.10.5 Plan Risk Responses

Plan Risk Responses is the process of developing options and actions to enhance opportunities and to reduce threats to project objectives. *Figure 44* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Risk management plan
.2 Risk register

Tools & Techniques

.1 Strategies for negative risks or threats
.2 Strategies for positive risks or opportunities
.3 Contingent response strategies
.4 Expert judgment

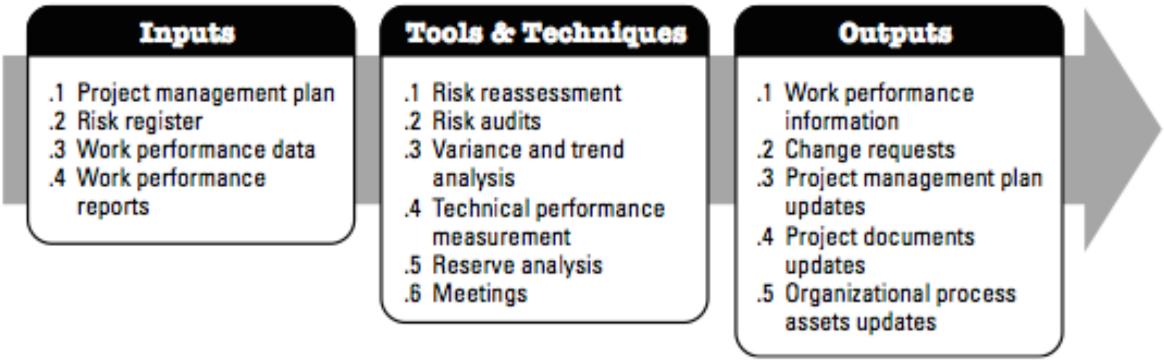
Outputs

.1 Project management plan updates
.2 Project documents updates

Figure 44. Plan Risk Responses: Inputs, tools and techniques, and outputs

3.2.10.6 Control Risks

Control Risks is the process of implementing risk response plans, tracking identified risks, monitoring residual risks, identifying new risks, and evaluating risk process effectiveness throughout the project. *Figure 45* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Risk register
.3 Work performance data
.4 Work performance reports

Tools & Techniques

.1 Risk reassessment
.2 Risk audits
.3 Variance and trend analysis
.4 Technical performance measurement
.5 Reserve analysis
.6 Meetings

Outputs

.1 Work performance information
.2 Change requests
.3 Project management plan updates
.4 Project documents updates
.5 Organizational process assets updates

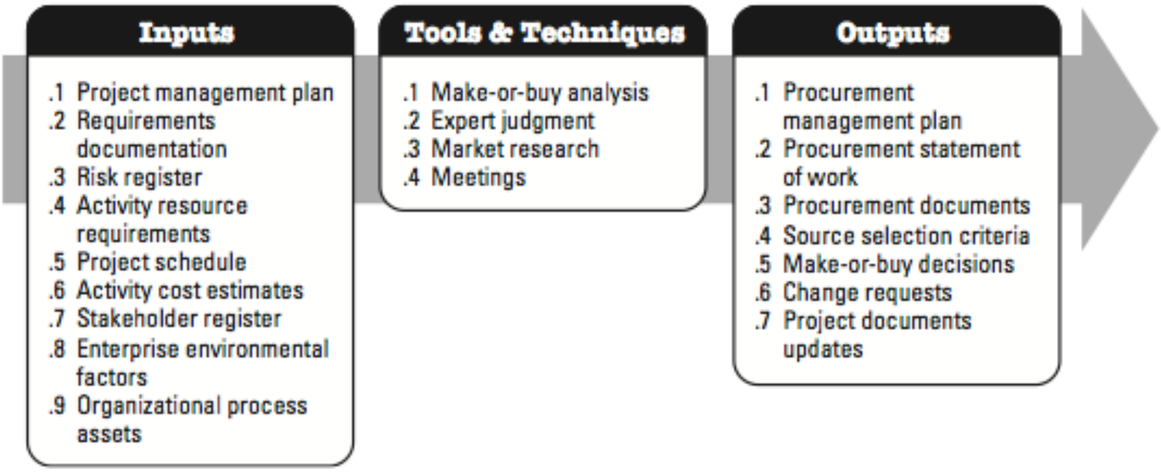
Figure 45. Control Risks: Inputs, tools and techniques, and outputs

3.2.11 Project Procurement Management

Project Procurement Management includes the processes necessary to purchase or acquire products, services, or results needed from outside the project team. The organization can be either the buyer or seller of the products, services, or results of a project.

3.2.11.1 Plan Procurement Management

Plan Procurement Management is the process of documenting project procurement decisions, specifying the approach, and identifying potential sellers. *Figure 46* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs

.1 Project management plan
.2 Requirements documentation
.3 Risk register
.4 Activity resource requirements
.5 Project schedule
.6 Activity cost estimates
.7 Stakeholder register
.8 Enterprise environmental factors
.9 Organizational process assets

Tools & Techniques

.1 Make-or-buy analysis
.2 Expert judgment
.3 Market research
.4 Meetings

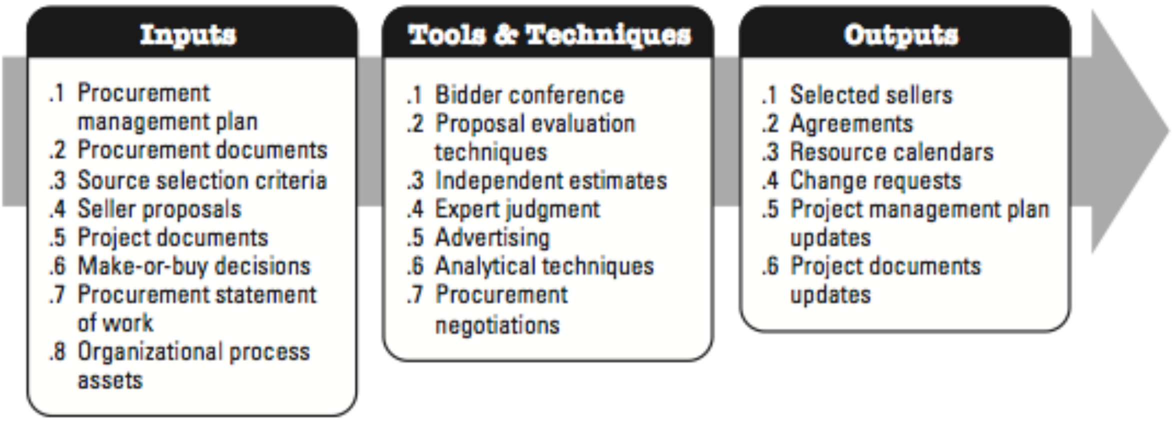
Outputs

.1 Procurement management plan
.2 Procurement statement of work
.3 Procurement documents
.4 Source selection criteria
.5 Make-or-buy decisions
.6 Change requests
.7 Project documents updates

Figure 46. Plan Procurement Management: Inputs, tools and techniques, and outputs

3.2.11.2 Conduct Procurements

Conduct Procurements is the process of obtaining seller responses, selecting a seller, and awarding a contract. *Figure 47* shows the inputs, tools and techniques and, outputs expected from this process.

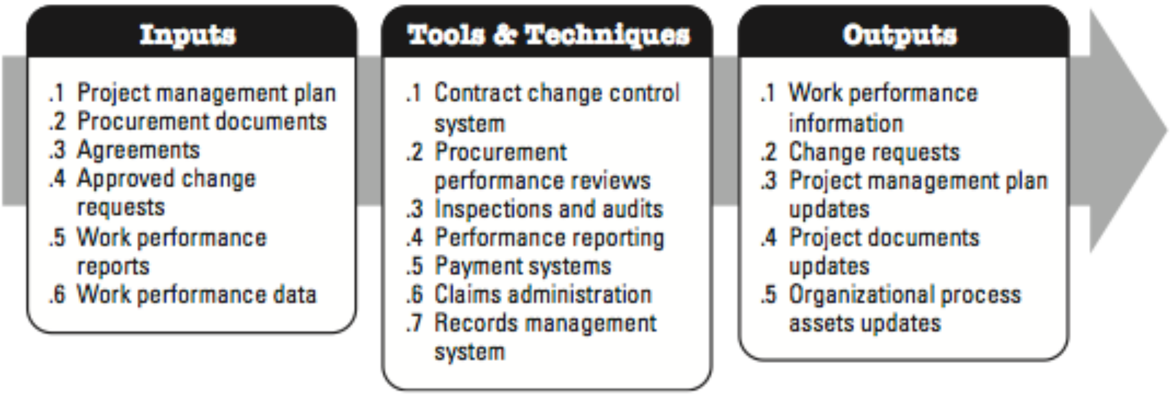


Inputs	Tools & Techniques	Outputs
.1 Procurement management plan .2 Procurement documents .3 Source selection criteria .4 Seller proposals .5 Project documents .6 Make-or-buy decisions .7 Procurement statement of work .8 Organizational process assets	.1 Bidder conference .2 Proposal evaluation techniques .3 Independent estimates .4 Expert judgment .5 Advertising .6 Analytical techniques .7 Procurement negotiations	.1 Selected sellers .2 Agreements .3 Resource calendars .4 Change requests .5 Project management plan updates .6 Project documents updates

Figure 47. Conduct Procurements: Inputs, tools and techniques, and outputs

3.2.11.3 Control Procurements

Control Procurements is the process of managing procurement relationships, monitoring contract performance, and making changes and corrections to contracts as appropriate. *Figure 48* shows the inputs, tools and techniques and, outputs expected from this process.



Inputs	Tools & Techniques	Outputs
.1 Project management plan .2 Procurement documents .3 Agreements .4 Approved change requests .5 Work performance reports .6 Work performance data	.1 Contract change control system .2 Procurement performance reviews .3 Inspections and audits .4 Performance reporting .5 Payment systems .6 Claims administration .7 Records management system	.1 Work performance information .2 Change requests .3 Project management plan updates .4 Project documents updates .5 Organizational process assets updates

Figure 48. Control Procurements: Inputs, tools and techniques, and outputs

3.2.11.4 Close Procurements

Close Procurements is the process of completing each procurement. *Figure 49* shows the inputs, tools and techniques and, outputs expected from this process.

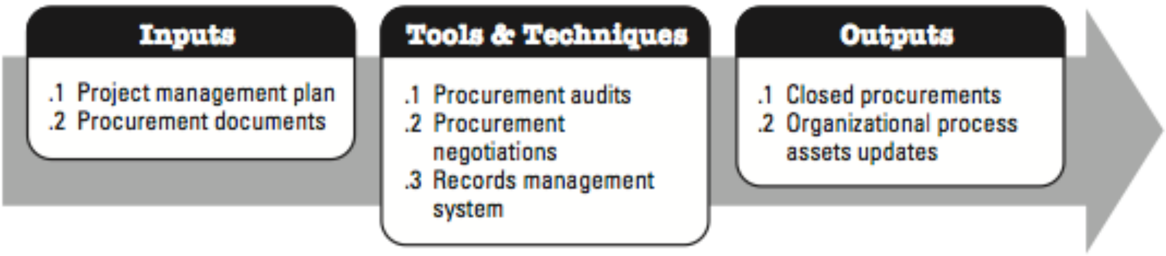

Inputs
.1 Project management plan
.2 Procurement documents
Tools & Techniques
.1 Procurement audits
.2 Procurement negotiations
.3 Records management system
Outputs
.1 Closed procurements
.2 Organizational process assets updates

Figure 49. Close Procurements: Inputs, tools and techniques, and outputs

3.2.12 Project Stakeholder Management

Project Stakeholder Management includes the processes required to identify the people, groups, or organizations that could impact or be impacted by the project, to analyze stakeholder expectations and their impact on the project, and to develop appropriate management strategies for effectively engaging stakeholders in project decisions and execution.

3.2.12.1 Identify Stakeholders

Identify Stakeholders is the process of identifying the people, groups, or organizations that could impact or be impacted by a decision, activity, or outcome of the project, analyzing and documenting relevant information regarding their interests, involvement, interdependencies, influence, and potential impact on project success. *Figure 50* shows the inputs, tools and techniques and, outputs expected from this process.

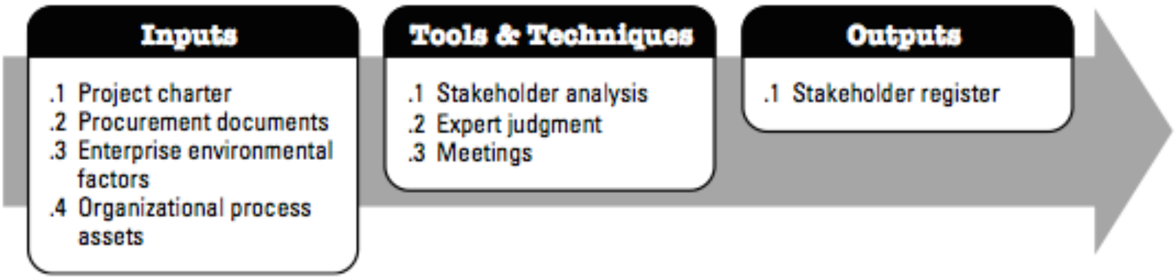


Figure 50. Identify Stakeholders: Inputs, tools and techniques, and outputs

3.2.12.2 Plan Stakeholder Management

Plan Stakeholder Management is the process of developing appropriate management strategies to effectively engage stakeholders throughout the project life cycle, based on the analysis of their needs, interests, and potential impact on project success. *Figure 51* shows the inputs, tools and techniques and, outputs expected from this process.

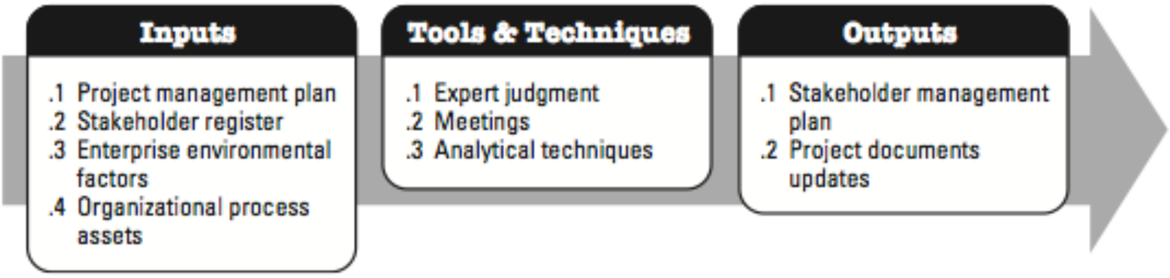

Inputs
.1 Project management plan
.2 Stakeholder register
.3 Enterprise environmental factors
.4 Organizational process assets
Tools & Techniques
.1 Expert judgment
.2 Meetings
.3 Analytical techniques
Outputs
.1 Stakeholder management plan
.2 Project documents updates

Figure 51. Plan Stakeholders Management: Inputs, tools and techniques, and outputs

3.2.12.3 Manage Stakeholder Engagement

Manage Stakeholder Engagement is the process of communicating and working with stakeholders to meet their needs/expectations, address issues as they occur, and foster appropriate stakeholder engagement in project activities throughout the project life cycle. *Figure 52* shows the inputs, tools and techniques and, outputs expected from this process.

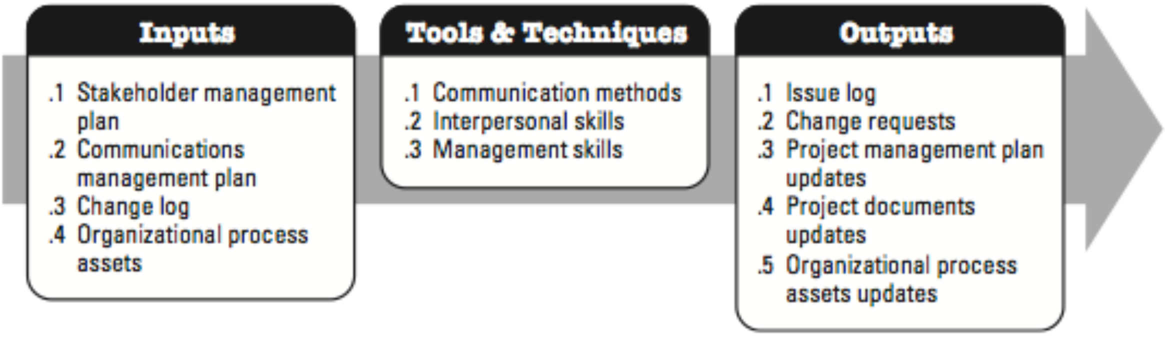

Inputs
.1 Stakeholder management plan
.2 Communications management plan
.3 Change log
.4 Organizational process assets
Tools & Techniques
.1 Communication methods
.2 Interpersonal skills
.3 Management skills
Outputs
.1 Issue log
.2 Change requests
.3 Project management plan updates
.4 Project documents updates
.5 Organizational process assets updates

Figure 52. Manage Stakeholders Engagement: Inputs, tools and techniques, and outputs

3.2.12.4 Control Stakeholder Engagement

Control Stakeholder Engagement is the process of monitoring overall project stakeholder relationships and adjusting strategies and plans for engaging stakeholders.

4 GSwE2009 Description

4.1 Outcomes

The expected outcomes from a graduated student that goes through a maser level that satisfies the GSwE2009 recommendations will:

- Master the CBOK.
- Master software engineering in at least one application domain, such as finance, medical, transportation, or telecommunications, and one application type, such as real-time, embedded, safety-critical, or highly

distributed systems. That mastery includes understanding how differences in domain and type manifest themselves in both the software itself and in its engineering, and includes understanding how to learn a new application domain or type.

- Master at least one KA or sub-area from the CBOK to at least the Bloom Synthesis level.

- Be able to make ethical professional decisions and practice ethical professional behavior.

- Understand the relationship between SwE and SE and be able to apply SE principles and practices in the engineering of software.

- Be an effective member of a team, including teams that are international and geographically distributed, effectively communicate both orally and in writing, and lead in one area of project development, such as project management, requirements analysis, architecture, construction, or quality assurance.

- Be able to reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, existing systems, and organizations.
- Understand and appreciate feasibility analysis, negotiation, and good communications with stakeholders in a typical software development environment, and be able to perform those tasks well; have effective work habits and be a leader.

- Be able to learn new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.

- Be able to analyze a current significant software technology, articulate its strengths and weaknesses, compare it to alternative technologies, and specify and promote improvements or extensions to that technology.

4.2 Expected Background

GSwE2009 presumes that an entering student has:

- The equivalent of an undergraduate degree in computing or an undergraduate degree in an engineering or scientific field and a minor in computing.

- The equivalent of an introductory course in software engineering.

- At least two years of practical experience in some aspect of software engineering or software development.

4.3 Core Body of Knowledge (CBOK)

4.3.1 Development of the CBOK

The primary source for developing the CBOK was the SWEBOK. Knowledge elements were also derived from SE2004, (INCOSE, 2003) and especially (Haskins, 2007). In the study and analysis of these sources, it was decided that although the SWEBOK organization and content would dominate, various changes in areas and topics were needed to support the GSwE2009 expected student outcomes and to accommodate the needs and views of academia, industry, and the computing professional societies. For example, two KAs, not in the current version of the SWEBOK, were added: Systems Engineering Fundamentals, and Ethics and Professional Conduct. In addition, some units and topics were added, rearranged or modified. These included:

- Addition of Human Computer Interface design in the Software Design KA

- Addition of an Engineering Economics unit in the Software Engineering Management KA

- Addition of a Risk Management unit in the Software Engineering Management KA

- Addition of a Verification and Validation (V&V) unit in the Software Quality KA

- Changes in the names and the unit/topic organization in three KAs: (a) Software Requirements to Requirements Engineering, (b) Software Testing to Testing and (c) Software Configuration Management to Configuration Management. These changes were made to accommodate and emphasize the role of SE in GSwE2009.

It should be noted that as of the publication date of GSwE2009, the plans for a 2010 refresh of SWEBOK call for a new KA on Professional Practice and four new education KAs: Engineering Economy Foundations, Computing Foundations, Mathematical Foundations, and Engineering Foundations. GSwE2009 has attempted to accommodate the SWEBOK refresh by including these topics in the preparation knowledge (discussed in the next section) and in the additional KAs and units in the CBOK.

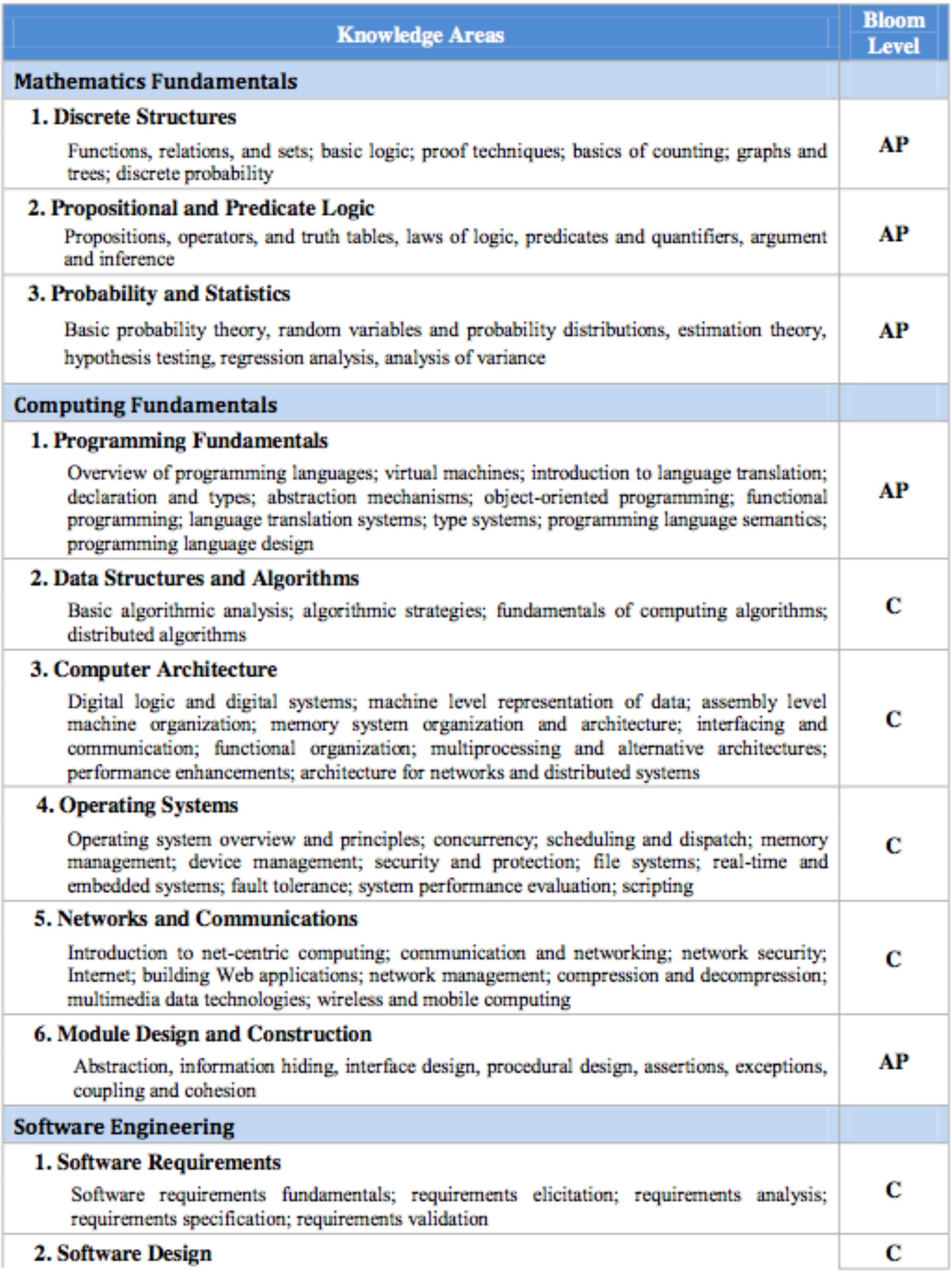
The CAT (Curriculum Author Team) has provided a recommended level to which a student should achieve each KA; these are defined in terms of Bloom‘s taxonomy. Appendix B describes Bloom‘s[1] cognitive levels and the process used to specify the student cognitive level for both the prerequisite KAs and the CBOK KAs. The following level designations are used in the tables in this section:

[1] Bloom, B.S. (Ed.), Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain, Longmans, 1956.

- Knowledge (K)
- Comprehension (C)
- Application (AP)
- Analysis (AN)

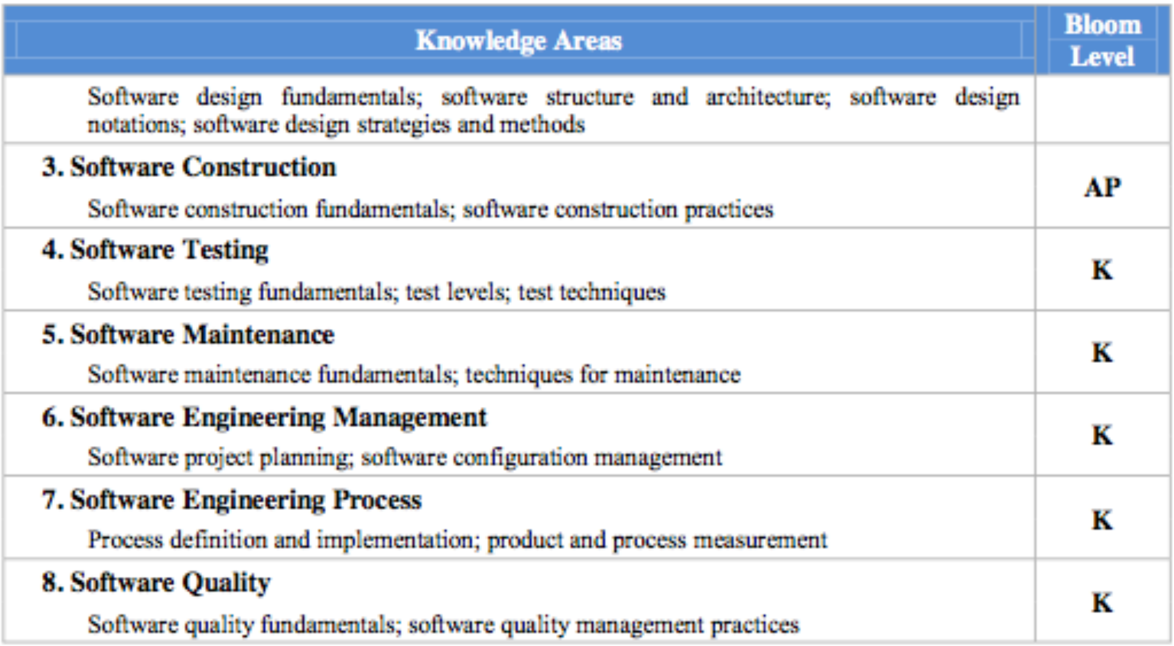
4.3.2 Preparation Knowledge

Table 1 shows the knowledge students should possess when entering a master's program, in order to be best prepared to achieve the GSwE2009 outcomes. SE2004 was the primary source for the knowledge elements. The knowledge may be acquired through undergraduate study, from software development experience, through leveling courses offered by an institution, or through some combination of these. The table is organized hierarchically into three levels, similar to the knowledge organization in the SE2004. The highest level of the hierarchy is the KA, such as Mathematical Fundamentals. Each KA is shown in blue and is broken down into smaller numbered divisions called units, which represent individual thematic modules within an area. Each unit is further subdivided into an unordered set of topics.



Knowledge Areas	Bloom Level
Mathematics Fundamentals	
1. Discrete Structures Functions, relations, and sets; basic logic; proof techniques; basics of counting; graphs and trees; discrete probability	AP
2. Propositional and Predicate Logic Propositions, operators, and truth tables, laws of logic, predicates and quantifiers, argument and inference	AP
3. Probability and Statistics Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, analysis of variance	AP
Computing Fundamentals	
1. Programming Fundamentals Overview of programming languages; virtual machines; introduction to language translation; declaration and types; abstraction mechanisms; object-oriented programming; functional programming; language translation systems; type systems; programming language semantics; programming language design	AP
2. Data Structures and Algorithms Basic algorithmic analysis; algorithmic strategies; fundamentals of computing algorithms; distributed algorithms	C
3. Computer Architecture Digital logic and digital systems; machine level representation of data; assembly level machine organization; memory system organization and architecture; interfacing and communication; functional organization; multiprocessing and alternative architectures; performance enhancements; architecture for networks and distributed systems	C
4. Operating Systems Operating system overview and principles; concurrency; scheduling and dispatch; memory management; device management; security and protection; file systems; real-time and embedded systems; fault tolerance; system performance evaluation; scripting	C
5. Networks and Communications Introduction to net-centric computing; communication and networking; network security; Internet; building Web applications; network management; compression and decompression; multimedia data technologies; wireless and mobile computing	C
6. Module Design and Construction Abstraction, information hiding, interface design, procedural design, assertions, exceptions, coupling and cohesion	AP
Software Engineering	
1. Software Requirements Software requirements fundamentals; requirements elicitation; requirements analysis; requirements specification; requirements validation	C
2. Software Design	C

Table 1.1 Preparation Knowledge for Core Body of Knowledge (part 1)



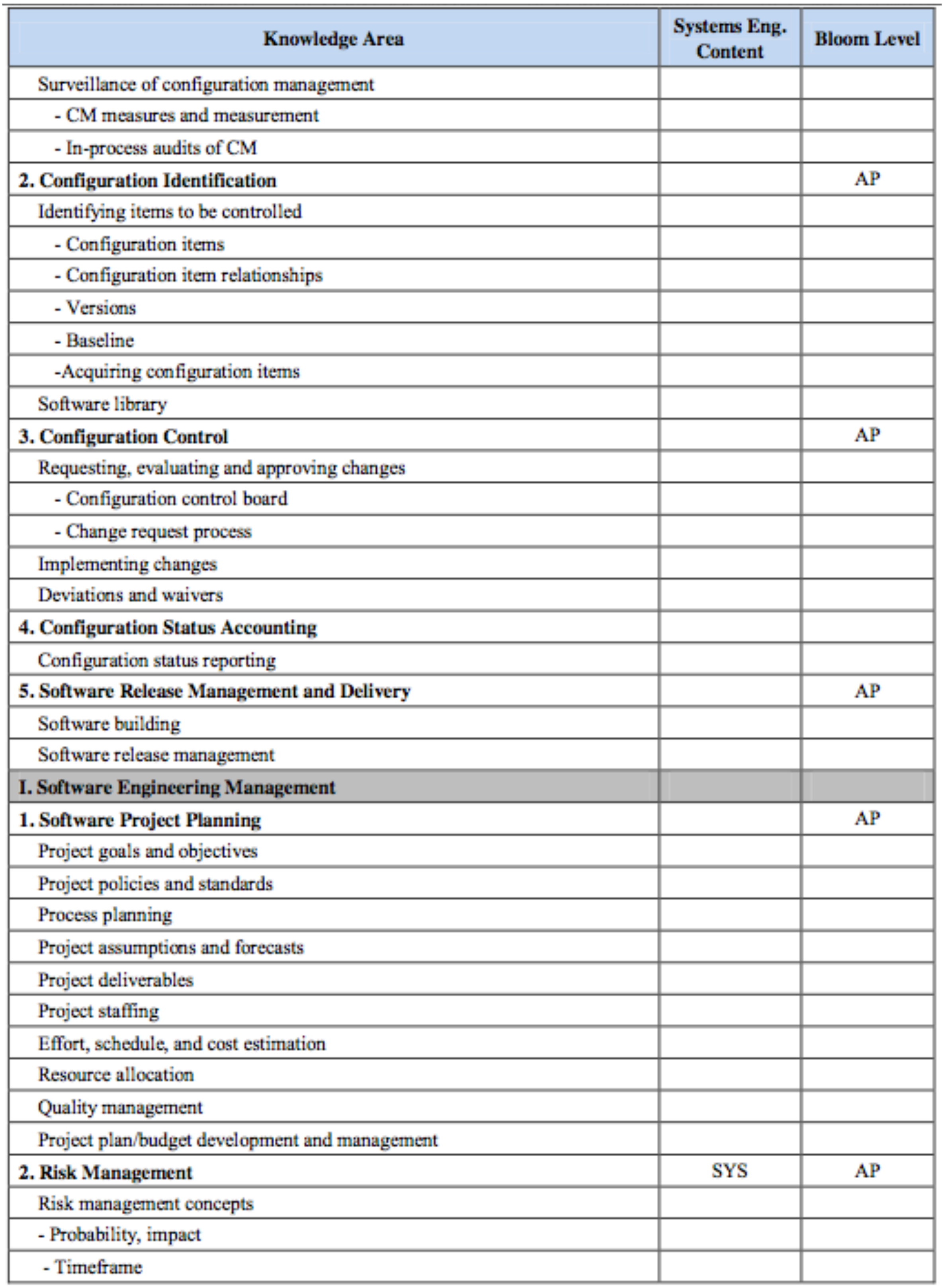
Knowledge Areas	Bloom Level
Software design fundamentals; software structure and architecture; software design notations; software design strategies and methods	
3. Software Construction Software construction fundamentals; software construction practices	AP
4. Software Testing Software testing fundamentals; test levels; test techniques	K
5. Software Maintenance Software maintenance fundamentals; techniques for maintenance	K
6. Software Engineering Management Software project planning; software configuration management	K
7. Software Engineering Process Process definition and implementation; product and process measurement	K
8. Software Quality Software quality fundamentals; software quality management practices	K

Table 1.2. Preparation Knowledge for Core Body of Knowledge (part 2)

4.3.3 CBOK Concepts and Organization

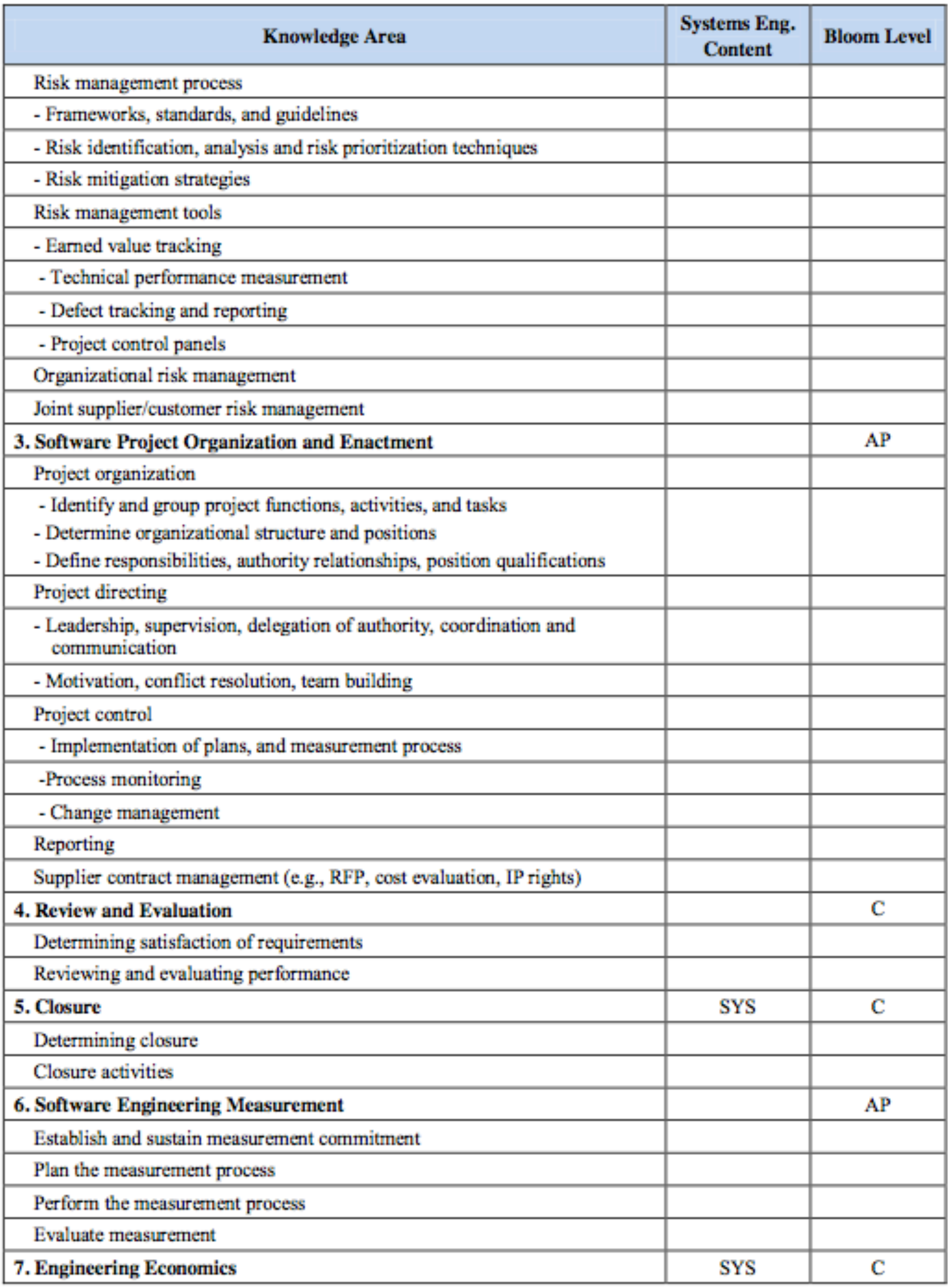
Table 2 show the table in which the GSwE2009 presents the project management related outline of the GSwE2009's CBOK. It is organized hierarchically in the same manner as Figures 51 and 52. The CBOK knowledge units and their Bloom level designations were developed in such a way that the core could be covered in the equivalent of approximately 15 credit hours or approximately 200 contact hours (using a North American academic model). The core is designed to comprise a little less than 50% of the total credit hours recommended for a master's degree. Hence, additional time and courses can be allocated to provide additional depth in the core areas (at higher Bloom levels) and to focus on a chosen application domain. An actual workload measure (such as that used in the European Commission's European Credit Transfer System[2]) could have been used, but it was felt that contact hours were sufficient for the intended level of this curriculum guidance.

[2] European Commission, Education & Training, ―European Credit Transfer and Accumulation System (ECTS)‖ website. http://ec.europa.eu/education/programmes/socrates/ects/index_en.html#1



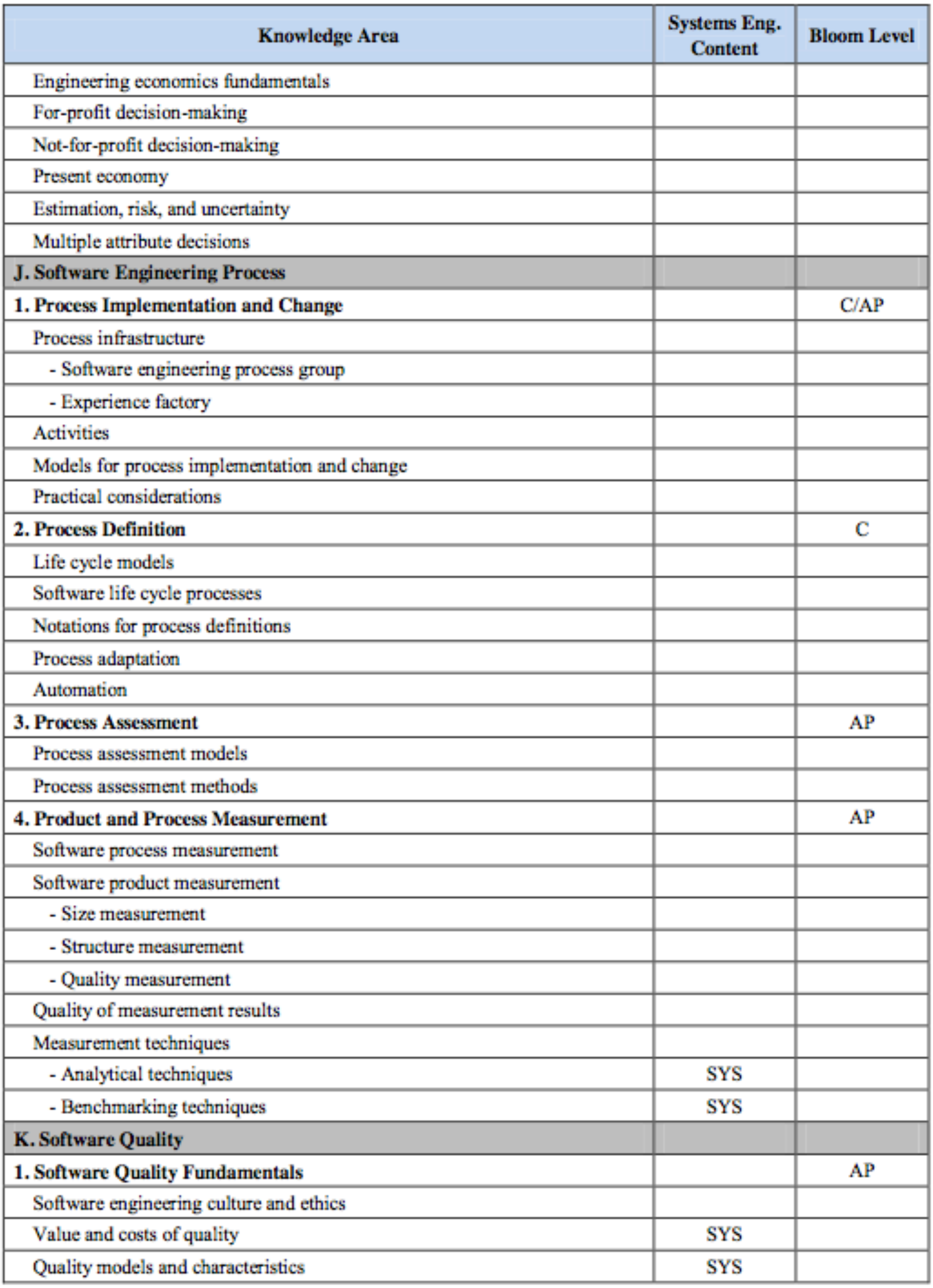
Knowledge Area	Systems Eng. Content	Bloom Level
Surveillance of configuration management		
- CM measures and measurement		
- In-process audits of CM		
2. Configuration Identification		AP
Identifying items to be controlled		
- Configuration items		
- Configuration item relationships		
- Versions		
- Baseline		
-Acquiring configuration items		
Software library		
3. Configuration Control		AP
Requesting, evaluating and approving changes		
- Configuration control board		
- Change request process		
Implementing changes		
Deviations and waivers		
4. Configuration Status Accounting		
Configuration status reporting		
5. Software Release Management and Delivery		AP
Software building		
Software release management		
I. Software Engineering Management		
1. Software Project Planning		AP
Project goals and objectives		
Project policies and standards		
Process planning		
Project assumptions and forecasts		
Project deliverables		
Project staffing		
Effort, schedule, and cost estimation		
Resource allocation		
Quality management		
Project plan/budget development and management		
2. Risk Management	SYS	AP
Risk management concepts		
- Probability, impact		
- Timeframe		

Table 2.1. Core Body of Knowledge – Software Management (part 1)



Knowledge Area	Systems Eng. Content	Bloom Level
Risk management process		
- Frameworks, standards, and guidelines		
- Risk identification, analysis and risk prioritization techniques		
- Risk mitigation strategies		
Risk management tools		
- Earned value tracking		
- Technical performance measurement		
- Defect tracking and reporting		
- Project control panels		
Organizational risk management		
Joint supplier/customer risk management		
3. Software Project Organization and Enactment		AP
Project organization		
- Identify and group project functions, activities, and tasks - Determine organizational structure and positions - Define responsibilities, authority relationships, position qualifications		
Project directing		
- Leadership, supervision, delegation of authority, coordination and communication		
- Motivation, conflict resolution, team building		
Project control		
- Implementation of plans, and measurement process		
-Process monitoring		
- Change management		
Reporting		
Supplier contract management (e.g., RFP, cost evaluation, IP rights)		
4. Review and Evaluation		C
Determining satisfaction of requirements		
Reviewing and evaluating performance		
5. Closure	SYS	C
Determining closure		
Closure activities		
6. Software Engineering Measurement		AP
Establish and sustain measurement commitment		
Plan the measurement process		
Perform the measurement process		
Evaluate measurement		
7. Engineering Economics	SYS	C

Table 2.2. Core Body of Knowledge – Software Management (part 2)



Knowledge Area	Systems Eng. Content	Bloom Level
Engineering economics fundamentals		
For-profit decision-making		
Not-for-profit decision-making		
Present economy		
Estimation, risk, and uncertainty		
Multiple attribute decisions		
J. Software Engineering Process		
1. Process Implementation and Change		C/AP
Process infrastructure		
- Software engineering process group		
- Experience factory		
Activities		
Models for process implementation and change		
Practical considerations		
2. Process Definition		C
Life cycle models		
Software life cycle processes		
Notations for process definitions		
Process adaptation		
Automation		
3. Process Assessment		AP
Process assessment models		
Process assessment methods		
4. Product and Process Measurement		AP
Software process measurement		
Software product measurement		
- Size measurement		
- Structure measurement		
- Quality measurement		
Quality of measurement results		
Measurement techniques		
- Analytical techniques	SYS	
- Benchmarking techniques	SYS	
K. Software Quality		
1. Software Quality Fundamentals		AP
Software engineering culture and ethics		
Value and costs of quality	SYS	
Quality models and characteristics	SYS	

Table 2.3. Core Body of Knowledge – Software Management (part 3)

Figure 53 depicts the percentages of the curriculum that are recommended for each core KA. These percentages were initially determined by using a quasi Wideband Delphi technique to allocate the 200 contact hours, and then the hours were converted to percentages (of the 50% core) and adjusted to ranges of approximately 1%-2%. As indicated in *Figure 53*, the percentages for each area apply only to the core, which represents approximately 50% of the curriculum. The percentages should be considered as general high-level guidance, not as precise curriculum specification.

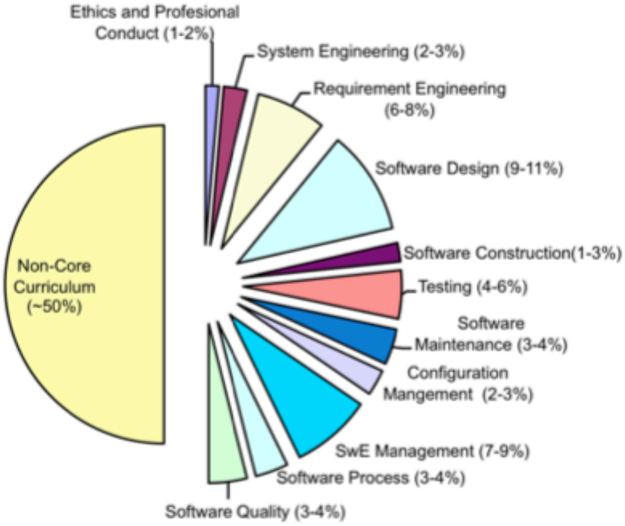


Figure 53. Percentage Devoted to Core Body of Knowledge Areas

5 SPM Knowledge Mapping

5.1 Mapping Strategy

In order to define the mapping of concepts between the SWEBOK and the PMBOK we have developed a "Junction Table" represented in *Table 3*. This table will be used to show the coverage present in the SWEBOK about the topics in the PMBOK. The PMBOK is used as the reference because is a much broader source of information about the Project Management area, and by matching what information is presented in the SWEBOK it will be spotted what is missing too. From what is missing it is intended to pick up the pieces needed for the proposals on enhancement of the way of teaching Project Management in the field of technology.

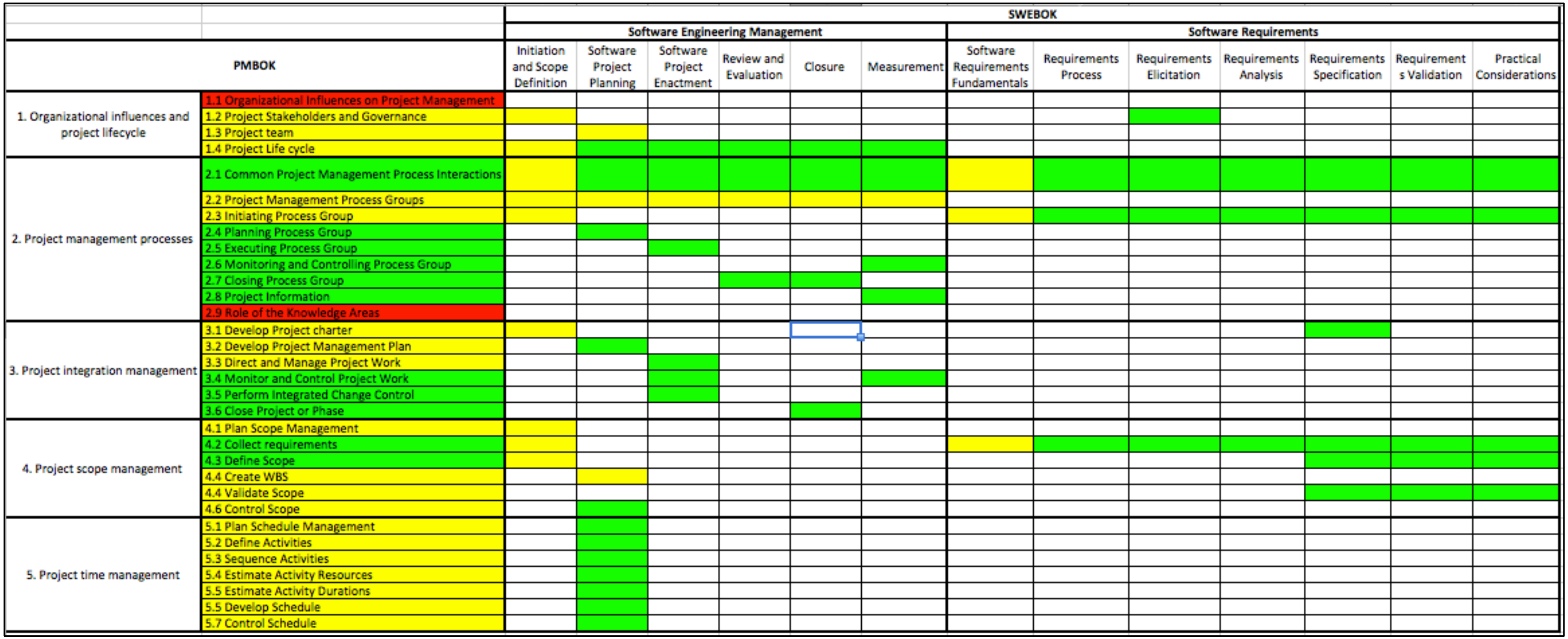
Under the two columns under *PMBOK* is shown first, all the sections of the guide and next to them the topics that are composed from. Under *SWEBOK* are shown the different knowledge areas and the topics that compose them.

The colors assigned to the topics of the SWEBOK mean: yellow if the topic contains only "Theory" about the matched part of the PMBOK and green if contains "Theory and Practice". A topic of the SWEBOK is left with no color if is irrelevant to the matched PMBOK topic.

The meanings of the colors to be assigned to the PMBOK topics are: red for "No coverage", yellow for "Incomplete Coverage" and green for "Full coverage" of a particular knowledge in the SWEBOK. The color assignation is made by comparing the content presented in the PMBOK with the content in the KAs of the SWEBOK, and then is checked if there is enough content in some part(s) of the SWEBOK to state that they both provide the same amount of information. The yellow color can be assigned due to several reasons: missing concepts considered important to understand the topic, no suggestions on practices of the topic, or just lack of information in general.

5.2 SPM Mapping Results

Below we will discuss how each of the PMBOK topics is covered with the knowledge provided by SWEBOK according to the considerations in the section **5.2 Mapping Strategy**.

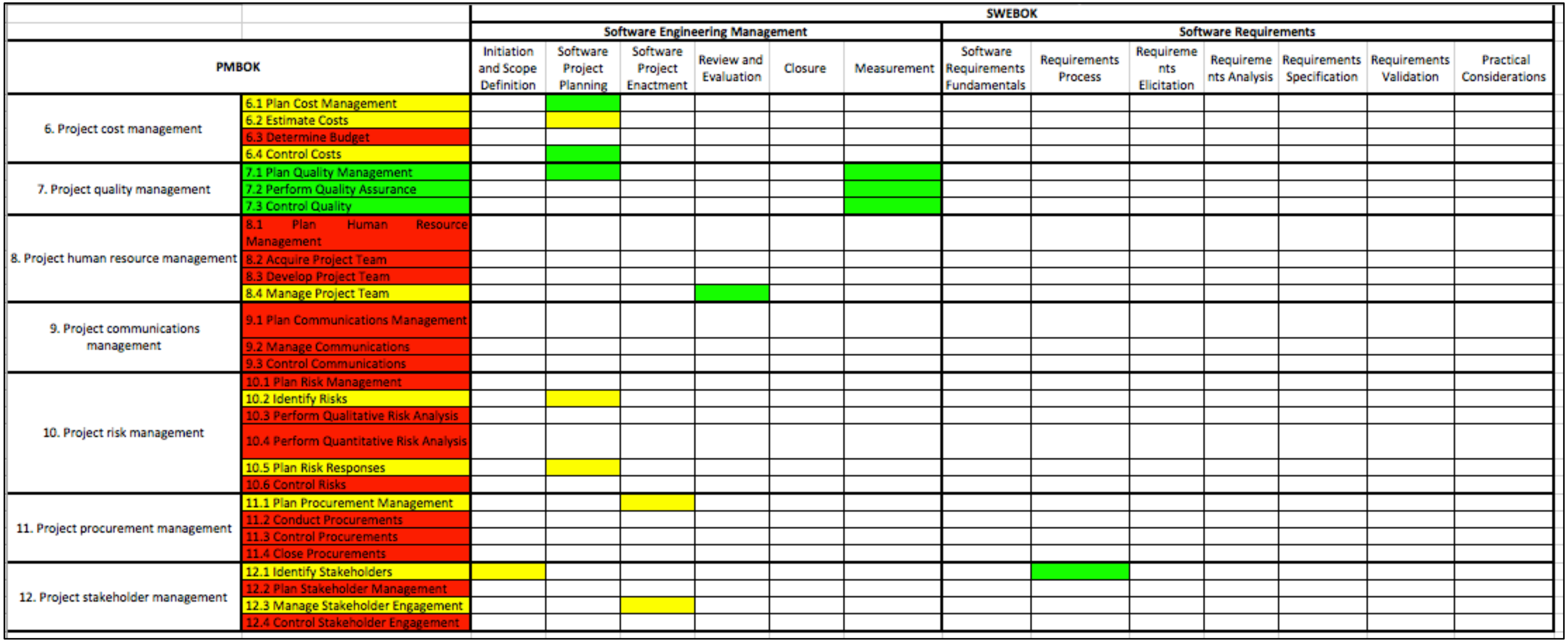


		SWEBOK												
		Software Engineering Management						Software Requirements						
PMBOK		Initiation and Scope Definition	Software Project Planning	Software Project Enactment	Review and Evaluation	Closure	Measurement	Software Requirements Fundamentals	Requirements Process	Requirements Elicitation	Requirements Analysis	Requirements Specification	Requirement s Validation	Practical Considerations
1. Organizational influences and project lifecycle	🟥 1.1 Organizational Influences on Project Management													
	🟨 1.2 Project Stakeholders and Governance	🟨								🟩				
	🟨 1.3 Project team		🟨											
	🟨 1.4 Project Life cycle	🟨	🟩	🟩	🟩	🟩	🟩							
2. Project management processes	🟩 2.1 Common Project Management Process Interactions	🟨	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩
	🟨 2.2 Project Management Process Groups	🟨	🟨	🟨	🟨	🟨	🟨							
	🟨 2.3 Initiating Process Group	🟨						🟨	🟩	🟩	🟩	🟩	🟩	🟩
	🟩 2.4 Planning Process Group		🟩											
	🟩 2.5 Executing Process Group			🟩										
	🟩 2.6 Monitoring and Controlling Process Group						🟩							
	🟩 2.7 Closing Process Group				🟩	🟩								
	🟩 2.8 Project Information						🟩							
	🟥 2.9 Role of the Knowledge Areas													
3. Project integration management	🟨 3.1 Develop Project charter	🟨										🟩		
	🟨 3.2 Develop Project Management Plan		🟩											
	🟨 3.3 Direct and Manage Project Work			🟩										
	🟩 3.4 Monitor and Control Project Work			🟩			🟩							
	🟩 3.5 Perform Integrated Change Control			🟩										
	🟩 3.6 Close Project or Phase					🟩								
4. Project scope management	🟨 4.1 Plan Scope Management	🟨												
	🟩 4.2 Collect requirements	🟨						🟨	🟩	🟩	🟩	🟩	🟩	🟩
	🟩 4.3 Define Scope	🟨										🟩	🟩	🟩
	🟨 4.4 Create WBS		🟨											
	🟨 4.4 Validate Scope											🟩	🟩	🟩
	🟨 4.6 Control Scope		🟩											
5. Project time management	🟨 5.1 Plan Schedule Management		🟩											
	🟨 5.2 Define Activities		🟩											
	🟨 5.3 Sequence Activities		🟩											
	🟨 5.4 Estimate Activity Resources		🟩											
	🟨 5.5 Estimate Activity Durations		🟩											
	🟨 5.5 Develop Schedule		🟩											
	🟨 5.7 Control Schedule		🟩											

Table 3.1. Junction Table. Part 1 – Topics 1 to 5 of PMBOK with Software Engineering Management and Software Requirements KAs of SWEBOK

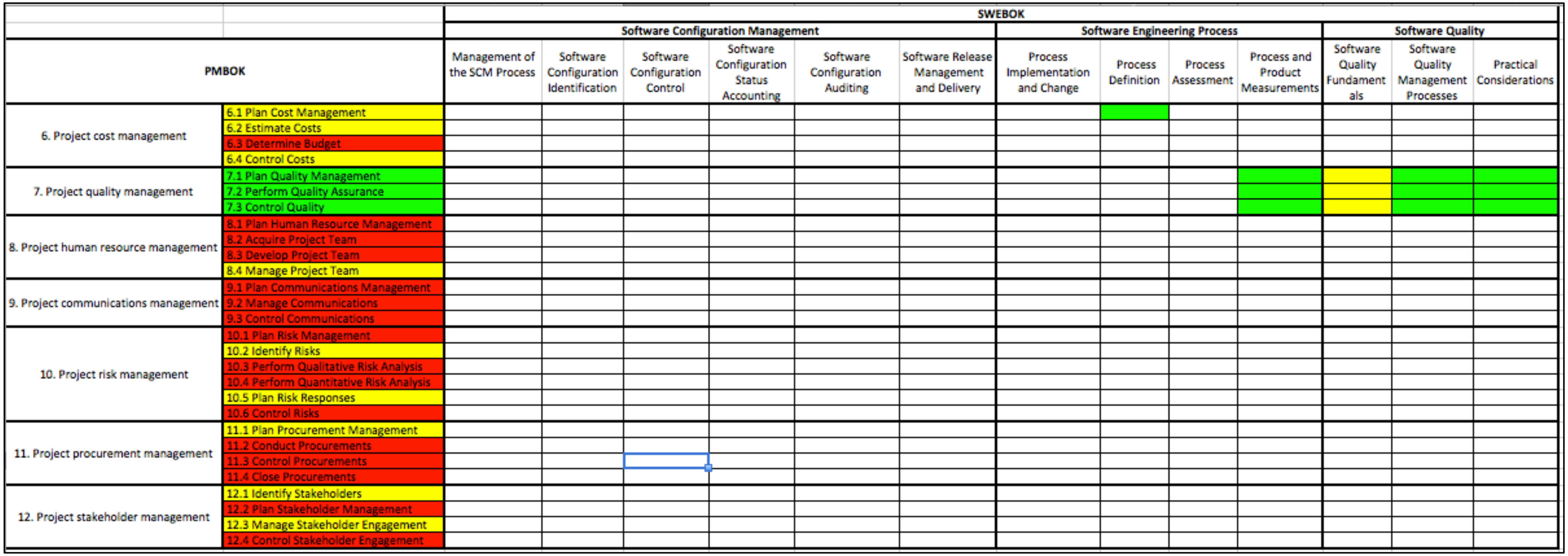
		SWEBOK												
		Software Configuration Management						Software Engineering Process				Software Quality		
PMBOK		Management of the SCM Process	Software Configuration Identification	Software Configuration Control	Software Configuration Status Accounting	Software Configuration Auditing	Software Release Management and Delivery	Process Implementation and Change	Process Definition	Process Assessment	Process and Product Measurements	Software Quality Fundamentals	Software Quality Managemen t Processes	Practical Considerations
1. Organizational influences and project lifecycle	🟥 1.1 Organizational Influences on Project Management													
	🟨 1.2 Project Stakeholders and Governance													
	🟨 1.3 Project team													
	🟨 1.4 Project Life cycle								🟨					
2. Project management processes	🟩 2.1 Common Project Management Process Interactions													
	🟨 2.2 Project Management Process Groups													
	🟨 2.3 Initiating Process Group													
	🟩 2.4 Planning Process Group	🟩	🟩					🟩	🟩					
	🟩 2.5 Executing Process Group			🟩	🟩									
	🟩 2.6 Monitoring and Controlling Process Group					🟩				🟩	🟩	🟨	🟩	🟩
	🟩 2.7 Closing Process Group									🟩	🟩		🟩	
	🟩 2.8 Project Information					🟩							🟩	
	🟥 2.9 Role of the Knowledge Areas													
3. Project integration management	🟨 3.1 Develop Project charter													
	🟨 3.2 Develop Project Management Plan	🟩	🟩						🟩					
	🟨 3.3 Direct and Manage Project Work	🟩	🟩	🟩	🟩	🟩	🟩							
	🟩 3.4 Monitor and Control Project Work			🟩	🟩	🟩							🟩	
	🟩 3.5 Perform Integrated Change Control	🟩	🟩	🟩	🟩	🟩	🟩							
	🟩 3.6 Close Project or Phase													
4. Project scope management	🟨 4.1 Plan Scope Management								🟩					
	🟩 4.2 Collect requirements													
	🟩 4.3 Define Scope													
	🟨 4.4 Create WBS													
	🟨 4.4 Validate Scope													
	🟨 4.6 Control Scope													
5. Project time management	🟨 5.1 Plan Schedule Management								🟩					
	🟨 5.2 Define Activities													
	🟨 5.3 Sequence Activities													
	🟨 5.4 Estimate Activity Resources													
	🟨 5.5 Estimate Activity Durations													
	🟨 5.5 Develop Schedule													
	🟨 5.7 Control Schedule													

Table 3.2. Junction Table. Part 2 – Topics 1 to 5 of PMBOK with Software Configuration Management, Software Engineering Process and Software Quality KAs of SWEBOK



		SWEBOK												
		Software Engineering Management						Software Requirements						
PMBOK		Initiation and Scope Definition	Software Project Planning	Software Project Enactment	Review and Evaluation	Closure	Measurement	Software Requirements Fundamentals	Requirements Process	Requirements Elicitation	Requirements Analysis	Requirements Specification	Requirements Validation	Practical Considerations
6. Project cost management	🟨 6.1 Plan Cost Management		🟩											
	🟨 6.2 Estimate Costs		🟨											
	🟥 6.3 Determine Budget													
	🟨 6.4 Control Costs		🟩											
7. Project quality management	🟩 7.1 Plan Quality Management		🟩				🟩							
	🟩 7.2 Perform Quality Assurance						🟩							
	🟩 7.3 Control Quality						🟩							
8. Project human resource management	🟥 8.1 Plan Human Resource Management													
	🟥 8.2 Acquire Project Team													
	🟥 8.3 Develop Project Team													
	🟨 8.4 Manage Project Team				🟩									
9. Project communications management	🟥 9.1 Plan Communications Management													
	🟥 9.2 Manage Communications													
	🟥 9.3 Control Communications													
10. Project risk management	🟥 10.1 Plan Risk Management													
	🟨 10.2 Identify Risks		🟨											
	🟥 10.3 Perform Qualitative Risk Analysis													
	🟥 10.4 Perform Quantitative Risk Analysis													
	🟨 10.5 Plan Risk Responses		🟨											
	🟥 10.6 Control Risks													
11. Project procurement management	🟨 11.1 Plan Procurement Management			🟨										
	🟥 11.2 Conduct Procurements													
	🟥 11.3 Control Procurements													
	🟥 11.4 Close Procurements													
12. Project stakeholder management	🟨 12.1 Identify Stakeholders	🟨							🟩					
	🟥 12.2 Plan Stakeholder Management													
	🟨 12.3 Manage Stakeholder Engagement			🟨										
	🟥 12.4 Control Stakeholder Engagement													

Table 3.3. Junction Table. Part 3 – Topics 6 to 12 of PMBOK with Software Engineering Management and Software Requirements KAs of SWEBOK



		SWEBOK												
		Software Configuration Management						Software Engineering Process				Software Quality		
PMBOK		Management of the SCM Process	Software Configuration Identification	Software Configuration Control	Software Configuration Status Accounting	Software Configuration Auditing	Software Release Management and Delivery	Process Implementation and Change	Process Definition	Process Assessment	Process and Product Measurements	Software Quality Fundamentals	Software Quality Management Processes	Practical Considerations
6. Project cost management	6.1 Plan Cost Management								🟩					
	6.2 Estimate Costs													
	6.3 Determine Budget													
	6.4 Control Costs													
7. Project quality management	7.1 Plan Quality Management										🟩	🟨	🟩	🟩
	7.2 Perform Quality Assurance										🟩	🟨	🟩	🟩
	7.3 Control Quality										🟩	🟨	🟩	🟩
8. Project human resource management	8.1 Plan Human Resource Management													
	8.2 Acquire Project Team													
	8.3 Develop Project Team													
	8.4 Manage Project Team													
9. Project communications management	9.1 Plan Communications Management													
	9.2 Manage Communications													
	9.3 Control Communications													
10. Project risk management	10.1 Plan Risk Management													
	10.2 Identify Risks													
	10.3 Perform Qualitative Risk Analysis													
	10.4 Perform Quantitative Risk Analysis													
	10.5 Plan Risk Responses													
	10.6 Control Risks													
11. Project procurement management	11.1 Plan Procurement Management													
	11.2 Conduct Procurements													
	11.3 Control Procurements													
	11.4 Close Procurements													
12. Project stakeholder management	12.1 Identify Stakeholders													
	12.2 Plan Stakeholder Management													
	12.3 Manage Stakeholder Engagement													
	12.4 Control Stakeholder Engagement													

Table 3.4. Junction Table. Part 4 – Topics 6 to 12 of PMBOK with Software Configuration Management, Software Engineering Process and Software Quality KAs of SWEBOK

5.2.1 Organizational Influences and Project Lifecycle

5.2.1.1 Organizational Influences on Project Management

There is no information in the SWEBOK about what factors could influence a project from an organizational perspective (red color).

5.2.1.2 Project Stakeholders and Governance

The PMBOK gives an introduction of the considerations that should be taken into account for *Project Stakeholders and Governance*.

About these considerations, SWEBOK mentions "Theory" (yellow color) about the interaction with project stakeholders in the topic *Initiation and Scope Definition* of the *Software Project Management* KA. This topic makes reference to the *Software Requirements* KA, where extended information about stakeholder management is detailed in the *Requirements Elicitation* topic, which mentions "Theory and Practice" (green color) to be used for the process of getting requirements.

On the other hand SWEBOK is missing things like; what kind of stakeholders could be expected from an organizational perspective and governance concepts like the need of decision-making models. Given that SWEBOK misses this information, *Project Stakeholders and Governance* is marked with "Incomplete coverage" (yellow color).

5.2.1.3 Project Team

PMBOK introduces some considerations about the *Project Team* topic. An equivalent to this can be found in the *Software Project Planning* topic of the *Software Project Management* KA of the SWEBOK, where the concept of resources is presented as assets to be allocated, estimated and managed. One of these resources is people, so we could say that somehow *Project Team* topic is covered in the theory from the SWEBOK (yellow color).

However, the SWEBOK doesn't give any introduction about what roles a *Project Team* usually have and what factors to be aware of when building those teams. For this reason *Project Team* is marked with "Incomplete Coverage" (yellow color).

5.2.1.4 Project Life Cycle

About *Project Life Cycle*, PMBOK gives introduction of the phases in a project lifecycle and the relationships that can be established between those phases.

In the SWEBOK an equivalent of each phase is presented in the topics of the *Software Project Management* KA. Those equivalents topics are: *Initiation and*

Scope Definition, Software Project Planning, Software Project Enactment, Review and Evaluation, Closure and *Measurement*. The information provided in these topics about theory and practice (green color) during the phases of a *Project Life Cycle*.

Even though the phases are detailed, the general concept and importance of defining a *Project Life Cycle* is not included in the SWEBOK, which is why is marked with "Incomplete Coverage" (yellow color).

5.2.2 Project Management Processes

5.2.2.1 *Common Project Management Process Interactions*

As discussed in chapter 3 (PMBOK Description), The *Common Project Management Process Interactions* topic of the PMBOK shows an overall view of the interaction between the different process groups (initiating, planning, executing, monitoring and closing) in a project.

The SWEBOK talks about the phases of a project and the activities to be carried out in each phase. Those activities in the end belong to a process group from the PMBOK and the SWEBOK presents them in the topics of the *Software Project Management* KA. The topics where the processes/activities are mentioned are: *Initiation and Scope Definition, Software Project Planning, Software Project Enactment, Review and Evaluation, Closure* and *Measurement.* The *Initiation and Scope Definition* topic presents mostly "Theory" (yellow color), but references the *Software Requirements* KA that has both "Theory and Practice" (green color), except for the *Requirements Fundamentals* topic. The rest of the topics of the *Software Engineering Management* KA provide "Theory and Practice" (green color) related to the project management processes.

SWEBOK also talks about the interaction between the activities/processes in the KAs. So, in general the different parts of the *Common Project Management Process Interactions* topic are covered by SWEBOK, that's why it has been marked with "Complete Coverage" (green color).

5.2.2.2 *Project Management Process Groups*

PMBOK talks about the *Project Management Process Groups* (initiating, planning, executing, monitoring and closing) and their difference with the phases of a project lifecycle.

The "Theory" (yellow color) about the processes carried out in each process group are mentioned in the following topics of *Software Engineering Management* KA of the SWEBOK: *Initiation and Scope Definition, Software Project Planning, Software Project Enactment, Review and Evaluation, Closure* and *Measurement.*

Given that the importance and difference between process groups and phases of a project are not detailed in the SWEBOK, the *Project Management Process Groups* topic is marked with "Incomplete Coverage" (yellow color).

5.2.2.3 Initiating Process Group

In the *Initiating Process Group* topic, the PMBOK gives an overview on initiation process activities.

Requirements as part of the initiating process of a software project, is widely covered by the SWEBOK. The *Initiation and Scope Definition* topic presents mostly "Theory" (yellow color), but references the *Software Requirements* KA that has both "Theory and Practice" (green color), except for the *Requirements Fundamentals* topic

On the other hand, the need of taking into account several factors at an organizational level is not mentioned, so that's why the *Junction Table* states that there is only "Incomplete Coverage" (yellow color) by the SWEBOK on this.

5.2.2.4 Planning Process Group

In the *Planning Process Group* topic the PMBOK gives an overview of the activities and importance of planning.

"Theory and practice" (green color) of planning for cost and time are presented in the *Software Project Planning* topic of the *Software Project Management* KA of the SWEBOK.

Given that the PMBOK gives only and introduction in *Planning Process Group* topic and that SWEBOK gives both theory and practice, it is considered that there is "Complete Coverage" (green color) of this in the SWEBOK.

5.2.2.5 Executing Process Group

The PMBOK introduces the activities and processes in the *Executing Process Group.*

The SWEBOK covers all necessary introduction about the topic in the *Software Project Enactment* KA. It has "Theory and Practice" (green color) for the "executing" phase a software project. The SWEBOK also provides extensive information about theory and practice (green color) on change management in the *Software Configuration Management* KA.

All the information provided in the *Executing Process Group* topic of the PMBOK is considered to be also provided in the SWEBOK, so it is marked with "Complete Coverage" (green color).

5.2.2.6 Monitoring and Controlling Process Group

The *Monitoring and Controlling Process Group* topic of the PMBOK introduces the activities for tracking and review of a project's performance.

This topic is widely covered by several parts of the SWEBOK. There is "Theory and Practice" (green color) all the following parts of the guide: *Measurement* topic of the *Software Project Management* KA, *Software Configuration Auditing* topic of the *Software Configuration Management* KA and the whole *Software Quality* KA except for the *Software Quality Fundamentals* that only provides "Theory" (yellow color).

If all the information in previously mentioned KAs of the SWEBOK's is summed up, everything related to the monitoring and controlling process of a software project is presented through concepts and practical techniques to be used. For this reason *Monitoring and Controlling Process Group* is marked with "Complete Coverage" (green color).

5.2.2.7 Closing Process Group

The PMBOK in its topic *Closing Process Group* introduces activities to conclude a project.

The SWEBOK details with theory and practical considerations (green color) these activities in the following sections: *Review and Evaluation* and *Closure* topics of the *Software Project Management* KA, also *Process and Product Measurements* at *Software Engineering Process* KA present concepts and practices useful for the closure of a project. Finally *Software Quality Management Processes* topic of the *Software Quality* KA provides information about all the closure activities related to quality in a software project such as storing data for future estimations and general usage.

Given that SWEBOK covers and exceeds what is presented in the PMBOK, *Closing Process Group* topic is marked with "Complete Coverage" (green color).

5.2.2.8 Project Information

PMBOK gives an overview of what to take into account when handling *Project Information*; why is important to gather such information and how this could be performed.

SWEBOK gives overview on concepts and practical considerations (green color) in the *Closure* topic at *Software Project Management* KA and in the *Software Quality Management Processes* topic of the *Software Quality* KA.

Given that SWEBOK covers and exceeds what is presented in the PMBOK, *Project Information* topic is marked with "Complete Coverage" (green color).

5.2.2.9 Role of the Knowledge Areas

The content of the *Role of the Knowledge Areas* topic is not applicable for comparison. The reason for this is that in this section of the PMBOK there is only the list of processes defined in the guide and how are distributed among the different Knowledge Areas or Process Groups (also defined in the guide).

Given that is considered not applicable, *Role of the Knowledge Areas* topic is marked with "No coverage" (red color).

5.2.3 Project Integration Management

5.2.3.1 Develop Project charter

In the *Develop Project Charter* the PMBOK explains that a project charter is a document in that "formally authorizes the existence of a project and provides the project manager with the authority to apply organizational resources to project activities". It also gives input and tools to be used to produce the project charter.

The SWEBOK talks about theory (yellow color) useful for the project charter in the *Initiation and Scope Definition* topic of the *Software Project Management* KA. Also the *Requirements Specification* topic of the *Software Requirements* KA provides theory and practices (green color) that could help justify the existence of the project.

On the other hand, the SWEBOK misses information like inputs, tools and techniques to be used for producing a complete and proper project chart. An example of the missing information is the importance of a "project existence exposure" section.

Because of the lack of information, *Develop Project Charter* topic is marked with "Incomplete Coverage" (yellow color).

5.2.3.2 Develop Project Management Plan

In *Develop Project Management Plan* the PMBOK states that this plan is "the document that describes how the project will be executed, monitored, and controlled". It also gives the inputs and tools and techniques to produce the plan.

The SWEBOK gives theory and practice (green color) of the management activities in the *Software Project Planning* of the *Software Project Management* KA. Also, practical considerations (green color) on the handling of change request to any relevant resource is mentioned in the *Management of SCM Process* topic and *Software Configuration Identification* topic of the *Software Configuration Management* KA.

However, there is never a direct mention in the SWEBOK of how to prepare a management plan or its importance. This is why *Develop Project Management Plant* topic is marked with “Incomplete Coverage” (yellow color).

5.2.3.3 Direct and Manage Project Work

The *Direct and Manage Project Work* is introduced in the PMBOK as “the process of leading and performing the work defined in the project management plan and implementing approved changes to achieve the project’s objectives”. It also gives the inputs, outputs and tools used to direct and manage a project.

The SWEBOK explains in detail theory and practice (green color) about change management during project enactment. This information is provided throughout all the topics in the *Software Configuration Management* KA.

However, SWEBOK misses introductory information about how to effectively direct and manage a project. This information includes: kind of actions to be taken (preventive, reactive and repair), and how to manage sellers and suppliers. Due to the missing information, the *Direct and Manage Project Work* topic is marked with “Incomplete Coverage” (yellow color).

5.2.3.4 Monitor and Control Project Work

PMBOK explains the concept of *Monitor and Control Project Work* saying that “is the process of tracking, reviewing, and reporting the progress to meet the performance objectives defined in the project management plan”. It also gives the input, outputs and tools used to execute this.

In the SWEBOK there is plenty of theory and practice (green color) on *Monitor and Control Project Work* that can be found in the following topics: *Software Project Enactment* and *Measurement* from *Software Engineering Management* KA, *Software Configuration Control, Software Configuration Status Accounting* and *Software Configuration Auditing* from *Software Configuration Management* KA, and *Software Quality Management* of *Software Quality* KA.

Because of the wide coverage in the SWEBOK, this *Monitor and Control Project Work* topic is marked with “Complete Coverage” (green color).

5.2.3.5 Perform Integrated Change Control

The PMBOK states that *Perform Integrate Change Control* is “is the process of reviewing all change requests”. It also gives the inputs, outputs and tools used to perform such control.

The SWEBOK covers widely the topic of change control, giving concepts and practical considerations (green color) in the *Software Project Enactment* topic of the *Software Project Management* KA, and going much more deeply in each one of the topics of the *Software Configuration Management* KA.

The *Perform Integrated Change Control* topic is considered to be completely covered, which is why is marked with the green color.

5.2.3.6 Close Project or Phase

About the *Close Project or Phase* topic the PMBOK talks about "the process of finalizing all activities across all of the Project Management Process Groups to formally complete the project or phase". It also gives information about input, outputs and tools to be used in the process.

The closure of a project is covered with theory and practice information (green color) in the SWEBOK in the *Closure* topic of the *Software Engineering Management* KA. About the closure of a phase in the project, it has to be taken into account that all the topics in the *Software Engineering Management* KA are equivalents of the phases in a project. Now, for every phase/topic in the mentioned KA there is information about when this is considered finished or activities to get to a closure.

The *Close Project or Phase* topic of the PMBOK is considered to have "Complete Coverage" (green color) in the SWEBOK.

5.2.4 Project Scope Management

5.2.4.1 Plan Scope Management

PMBOK defines *Project Scope Management* as "the process of creating a scope management plan that documents how the project scope will be defined, validated, and controlled". It also gives the input, outputs and tools to be used in the creation of the plan.

Theory (yellow color) useful for planning scope management is found in the *Initiation and Scope Definition* topic of the *Software Engineering Management* KA.

However, the SWEBOK misses information like: concepts and tools specific to plan scope management and what the plan should contain. Because of this the *Plan Scope Management* topic is marked with "Incomplete Coverage" (yellow color).

5.2.4.2 Collect Requirements

PMBOK presents *Collect Requirements* topic by saying that "is the process of determining, documenting, and managing stakeholder needs and requirements to meet project objectives". It also gives input, outputs and tools to be used for this process.

The SWEBOK presents theory (yellow color) on *Collect Requirements* topic in the *Initiation and Scope Definition* topic of the *Software Engineering Management* KA, and it covers in depth everything related to requirements including theory and practice (green color) for software projects in all the topics in the *Software Requirements* KA, except for the *Requirements Fundamentals* topic, which provides only theory "yellow color".

Given the completeness of the information about *Collect Requirements* in the SWEBOK, this is marked with "Complete Coverage" (green color).

5.2.4.3 Define Scope

According to the PMBOK *Define Scope* is "is the process of developing a detailed description of the project and product". The guide also gives input, outputs and tools to be used in this process.

SWEBOK covers the definition of the scope of a software based on requirements in a wide manner. This is done with theory (yellow color) in the *Initiation and Scope Definition* topic of the *Software Engineering Management* KA, and with theory and practice (green color) in the *Software Specification, Requirements Validation* and *Practical Considerations* topics of the *Software Requirements* KA.

The *Define Scope* is considered to have "Complete Coverage" in the SWEBOK, which is why is marked with green color.

5.2.4.4 Create WBS

The PMBOK says about *Create WBS* that "is the process of subdividing project deliverables and project work into smaller, more manageable components". It also gives input, output and tools to be used for the process.

In the SWEBOK there is theory (yellow color) on what a WBS is made for. This is said in the *Software Project Planning* topic of the *Software Project Management* KA.

But the SWEBOK misses information on inputs, tools and output for building a proper WBS. This is reason why *Create WBS* topic is marked with "Incomplete Coverage" (yellow color).

5.2.4.5 Validate Scope

PMBOK defines *Validate Scope* as "the process of formalizing acceptance of the completed project deliverables". It also gives input, output and tools to be used in the process.

SWEBOK presents concepts and practice (green color) information that is useful to perform validation of the scope of a software project. Such information can be found in the *Requirements Specification, Requirements Validation* and *Practical Considerations* topics of the *Software Requirements* KA.

However, in the SWEBOK there is no information on the process for deliverable determination and the general concepts and suggestion of tools for this process. Because of this *Validate Scope* is marked with "Incomplete Coverage" (yellow color).

5.2.4.6 Control Scope

The PMBOK states that *Control Scope* "is the process of monitoring the status of the project and product scope and managing changes to the scope baseline". It also gives input, output and tools to be used in this process.

Theory and practice (green color) about risk management is given in the *Software Project Planning* topic in the *Software Engineering Management* KA. Change is a risk so it could be said that change in the schedule is partially covered too.

Because there are no specific change control considerations for the scope, *Control Scope* is considered to have "Incomplete Coverage" (yellow color) in the SWEBOK.

5.2.5 Project Time Management

5.2.5.1 Plan Schedule Management

PMBOK defines *Plan Schedule Management* as "the process of establishing the policies, procedures, and documentation for planning, developing, managing, executing, and controlling the project schedule". It also provides input, output and tools to be used for the process.

SWEBOK gives a clue of what information is useful for the schedule management plan. The information consists of theory and practices (green color) in the *Software Project Planning* topic of the *Software Project Management* KA.

The SWEBOK misses information like: concepts and tools specific to plan schedule management and what the plan should contain. Because of this the *Plan Scope Management* topic is marked with "Incomplete Coverage" (yellow color).

5.2.5.2 Define Activities

The PMBOK says that *Define Activities* is "the process of identifying and documenting the specific actions to be performed to produce the project deliverables". It also provides information on input, output and tools to be used in the process.

SWEBOK introduces the need to define activities definition in order to be used for estimation and overall planning, and also suggests practical considerations (green color) like the use of Work Breakdown Structure. This information is given in the *Software Project Planning* topic of the *Software Engineering Management* KA.

However, the SWEBOK lacks of a proper introduction on the *Define Activities* topic, there is very few suggestions on tools to be used and it doesn't says which inputs affect the scheduling process. Because of this the topic is marked with "Incomplete Coverage" (yellow color).

5.2.5.3 Sequence Activities

In the PMBOK is presented that *Sequence Activities* is "the process of identifying and documenting relationships among the project activities". It also gives input, output and tools to be used in the process.

The SWEBOK mentions concepts and suggests practical considerations (green color) like "task dependencies" which is a result of analysis on relationships between activities. This information is given in the *Software Project Planning* topic of the *Software Engineering Management* KA.

However, in the information given by the SWEBOK there is no proper introduction on the *Sequence Activities* topic and also no suggestions about tools to be used. There is also no information about concepts such as Precedence Diagramming Method, Dependency Determination, among others. Because of the previously mentioned, this topic is marked with "Incomplete Coverage" (yellow color).

5.2.5.4 Estimate Activity Resources

The PMBOK defines *Estimate Activity Resources* as "The process of estimating the type and quantities of material, human resources, equipment, or supplies required to perform each activity". It also gives information about input, output and tools to be used in the process.

Theory and practical (green color) information about resource allocation is mentioned in the *Software Project Planning* topic of the *Software Engineering Management* KA in the SWEBOK.

The SWEBOK doesn't provide a wide introduction on the *Estimate Activity Resources* topic, and there are no suggestions about tools to be used such as "Bottom Up Estimation". Because of this the topic is marked with "Incomplete Coverage".

5.2.5.5 Estimate Activity Durations

The definition of the PMBOK about *Estimate Activity Durations* is "The process of estimating the number of work periods needed to complete individual activities with estimated resources". The guide also gives input, output and tools to be used in the process.

Theory and practical considerations (green color) to estimate tasks like the use of PERT charts, is found in the *Software Project Planning* topic in the *Software Engineering Management* KA of the SWEBOK.

However, the SWEBOK doesn't provide a wide introduction on the *Estimate Activity Durations* topic, and there aren't enough suggestions about tools to be used such as "Three Point Estimating". Because of this the topic is marked with "Incomplete Coverage".

5.2.5.6 Develop Schedule

PMBOK says about *Develop Schedule* that is "the process of analyzing activity sequences, durations, resource requirements, and schedule constraints to create the project schedule model". It also provides input, output and tools to be used in the process.

The content of the *Software Project Planning* topic of the *Software Engineering Management* KA of the SWEBOK exposes theory and practices (green color) like the use WBS or the PERT chart for the development of a schedule.

However, the SWEBOK doesn't provide a wide set of concepts about the *Develop Schedule* topic, and there aren't enough suggestions about tools to be used in the process. Because of this the topic is marked with "Incomplete Coverage".

5.2.5.7 Control Schedule

About *Control Schedule* in the PMBOK is found that is "the process of monitoring the status of project activities to update project progress and manage changes to the schedule baseline to achieve the plan". The guide also provides input, output and tools to be used in the process.

Theory and practice (green color) about risk management is given in the *Software Project Planning* topic in the *Software Engineering Management* KA. Change is a risk so it could be said that change in the schedule is partially covered too.

Because there are no specific change control considerations for the schedule, *Control Schedule* is considered to have "Incomplete Coverage" (yellow color) in the SWEBOK.

5.2.6 Project Cost Management

5.2.6.1 Control Cost Management

The PMBOK says about *Control Cost Management* that "is the process that establishes the policies, procedures, and documentation for planning, managing, expending, and controlling project costs". It also gives input, output and practical considerations to be used in the process.

SWEBOK gives information useful for the cost management. The information consists of theory and practices (green color) in the *Software Project Planning* topic of the *Software Project Management* KA.

The SWEBOK misses information like: concepts and tools specific to plan schedule management and what the plan should contain. Because of this the *Control Cost Management* topic is marked with "Incomplete Coverage" (yellow color).

5.2.6.2 Estimate Costs

PMBOK states that *Estimate Costs* is "the process of developing an approximation of the monetary resources needed to complete project activities". It also provides input, output and tools to be used in the process.

In the information provided by the SWEBOK there is only theory (yellow) that says that the estimation of costs should be done, but no practical considerations are given.

The *Estimate Costs* topic is marked with "Incomplete Coverage" because the SWEBOK misses information such as the factors that influence estimation.

5.2.6.3 Determine Budget

The PMBOK says about *Determine Budget* that "is the process of aggregating the estimated costs of individual activities or work packages to establish an authorized cost baseline". In the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.6.4 Control Costs

In the PMBOK is said that *Control Costs* is "the process of monitoring the status of the project to update the project costs and managing changes to the cost baseline". It also provides input, output and tools to be used in the process.

Theory and practice (green color) about risk management is given in the *Software Project Planning* topic in the *Software Engineering Management* KA. Change is a risk so it could be said that change in the costs is partially covered too.

Because there are no specific change control considerations for the schedule, *Control Costs* is considered to have "Incomplete Coverage" (yellow color) in the SWEBOK.

5.2.7 Project Quality Management

5.2.7.1 Plan Quality Management

About *Plan Quality Management* topic, the PMBOK states that "is the process of identifying quality requirements and/or standards for the project and its deliverables, and documenting how the project will demonstrate compliance with relevant quality requirements". It also gives input, output and tools to be used in the process.

Theory and practical considerations (green color) are given in the SWEBOK in the following topics: *Measurement* from *Software Engineering Management* KA, *Process and Product Measurements* from *Software Engineering Processes* KA and throughout all the topics in the *Software Quality* KA except for the *Software Quality Fundamentals* topic, which gives only theory (yellow color).

Given that for all the information in the PMBOK about *Plan Quality Management* can be also found in the SWEBOK, the topic is marked with "Complete Coverage" (green color).

5.2.7.2 Perform Quality Assurance

According to the PMBOK *Perform Quality Assurance* is "the process of auditing the quality requirements and the results from quality control measurements to ensure that appropriate quality standards and operational definitions are used". The guide also gives input, output and tools to be used in the process.

Theory and practical considerations (green color) like the use of reviews and inspections, are given in the SWEBOK in the following topics: *Measurement* from *Software Engineering Management* KA, *Process and Product Measurements* from *Software Engineering Processes* KA and throughout all the topics in the *Software Quality* KA except for the *Software Quality Fundamentals* topic, which gives only theory (yellow color).

Given that for all the information in the PMBOK about *Perform Quality Assurance* can be also found in the SWEBOK, the topic is marked with "Complete Coverage" (green color).

5.2.7.3 Control Quality

Control Quality is defined in the PMBOK as "the process of monitoring and recording results of executing the quality activities to assess performance and recommend necessary changes". It also provides input, output and tools to be used in the process.

Theory and practical considerations (green color) like the use of reviews and inspections, are given in the SWEBOK in the following topics: *Measurement* from *Software Engineering Management* KA, *Process and Product Measurements* from *Software Engineering Processes* KA and throughout all the topics in the *Software Quality* KA except for the *Software Quality Fundamentals* topic, which gives only theory (yellow color).

Given that for all the information in the PMBOK about *Control Quality* can be also found in the SWEBOK, the topic is marked with "Complete Coverage" (green color).

5.2.8 Project Human Resource Management

5.2.8.1 Plan Human Resource Management

The PMBOK states that *Plan Human Resource Management* "is the process of identifying and documenting project roles, responsibilities, required skills, reporting relationships, and creating a staffing management plan". In the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.8.2 Acquire Project Team

According to the PMBOK *Acquire Project Team* "is the process of confirming human resource availability and obtaining the team necessary to complete project activities". In the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.8.3 Develop Project Team

The PMBOK defines *Develop Project Team* as "the process of improving competencies, team member interaction, and overall team environment to enhance project performance". However in the SWEBOK there is no information about techniques for improvement and motivation of team members. So the topic is marked with "No Coverage" (red color).

5.2.8.4 Manage Project Team

Manage Project Team is defined in the PMBOK as “the process of tracking team member performance, providing feedback, resolving issues, and managing team changes to optimize project performance”. The guide also provides information about input, output and tools to be used in the process.

Some theory and practical information that is useful for the management of a project team is provided in the SWEBOK. This is provided in the *Review and Evaluation* topic of the *Software Engineering Management* KA.

However the SWEBOK doesn’t provide suggestions on factors influencing the development and management of teams. Because of this *Manage Project Team* is marked with “Incomplete Coverage” (yellow color).

5.2.9 Project Communications Management

5.2.9.1 Plan Communications Management

According to the PMBOK *Plan Communications Management* “is the process of developing an appropriate approach and plan for project communications based on stakeholder’s information needs and requirements, and available organizational assets”. In the SWEBOK there is no information that matches this specific topic and that’s why is marked with “No Coverage” (red color).

5.2.9.2 Manage Communications

The PMBOK states that *Manage Communications* “is the process of creating, collecting, distributing, storing, retrieving, and the ultimate disposition of project information in accordance to the communications management plan”. In the SWEBOK there is no information that matches this specific topic and that’s why is marked with “No Coverage” (red color).

5.2.9.3 Control Communications

The PMBOK states that *Control Communications* “is the process of monitoring and controlling communications throughout the entire project life cycle to ensure the information needs of the project stakeholders are met”. In the SWEBOK there is no information that matches this specific topic and that’s why is marked with “No Coverage” (red color).

5.2.10 Project Risk Management

5.2.10.1 Plan Risk Management

According to the PMBOK *Plan Risk Management* "is the process of defining how to conduct risk management activities for a project". In the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.10.2 Identify Risks

PMBOK defines *Identify Risks* as "the process of determining which risks may affect the project and documenting their characteristics". The guide also provides input, output and tools to be used in the process.

In the SWEBOK theory (yellow color) about *Identify Risks* topic is found in the *Software Project Planning* topic of the *Software Engineering Management* KA.

However, in the SWEBOK there is no information about tools to be used for *Identify Risks*. Because of this the topic is marked with "Incomplete Coverage" (yellow color).

5.2.10.3 Perform Qualitative Risk Analysis

According to the PMBOK *Perform Qualitative Risk Analysis* "is the process of prioritizing risks for further analysis or action by assessing and combining their probability of occurrence and impact". However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.10.4 Perform Quantitative Risk Analysis

According to the PMBOK *Perform Quantitative Risk Analysis* "is the process of numerically analyzing the effect of identified risks on overall project objectives". However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.10.5 Plan Risk Responses

The PMBOK states that *Plan Risk Responses* is "the process of developing options and actions to enhance opportunities and to reduce threats to project objectives". It also provides information about input, output and tools to be used in the process.

There is theory (yellow color) about *Plan Risk Responses* in the SWEBOK. This theory is in the *Software Project Planning* of the *Software Engineering Management* KA.

Given that specific tools are not provided, *Plan Risk Responses* topic is marked with "Incomplete Coverage" (yellow color).

5.2.10.6 Control Risks

About *Control Risks* the PMBOK states that "is the process of implementing risk response plans, tracking identified risks, monitoring residual risks, identifying new risks, and evaluating risk process effectiveness throughout the project". The guide also provides input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.11 Project Procurement Management

5.2.11.1 Plan Procurement Management

According to the PMBOK *Plan Procurement Management* is "the process of documenting project procurement decisions, specifying the approach, and identifying potential sellers". The guide also gives information about input, output and tools to be used in the process.

Useful theory (yellow color) like the need of handling supplier contracts is given in the SWEBOK. This is done in the *Software Project Enactment* topic of the *Software Engineering Management* KA.

Because the SWEBOK lacks of practical considerations about *Plan Procurement Management,* the topic is marked with "Incomplete Coverage" (yellow color).

5.2.11.2 Conduct Procurements

In the PMBOK *Conduct Procurements* is defined as "the process of obtaining seller responses, selecting a seller, and awarding a contract". It also provides input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.11.3 Control Procurements

According to the PMBOK *Control Procurements* is "the process of managing procurement relationships, monitoring contract performance, and making changes and corrections to contracts as appropriate". It also provides input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

5.2.11.4 Close Procurements

The PMBOK states that *Close Procurements* is “the process of completing each procurement”. It also provides input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that’s why is marked with “No Coverage” (red color).

5.2.12 Project Stakeholder Management

5.2.12.1 Identify Stakeholders

The PMBOK defines *Identify Stakeholders* as “the process of identifying the people, groups, or organizations that could impact or be impacted by a decision, activity, or outcome of the project, analyzing and documenting relevant information regarding their interests, involvement, interdependencies, influence, and potential impact on project success”. The guide also provides information about input, output and tools to be used in the process.

In the SWEBOK is provided theory (yellow color) about *Identify Stakeholders* in the *Initiation and Scope Definition* topic of the *Software Engineering Management* KA, and theory and practice (green color) in the *Requirements Process* topic of the *Software Requirements* KA.

However, the SWEBOK doesn’t provide a formal process to *Identify Stakeholders,* and that’s why is the topic is marked with “Incomplete Coverage” (yellow color).

5.2.12.2 Plan Stakeholder Management

According to the PMBOK *Plan Stakeholder Management* is “the process of developing appropriate management strategies to effectively engage stakeholders throughout the project life cycle, based on the analysis of their needs, interests, and potential impact on project success”. It also provides input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that’s why is marked with “No Coverage” (red color).

5.2.12.3 Manage Stakeholder Engagement

In the PMBOK *Manage Stakeholder Engagement* is defined as “the process of communicating and working with stakeholders to meet their needs/expectations, address issues as they occur, and foster appropriate stakeholder engagement in project activities throughout the project life cycle”. The guide also provides information about input, output and tools to be used in the process.

The SWEBOK provides theory (yellow color) about *Manage Stakeholder Engagement,* and talks about the need of providing measurements results to the involved stakeholders of a project. This information is provided in the *Software Project Enactment* topic of the *Software Engineering Management* KA.

On the other hand, the SWEBOK doesn't give information about *Manage Stakeholder Engagement*, which is why the topic is marked with "Incomplete Coverage".

5.2.12.4 Control Stakeholder Engagement

About *Control Stakeholder Engagement* the PMBOK states that "is the process of monitoring overall project stakeholder relationships and adjusting strategies and plans for engaging stakeholders". It also gives information on input, output and tools to be used in the process. However, in the SWEBOK there is no information that matches this specific topic and that's why is marked with "No Coverage" (red color).

6 Recommendations

In this section the aim is to give recommendations to practitioners and to the academic community, based on the results presented in the section *5. SPM Knowledge Mapping*.

6.1 Recommendations for Practitioners

The recommendations for practitioners intend to help them improve their skills on project management applied to software engineering. As already discussed, given that the accepted reference for software engineering professionals is the SWEBOK, the topics that the PMBOK could extend or add to what is contained in the SWEBOK, will be used to guide the practitioner to achieve a better project management knowledge. For every Knowledge Area in the PMBOK a set of recommendations will be given.

As a reminder, the following are the meanings of the colors used in the tables that are referenced in the recommendations: red for "No coverage", yellow for "Incomplete Coverage" and green for "Full coverage" of a particular knowledge in the SWEBOK. For more information about the tables, check the section *5.1. Mapping Strategy.*

6.1.1 Organizational Influence and Project Lifecycle

The "Organization Influence and Project Lifecycle" topic presents general concepts about the environment and context of a project. In *Table 4* it can be seen the coverage the SWEBOK has about this topic.

1. Organizational Influences and Project Lifecycle	🟥 1.1 Organizational Influences on Project Management
	🟨 1.2 Project Stakeholders and Governance
	🟨 1.3 Project team
	🟨 1.4 Project Life cycle

Table 4. Coverage in the SWEBOK of "Organizational Influence and Project Lifecycle" topic of the PMBOK

It is recommended to make a broader search of resources (i.e. the PMBOK) for this topic, especially about theory of "Organizational Influences on Project Management" which has no coverage in the SWEBOK. In the topics of "Project Stakeholders and Governances", "Project Team" and "Project Life Cycle", the SWEBOK misses theory about motivation, importance of stakeholder management and general concepts about phases and importance of a project life cycle.

6.1.2 Project Management Processes

The "Project Management Processes" topic presents general concepts about the process groups present in a project, and the interaction between them. In *Table 5* it can be seen the coverage the SWEBOK has about this topic.

2. Project Management Processes	🟩 2.1 Common Project Management Process Interactions
	🟨 2.2 Project Management Process Groups
	🟨 2.3 Initiating Process Group
	🟩 2.4 Planning Process Group
	🟩 2.5 Executing Process Group
	🟩 2.6 Monitoring and Controlling Process Group
	🟩 2.7 Closing Process Group
	🟩 2.8 Project Information
	🟥 2.9 Role of the Knowledge Areas

Table 5. Coverage in the SWEBOK of "Project Management Processes" topic of the PMBOK

The SWEBOK in its KAs has coverage of activities that belong to each process group detailed in the PMBOK. For general knowledge of this process groups and the activities developed in each one, the SWEBOK is a good source of knowledge. However, it is recommended to go to the PMBOK for a formal definition of "Project Management Process Groups". Also, the SWEBOK doesn't provide a theory of the difference between a phase of a project and a process group. The PMBOK can also provide better information (theory) about the general activities that fit in the "Initiation Process Group".

The "Role of the Knowledge Areas" is not considered because it has to do with the organization of the PMBOK guide, and it may not have sense to use it in the context of the recommendations.

6.1.3 Project Integration Management

The "Project Integration Management" topic presents information about planning and coordination of the activities in a project. In *Table 6* it can be seen the coverage the SWEBOK has about this topic.

3. Project Integration Management	3.1 Develop Project charter
	3.2 Develop Project Management Plan
	3.3 Direct and Manage Project Work
	3.4 Monitor and Control Project Work
	3.5 Perform Integrated Change Control
	3.6 Close Project or Phase

Table 6. Coverage in the SWEBOK of "Project Integration Management" topic of the PMBOK

It is recommended to check the PMBOK in order to get more information about theory and practices (what it contains and how to produce it) related to "Develop Project Charter", and practices (inputs and tools for producing the plan) for "Develop Project Management Plan". The project charter importance comes among other things, from the fact that the project charter exposes the contents that justify the existence and development of the project. On the other hand, the management plan will define the base processes for the whole project management.

The practitioner should search for theory and practice about "Direct and Manage Project Work". Even though the topic is fairly treated by the SWEBOK in topics like "Software Configuration Management" and "Software Engineering Management", it misses general concepts and tools. Examples of the missing theory and practices are: kind of actions to be taken (preventive, reactive and repair) and how to manage sellers and suppliers among others.

6.1.4 Project Scope Management

The “Project Scope Management” topic presents information on activities to define and control the scope of a project. In *Table 7* it can be seen the coverage the SWEBOK has about this topic.

4. Project Scope Management	4.1 Plan Scope Management
	4.2 Collect requirements
	4.3 Define Scope
	4.4 Create WBS
	4.4 Validate Scope
	4.6 Control Scope

Table 7. Coverage in the SWEBOK of “Project Scope Management” topic of the PMBOK

The practitioner should look for theory and practices for “Plan Scope Management” topic, given that the SWEBOK doesn’t provide an explicit guide of the content of the plan. The PMBOK is a good source of information on how to control a scope management plan. This plan contains information about how the scope definition and scope control process should be carried out.

On the other hand, it is also recommended to look for further information in the PMBOK about how to build (input, tools and output) the WBS (Work Breakdown Structure).

The “Validation Scope” and “Control Scope” topics are covered in the SWEBOK from the point of view of requirements, but it misses practices and theory like the importance of deliverable determination and the process for deliverable determination.

6.1.5 Project Time Management

The “Project Time Management” topic presents concepts and practices related to definition, time estimation and control of the activities in a project. In *Table 8* it can be seen the coverage the SWEBOK has about this topic.

5. Project Time Management	5.1 Plan Schedule Management

	5.2 Define Activities
	5.3 Sequence Activities
	5.4 Estimate Activity Resources
	5.5 Estimate Activity Durations
	5.5 Develop Schedule
	5.7 Control Schedule

Table 8. Coverage in the SWEBOK of "Project Time Management" topic of the PMBOK

The "Plan Schedule Management" topic is not explicitly covered in the SWEBOK, so theory and practice should be looked for in order for about the creation process of the plan. About "Define Activities", "Sequence Activities", "Estimate Activity Resources", "Estimate Activity Durations", "Develop Schedule" and "Control Schedule", the SWEBOK only mentions a couple of practices to apply, but other than that there is very few information about the topics. That's why it is strongly recommended to look for concepts and practices in resources like the PMBOK.

Information provided by the PMBOK and missed in the SWEBOK includes: resources (inputs) that serve as base to estimate activities' duration, how to establish a relationship between activities, and a proper guide of useful theory and practice about how to produce and maintain a schedule.

6.1.6 Project Cost Management

The "Project Cost Management" topic gives information on how to estimate costs and determine a control the budget of a project. In *Table 9* it can be seen the coverage the SWEBOK has about this topic.

6. Project Cost Management	6.1 Plan Cost Management
	6.2 Estimate Costs

	6.3 Determine Budget
	6.4 Control Costs

Table 9. Coverage in the SWEBOK of "Project Cost Management" topic of the PMBOK

As with the time management topic, the SWEBOK only presents very general information about "Project Cost Management" topic. In the "Estimate Costs" topic only some theory is mentioned but practices and tools are not exposed. "Determine Budget" topic has no coverage in the SWEBOK, so research about theory and practice is also recommended.

The PMBOK is a good a resource for what is missing in the SWEBOK, providing theory and practices to be considered like: CoQ (Cost of Quality), techniques for estimation and Group Decision-Making Techniques.

6.1.7 Project Quality Management

The "Project Quality Management" topic gives information on how put in place quality oriented processes in a project. In *Table 10* it can be seen the coverage the SWEBOK has about this topic.

7. Project Quality Management	7.1 Plan Quality Management
	7.2 Perform Quality Assurance
	7.3 Control Quality

Table 10. Coverage in the SWEBOK of "Project Quality Management" topic of the PMBOK

The SWEBOK covers widely the guidance on project quality in the context of a software project. That's why is not considered relevant to make recommendations in this topic.

6.1.8 Project Human Resource Management

The "Project Resource Management" topic presents guide on how to manage and lead a project team. In *Table 11* it can be seen the coverage the SWEBOK has about this topic.

8. Project Human Resource Management	8.1 Plan Human Resource Management
	8.2 Acquire Project Team
	8.3 Develop Project Team

	8.4 Manage Project Team

Table 11. Coverage in the SWEBOK of "Project Human Resource Management" topic of the PMBOK

Given the little coverage by the SWEBOK about "Project Human Resource Management", it is recommended to search for resources like the PMBOK for ways to approach the topic. The SWEBOK doesn't mention either "Plan Human Resource Management", "Acquire Project Team" or "Develop Project Team" topics. It only gives some theory and practical considerations about reviews and evaluation, which could be taken as part of "Manage Project Team" topic.

The PMBOK provides theory and practice missing in the SWEBOK, which goes from acquisition of the team members to motivation of the team. It is worth saying that team management is a key skill for every project manager, no matter the context.

6.1.9 Project Communications Management

The "Project Communications Management" topic presents guide on what's necessary to ensure that the project information is available to the relevant stakeholders. In *Table 12* it can be seen the coverage the SWEBOK has about this topic.

9. Project Communications Management	9.1 Plan Communications Management
	9.2 Manage Communications
	9.3 Control Communications

Table 12. Coverage in the SWEBOK of "Project Communications Management" topic of the PMBOK

There is no information (coverage) in the SWEBOK on how to execute "Project Communication Management". It misses theory and practical considerations about topics like: Communication Requirements Analysis, Communication Technology, Communication Models and Communication Methods. So, the recommendation is to search for the topic in the PMBOK or other resources.

6.1.10 Project Risk Management

The "Project Risk Management" topic gives information about how to indent and handle risk in a project. In *Table 13* it can be seen the coverage the SWEBOK has about this topic.

10. Project Risk Management	🟥 10.1 Plan Risk Management
	🟨 10.2 Identify Risks
	🟥 10.3 Perform Qualitative Risk Analysis
	🟥 10.4 Perform Quantitative Risk Analysis
	🟨 10.5 Plan Risk Responses
	🟥 10.6 Control Risks

Table 13. Coverage in the SWEBOK of "Project Risk Management" topic of the PMBOK

The "Software Engineering Management" of the SWEBOK mentions the "Risk Management" topic and some useful practices. However, it is recommended to get knowledge from the PMBOK and other resources, given that there is no coverage of topics like: "Plan Risk Management", "Perform Qualitative Risk Analysis", "Perform Quantitative Analysis" and "Control Risks".

Some theory and practices are presented by the SWEBOK, like the need of "Identify Risks" and "Plan Risk Responses" by using tools like of decision trees and project abandon policies. However, the PMBOK gives much broader information about input, techniques and outputs related to the topics. The information in the PMBOK includes: Diagramming Techniques (i.e. Influence Diagrams), SWOT Analysis (strengths, weaknesses, opportunities, and threats) and Risk Register Process among other useful theory.

6.1.11 Project Procurement Management

The "Project Procurement Management" topic provides a guide for the purchase of resources from outside the project team. In *Table 14* it can be seen the coverage the SWEBOK has about this topic.

11. Project Procurement Management	🟨 11.1 Plan Procurement Management
	🟥 11.2 Conduct Procurements
	🟥 11.3 Control Procurements
	🟥 11.4 Close Procurements

Table 14. Coverage in the SWEBOK of "Project Procurement Management" topic of the PMBOK

The SWEBOK mentions supplier contract management in its "Software Engineering Management" KA. However it misses theory and practices information about “Conduct Procurements” (selecting seller an awarding contract), “Control Procurements” (monitor seller performance) and “Close Procurements” (complete procurements) topics.

Because of the above, is recommended to look for information in the PMBOK and other resources, about how to plan and conduct the procurements until they are “close”.

6.1.12 Project Stakeholder Management

The “Project Stakeholder Management” topic provides a guide for identifying the stakeholders and managing their involvement in the project. In *Table 15* it can be seen the coverage the SWEBOK has about this topic.

12. Project Stakeholder Management	12.1 Identify Stakeholders
	12.2 Plan Stakeholder Management
	12.3 Manage Stakeholder Engagement
	12.4 Control Stakeholder Engagement

Table 15. Coverage in the SWEBOK of “Project Stakeholder Management” topic of the PMBOK

The “Software Requirements” KA of the SWEBOK provides information about interaction with stakeholders for the requirements elicitation process. However, the engagement of the stakeholders should be managed throughout the whole project.

It is recommended to look for guidance in the topic of “Identify Stakeholders”, in which it can be defined who will interact in different phases of the project. Also the establishment of a plan to manage the stakeholders involvement (Manage Stakeholder Engagement topic) is a required knowledge.

On the other hand, the SWEBOK has no coverage for “Plan Stakeholder Management” and “Control Stakeholder Engagement”. So it has no explicit information about the appropriate way to establish strategies to keep communication with the stakeholders, and also misses the monitoring process of those strategies.

6.2 Recommendations for Academic Community

In the context of education a broadly accepted reference for master level programs is the GSwE2009 Standard. The standard is based on the SWEBOK and extends it to "accommodate the needs and views of academia, industry, and the computing professional societies".

The extended parts in the GSwE2009 are:

- Human Computer Interface design in the Software Design KA

- Addition of an Engineering Economics unit in the Software Engineering Management KA

- Addition of a Risk Management unit in the Software Engineering Management KA

- Addition of a Verification and Validation (V&V) unit in the Software Quality KA

- Changes in the names and the unit/topic organization in three KAs: (a) Software Requirements to Requirements Engineering, (b) Software Testing to Testing and (c) Software Configuration Management to Configuration Management. These changes were made to accommodate and emphasize the role of SE in GSwE2009.

From what was added to the GSwE2009, Engineering Economics and the Risk Management Unit are the direct management-related content. Engineering Economics is not considered in the PMBOK, so is discarded from the comparison and the recommendations. Risk Management on the other hand, given that is extended, is not considered for the recommendations because is already covered by the content of the master level programs that are based in the GSwE2009.

Given the similarities between the GSwE2009 and the SWEBOK, the recommendation in the section *6.1 Recommendations for Practitioners* are taken as valid for the academic community too. It is recommended then, that the academic community help the practitioners that come to a master program, to achieve everything that is pointed out in the section *6.1 Recommendations for Practitioners* as a weak point in the SWEBOK and therefore in the GSwE2009. In order to get the skills in the KAs gathered in the PMBOK, a master should provide the students at least with the level of "Application" of the Bloom's Taxonomy (*Table 16*).

	Bloom's Taxonomy Level	Associated Keywords
K	**Knowledge:** Recall data	Defines, describes, identifies, knows, labels, lists, matches, names, outlines, recalls, recognizes, reproduces, selects, states
C	**Comprehension:** Understand the meaning, translation, interpolation, and interpretation of instructions and problems; state a problem in one's own words.	Comprehends, converts, defends, distinguishes, estimates, explains, extends, generalizes, gives examples, infers, interprets, paraphrases, predicts, rewrites, summarizes, translates

AP	**Application:** Use a concept in a new situation or use an abstraction unprompted; apply what was learned in the classroom to novel situations in the workplace	Applies, changes, computes, constructs, demonstrates, discovers, manipulates, modifies, operates, predicts, prepares, produces, relates, shows, solves, uses
AN	**Analysis:** Separate material or concepts into component parts so that its organizational structure may be understood; distinguish between facts and inferences	Analyzes, breaks down, compares, contrasts, diagrams, deconstructs, differentiates, discriminates, distinguishes, identifies, illustrates, infers, outlines, relates, selects, separates
S	**Synthesis:** Build a structure or pattern from diverse elements; put parts together to form a whole, with emphasis on creating a new meaning or structure	Categorizes, combines, compiles, composes, creates, devises, designs, explains, generates, modifies, organizes, plans, rearranges, reconstructs, relates, reorganizes, revises, rewrites, summarizes, tells, writes
E	**Evaluation:** Make judgments about the value of ideas or materials	Appraises, compares, concludes, contrasts, criticizes, critiques, defends, describes, discriminates, evaluates, explains, interprets, justifies, relates, summarizes, supports

Table 16. Bloom's Taxonomy

7 Conclusion

There are plenty of information sources for learning about project management and software engineering. It is important for practitioners and members of the academic community to find references that make easier the guidance on these topics. The conducted work in this thesis intends to be one more source of guidance, by exposing the weaknesses and strengths of some of the standard references (SWEBOK, PMBOK and GSwE2009) for the industry and the academic world.

For a practitioner or teacher, the recommendations made in this thesis could help improve substantially their approach to the learning and teaching of project management in the context of software engineering. The presented results by this research should be considered one more tool but not an absolute true about the learning of project management.

The analysis made for practitioners recommendations, show that the reference (SWEBOK) for software engineering professionals leaves some gaps for management and organizational matters. Important topics like team management, estimation of time and costs, and organizational influences, don't have enough visibility in the SWEBOK.

For a proper teaching of software engineering management, the academic community needs to focus also in the gaps left by the professional standard (SWEBOK). It's in the hands of the academic community, to gather and provide useful resources when designing the content of a master program. A reinforcement of the project management skills at the master level, will most likely result better professionals, and as a consequence in a higher chance of successful projects in the industry.

The work carried out in this thesis could be complemented by other studies. For example, which of the spotted missing topics in the SWEBOK and the GSwE2009 are more critical for project success. Also, how the missing topics could get more chances to be properly taught, and overcome the limitations of lectures in a classroom and not in a more practical context. The level (Bloom's taxonomy) at which every topic should taught will depend on the audience of the program (newly graduated professionals, executive and/or non-executive company employees, etc.).

In the end, every strategy that leads to more preparation for real life projects will provide better reinforcement to software engineering management learning process.

8 References

[1] PMBOK Guide Fifth Edition. Project Management Institute. 2013.

[2] Why Projects Fail: Avoiding the Classic Pitfalls, Oracle Inc. http://www.oracle.com/us/solutions/018860.pdf. October 2011.

[3] Why Software Projects Fail: A New Assessment of Risk. Alpha Software Corporation. https://www.alphasoftware.com/documentation/200801_wpaper/low_risk_a5.pdf. January 2008.

[4] Project management in defense: The essential capability, McKinsey & Co. 2013.

[5] The Value of Project Management, Project Management Institue. 2010.

[6] G. Bavota, A. D. Lucía, F. Fasano, R. Oliveto and C. Zottoli. Teaching software engineering and software project management: An integrated and practical approach. Software Engineering (ICSE), 2012 34th International Conference.

[7] Guide to the Software Engineering Body of Knowledge, 2004 Version. IEEE Computer Society.

[8] Graduate Software Engineering 2009(GSwE2009) Curriculum Guidelines for Graduate Degree Programs in Software Engineering. Integrated Software & Systems Engineering Curriculum (iSSEc) Project. 2009.

[9] M. Ardis and P. Henderson Standing on our Shoulders. Software Engineering Education (SEEd), ACM SIGSOFT Software Engineering Notes 35(2), 2010.