

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

CMMI Level 3 Engineering Processes Implementation Project

Mariana Figueiredo

Project's Report

Master in Informatics and Computing Engineering

Supervisor: Ana Cristina Paiva (PhD)

July 2008

CMMI Level 3 Engineering Processes Implementation Project

Mariana Figueiredo

Project's Report

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: João Pascoal Faria (PhD)

Main Examiner: Ricardo Jorge Machado (PhD)

Examiner: Ana Cristina Paiva (PhD)

15th July 2008

Abstract

This project was part of a process improvement organizational effort to achieve CMMI Maturity Level 3.

It was focused in the Engineering Level 3 Process Areas: Technical Solution (TS), Product Integration (PI), Verification (VER) and Validation (VAL), and its main goals were to define the Integration, Verification and Validation, and the Design and Implementation processes. The former was built from scratch to cover the PI, VER and VAL while the definition of the latter was monitored to make sure it covered the TS process area and applied to the reality of the organization.

The definition of the processes and related support documentation was based on a combined analysis of the CMMI process areas, the Quality Management System in place and of the current practices in use. To deepen the analysis to the current situation, a set of interviews, to 16 interviewees, was carried out and the data from the interviews analysed.

The project faced other problem, that was the publishing of the QMS. Up until this project, the QMS was published as a set of documents in the intranet, but an opportunity arose to consider a framework that could improve the navigability of the QMS contents. So, a study was conducted and the migration effort estimated.

Besides the process definition, some procedures, guidelines and templates were also defined: Integration, Verification and Validation Plan template, Test Report template, Release Procedure, Release Notes templates, Release Checklists.

Although no proof could yet be provided about the performance of these processes, the organization will use them and rework them until the processes trully reflect the organization's reality. Even if the processes can not be found succesfully implemented, based on the studies performed, the author strongly believes that these processes will be a corner stone to the organization's CMMI Maturity Level 3 certification.

Resumo

Este projecto fez parte de um esforço organizacional para conseguir a certificação de nível 3 de CMMI.

Focou-se nas áreas de engenharia de nível 3: Technical Solution (TS), Product Integration (PI), Verification (VER) and Validation (VAL) e os seus principais objectivos eram definir o processo Integration, Verification and Validation e o processo Design and Implementation. O primeiro foi definido de raiz de maneira a cobrir as práticas das áreas PI, VER e VAL, enquanto que a definição do segundo foi acompanhada de modo a garantir que a área TS era coberta e que o processo se adaptava à realidade da organização.

A definição dos processos e respectiva documentação de suporte baseou-se numa análise das áreas de CMMI, do QMS e das práticas correntes da organização. De modo a aprofundar a análise da situação actual da organização, realizou-se um conjunto de entrevistas, com 16 entrevistados.

O projecto tinha de considerar ainda outra questão, a da publicação do QMS. Até à realização deste projecto, o QMS era disponibilizado como um conjunto de documentos na intranet da empresa, mas surgiu a oportunidade de analisar uma framework que melhoraria a navegabilidade dos conteúdos do QMS. Assim estimaram-se esforços de migração do QMS que estava a ser desenvolvido para a nova framework – o WebViewer.

Para além da definição de processos, definiram-se também procedimentos, guidelines e templates: Integration, Verification and Validation Plan template, Test Report template, Release Procedure, Release Notes templates, Release Checklists.

Embora não tenha sido possível arranjar provas concretas do comportamento dos processos aqui definidos, a organização irá utilizá-los, pô-los à prova, melhorá-los, até que os processos reflectam toda e qualquer realidade da organização. Portanto, mesmo que não se possa considerar que os processos foram implementados com sucesso, com base nos estudos realizados, a autora acredita que estes processos serão uma pedra basilar na obtenção da certificação de nível 3.

Acknowledgments

I would like to thank, first and foremost, my supervisor, Prof. Ana Paiva, for being a great teacher and for helping me acquire the taste for quality.

To Teresa Carreiro, Porto DC's Quality Manager, a specially thanks for making me feel at ease with the quality concepts within an organization and specially for believing in me and for giving me the freedom to do what I thought was right.

To Bruno Martins, for being there to debate any issue I needed to clarify, and for being right there, every step of the way.

To Hugo Magalhães, for all the guidance and for always being available to review my workproducts.

To Emanuel Pinto, the go-to guy, for always having an answer and helping me find my own.

And to Frederico Sampaio, Alexandrina Lemos, André Lessa, and all the Quality Management Team, for making me feel as part of the team from the very first day.

Mariana Figueiredo

“All models are wrong but some are useful.”

George Box, Quality and Statistics Engineer

Contents

Abstract	i
Resumo	iii
Acknowledgments	v
1 Introduction	1
1.1 Context	1
1.2 Project	1
1.3 Motivation and Goals	2
1.4 Report Structure	2
2 State of the Art	5
2.1 Process Improvement	5
2.2 CMMI	9
2.2.1 About CMMI: evolution of CMMI	9
2.2.2 CMMI for Development	11
2.2.3 Adopting CMMI	18
2.3 Qimonda Porto Development Center's Quality Management System (QMS)	19
2.3.1 Organizational Environment	19
2.3.2 Quality Management System	20
2.4 Porto DC and CMMI	22
2.5 Conclusions	24
3 Problem and Current Situation Analysis	27
3.1 The Problem	27
3.2 The Approach	28
3.3 Conclusions	31
4 The New QMS Engineering Processes	33
4.1 Solution Development	33
4.1.1 Release Notes	33
4.1.2 Release Mail	40
4.1.3 Release Checklists	41
4.2 The New Processes and Support Documents	41
4.2.1 Design and Implementation	41

CONTENTS

4.2.1.1	Support Documents	45
4.2.2	Integration, Verification and Validation Process	50
4.2.2.1	Support Documents	65
4.3	The WebViewer	68
4.4	Conclusions	70
5	Conclusions and Future Work	73
5.1	Conclusions	73
5.2	Future Work	74
	References	78
A	ISO/IEC 15504	79
B	Capability Levels	81
C	Goals and Practices	83
C.1	Generic Goals and Practices	83
C.2	Specific Goals and Practices	83
D	SCAMPI B Findings	87
E	Release Notes QMS 1.1.1 Template	89
F	Release Checklists	93
G	Integration, Verification and Validation Plan	95
H	Release Templates	105
H.1	Word Template	105
H.2	Excel Template	114

List of Figures

2.1	Qimonda IT PA organization	19
2.2	QMS 1.1.1 Structure	21
2.3	Mapping CMMI v1.2 – QMS 1.1.1	22
2.4	Mapping CMMI v1.2 – QMS 2.0	23
3.1	Method followed in the approach to the problem	29
4.1	Table from Chapter 3 of the Release Notes template	34
4.2	Master Data section 3.1 table	34
4.3	Master Data section 3.2 table	34
4.4	Table from Chapter 2 of the Release Notes template	35
4.5	Components chapter 2 table	35
4.6	Components chapter 3 table	35
4.7	Table from Chapter 6 of the Release Notes template	36
4.8	Components chapter 6 table	36
4.9	Components section 6.1 table	36
4.10	Components chapter 7 table	37
4.11	WT chapter 2 table	39
4.12	WT chapter 6 table	39
4.13	Figure 1 – Relationship between the Design and Implementation process activities and the Software Development Process phases	46
4.14	DI.A1	47
4.15	DI.A2	48
4.16	DI.A3	49
4.17	Figure 1 - IVV Process Overview	52
4.18	Figure 2 - Relationship between the process activities and the Software Development Process phases	56
4.19	IVV.A1	57
4.20	IVV.A2	58
4.21	IVV.A3	59
4.22	IVV.A4	60
4.23	IVV.A5	61
4.24	IVV.A6	62
4.25	IVV.A7	63
4.26	IVV.A8	64
D.1	SCAMPI B findings by <i>Colin Benton</i>	87

LIST OF FIGURES

E.1	Table 1 - Status	90
E.2	Table 2 - Functional changes	90
E.3	Table 3 - Language support	90
E.4	Table 4 - Solved problems	91
F.1	Release Checklist Template	93
G.1	IVV Plan - Page 1	96
G.2	IVV Plan - Page 2	97
G.3	IVV Plan - Page 3	98
G.4	IVV Plan - Page 4	99
G.5	IVV Plan - Page 5	100
G.6	IVV Plan - Page 6	101
G.7	IVV Plan - Page 7	102
G.8	IVV Plan - Page 8	103
G.9	IVV Plan - Page 9	104
H.1	Release Notes template - Page 1	106
H.2	Release Notes template - Page 2	107
H.3	Release Notes template - Page 3	108
H.4	Release Notes template - Page 4	109
H.5	Release Notes template - Page 5	110
H.6	Release Notes template - Page 6	111
H.7	Release Notes template - Page 7	112
H.8	Release Notes template - Page 8	113
H.9	Release Notes template - FrontPage	114
H.10	Release Notes template - ClearCase view Configuration	114
H.11	Release Notes template - Overview	115
H.12	Release Notes template - Test Results	115
H.13	Release Notes template - Known Issues	115
H.14	Release Notes template - Code Coverage	115

List of Tables

2.1	Performance Results, July 2006	7
2.2	Comparative Advantages of Continuous and Staged Representations .	12
2.3	Capability and Maturity Levels	13
2.4	Process Areas by Categories and its Maturity Levels	16
3.1	Interviews Questions	30
4.1	Table1- Process Resources	43
4.2	Process Data	70
4.3	Migration Effort Estimates (part 1)	71
4.4	Migration Effort Estimates (part 2)	72
4.5	Total Estimates	72
A.1	Capability Levels	80

LIST OF TABLES

Abbreviations

ARC	Appraisal Requirements for CMMI
AS	Analysis Solutions
BPA	Business Process Automation
CAR	Causal Analysis and Resolution
CM	Configuration Management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMMI-DEV	CMMI for Development
CPT	Cross Platform Technologies
DAR	Decision Analysis and Resolution
DBT	DataBase Technologies
DC	Development Center
DC PT	Porto Development Center
DC SHZ	SuZhou Development Center
DF	Domain Function
DI	Design and Implementation
DQ	Delivery and Quality
EIA	Electronic Industries Alliance
EPIC	Enterprise Process Improvement Collaboration
ES	Execution Solutions
GG	Generic Goals
GP	Generic Practices
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
IPD-CMM	Integrated Product Development CMM
IPM	Integrated Project Management
IPPD	Integrated Product and Process Development
ISO	International Organization for Standardization
IT	Information Technology
IT PA	Information Technology Product Automation
IVV	Integration, Verification and Validation
MA	Measurement and Analysis
NA	Not Applicable
OID	Organizational Innovation and Deployment
OPD	Organizational Process Definition

ABBREVIATIONS

OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PA	Process Area
PE	Program Execution
PI	Product Integration
PM	Project Management
PMC	Project Monitoring and Control
PP	Project Planning
PPQA	Process and Product Quality Assurance
P-CMM	People Capability Maturity Model
Q IT PA	Qimonda Information Technology Product Automation
QM	Quality Management
QMS	Quality Management System
QPM	Quantitative Project Management
RD	Requirements Development
REQM	Requirements Management
RSKM	Risk Management
SA-CMM	Software Acquisition CMM
SAM	Supplier Agreement Management
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SDP	Software Development Process
SECAM	Systems Engineering Capability and Assessment Method
SECM	Systems Engineering Capability Model
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
SG	Specific Goals
SI	System Integration
SP	Specific Practices
SPICE	Software Process Improvement and Capability dEtermination
SQA	Software Quality Assurance Engineer
ST	Software Technology
SW-CMM	CMM for Software
TR	Technical Report
TS	Technical Solution
UTP	Unified Tracking Platform
VAL	Validation
VER	Verification
WT	Windows Technologies

Chapter 1

Introduction

1.1 Context

This project focused on the CMMI Engineering Level 3 Process Areas – Technical Solution, Product Integration, Verification, and Validation – implementation on the Porto Development Center of Qimonda AG.

The Porto Development Center owns the implementation phase within the Qimonda IT PA, i.e., is responsible for the detailed design, implementation, testing and integration of solutions.

This project was part of a process improvement organization effort in order to achieve CMMI Maturity Level 3 certification. It was focused on the mentioned process areas as those were the ones that covered the major responsibilities of the Porto DC.

1.2 Project

As part of an organizational effort to evolve the implemented processes to achieve a CMMI Maturity Level 3, this project was expected to be a part of the efforts to define the new processes that were going to cover the engineering level 3 process areas. It was of this projects scope the definition of the Integration, Verification and Validation process, that covers the Product Integration, Verification and Validation process areas; and to monitor the efforts of SuZhou DC in the definition of the Design and Implementation process, that covers the Technical Solution process area.

The author was expected to study the CMMI process areas and to analyse the Quality Management System in place, to identify the practices that already responded to the new CMMI process areas practices. In order to deepen the analysis

of Porto DC current practices, interviews with project team members appointed by the local SEPG took place and allowed the author to gather more data on the DC's current practices.

After all the analysis – the CMMI study, the QMS study and the analysis to the data collected from the interviews – the process was defined, along with all its support documentation.

1.3 Motivation and Goals

This project was expected to define the Integration, Verification and Validation process, and its related documentation – procedures, templates, guidelines – as well as to monitor the definition of the Design and Implementation process. This would allow the organization to achieve a CMMI maturity level 3.

1.4 Report Structure

The first Chapter of this report, **Introduction**, provides general information about the Project. There is a presentation of the context of the project and of the project itself, along with the goals to achieve and the motivation behind the project. This chapter also presents the structure of this report.

The second Chapter, **State-of-the-Art**, contains an introduction to process improvement, the model approached in this project, CMMI - Capability Maturity Model Integration, a presentation of the Qimonda Porto Development Center's Quality Management System (QMS) and the implementation of CMMI in Porto DC. The section CMMI contains an explanation of the evolution of the CMMI, the CMMI for Development and of what it takes to adopt CMMI in an organization. The section Qimonda Porto development Center's QMS provides a quick contextualization of the project in the organization (Qimonda and Porto DC), along with a full presentation of the QMS in use. The section Porto DC and CMMI presents the beginning of the CMMI in the organization, along with the route plan do CMMI Level 3 and an explanation of the focus of this project.

The third Chapter, **Problem and Current Situation Analysis**, presents the problem that this project faced and all that was done to understand the current practices in Porto DC and to define the processes.

The forth Chapter, **The New QMS Engineering Processes**, provides all the information about the development of the solution, the new processes defined to address the new process areas of CMMI level 3 and all the support documentation (procedures, guidelines, templates) related to those processes. It also includes the proposed approach to the publication of the QMS – the WebViewer. - conclusions

Introduction

and future work The fifth and Final Chapter, **Conclusions and Future Work**, contains the conclusions drawn from the development of the project during the four months it lasted, as well as considerations about the future work on this project.

Introduction

Chapter 2

State of the Art

2.1 Process Improvement

IT'S TIME FOR PROCESS IMPROVEMENT

When your organization consistently misses its commitments, its management has no visibility at all into progress, there are a series of quality problems and poor morale, there is a major warning of process failure in the air ...it's time for Process Improvement.

If you struggle with late deliveries, last minute crunches and spiraling costs: your organization is missing its commitments, ...it's time for Process Improvement.

If you, as top management, are always being surprised (and not in a good way), ...it's time for Process Improvement.

If your team struggles with too much rework, functions that do not work correctly and customer complaints after delivery, ...it's time for Process Improvement.

Process Improvement

Organizations typically focus on three critical dimensions: people, procedures and methods, and tools and equipment, but the processes used in the organization are what holds everything together; for they provide a way to incorporate knowledge of how to do things better in the organization, allowing it to align the way it does business, and to make the most of its resources.

A focus on processes instead of in the final product provides a way to maximize the productivity of people and the use of technology to be more competitive.

“Manufacturing has long recognized the importance of process effectiveness and efficiency. [...] Process helps an organization’s workforce [...] work smarter, not harder.” [1, section About Capability Maturity Models]

Watts Humphrey [2], and others, extended the principles created by Walter Shewart [3] – based on statistical quality control – and began applying them to software in their work at the SEI, providing a description of the basic principles and concepts on which many of the capability maturity models (CMMs) are based.

The *Process Management Premise*, a long-established premise in manufacturing, states that

“The quality of a system or product is highly influenced by the quality of the process used to develop and maintain it.”

This premise implies a focus on processes, more than on products, and belief in it is seen worldwide in quality movements, as evidenced by the ISO/IEC body of standards or the SEI CMMs.

The culture of the organization is a major enabler or obstacle of process improvement, for those who have to implement the process in their daily work. There are some common misconceptions about processes, like the ones stated in the CMMI Overview Presentation available in SEIs webpage [4].

“I don’t need process, I have
really good people
advanced technology
an experienced manager

Process
interferes with creativity
equals bureaucracy + regimentation
isn’t needed when building prototypes
is only useful on large projects
hinders agility in fast-growing markets
costs too much”

In fact process implementation has a great impact on reducing cost, enhancing predictability (schedule), the bureaucracy and regimentation are really not an overhead as process improves productivity, and it increases customer satisfaction and the overall quality of the product (see table 2.1). Process should be used in small as well as in large projects for the lessons learned it incorporates are important for every product and project.

How to improve the processes in your organization?

PROCESS MODELS

“Process models are structured collections of practices that describe the characteristics of effective processes. The practices included are those proven by experience to be effective.” [4]

A process model can be a guide for the improvement of processes within an organization, providing help to define process improvement objectives and priorities, and helping ensure stable and mature processes. It is also used with an appraisal method to diagnose the state of an organization’s current practices. A process model also provides a place to start improving the processes, providing the benefit of a community’s prior experiences.

WHY CMMI?

CMMI, Capability Maturity Model Integration, was created by SEI and first released in 2000, incorporating lessons learned from the use of the SW-CMM and EIA-731, among other standards and models.

As a process model, CMMI provides guidance for improvement across multiple process disciplines in an organization. One should, hence, use CMMI as a collection of best practices and as a framework for organizing and prioritizing the activities of process improvement. This will allow one to emphasize the alignment of the process improvement objectives with organizational business objectives.

On July 2006, SEI released some performance results of CMMI-based process improvement [5]. The results were based on the results from 30 different organizations and were reflected on six performance categories - cost, schedule, productivity, quality, customer satisfaction, and return on investment.

Performance Category	Median Improvement
Cost	34%
Schedule	50%
Productivity	61%
Quality	48%
Customer Satisfaction	14%
Return on Investment	4:1

Table 2.1: Performance Results, July 2006

Good examples of these results are 3H Technology [6] – after two years of process improvement based on CMMI, had significant improvement in average number of defects found – and the Software maintenance group at Warner Robins Air Logistics Center [7] – after achieving level 5 maturity, had the schedule variance significantly reduced.

SEI lists Accenture, Boeing, FAA, Intel, NASA, Nokia, US Army, Bank of America, General Motors, IBM, BMW, among others as organizations using CMMI.

Over all, CMMI improved schedule and budget predictability, cycle time, quality (as measured by defects), and employee morale; increased productivity, customer satisfaction, and the return on investment; and decreased the cost of quality [4, slide 39].

IS IT JUST CMMI? OR ARE THERE OTHERS?

ISO/IEC TR 15504, also known as SPICE - Software Process Improvement and Capability dEtermination - first appeared on 1997, as a framework for the assessment of processes, developed by a Joint Technical Subcommittee between ISO - International Organization for Standardization - and IEC - International Electrotechnical Commission. For more on this standard please consult Appendix A.

This standard was concerned about the many national maturity model proposals and established an international standard in this area. It had a reference model, but in a major revision to the draft standard in 2004, the process reference model was removed, so that the standard now specifies the measurement framework and can use different process reference models.

However, ISO/IEC TR 15504 [8] has not yet been as successful as the CMMI, some say because:

- This standard is not available as free download, it must be purchased from the ISO, while CMM and CMMI are available as free downloads from the SEI website.
- CMMI is actively sponsored (specially by the US Department of Defense).
- CMM was created first and reached critical “market” share before this standard became available.
- The CMM has been replaced with CMMI, which incorporates many of the ideas of ISO/IEC 15504 but also retains the benefits of the CMM.

Bottom line was that by the time the organization in which this project was developed – Qimonda’s Porto Development Center – had to choose a standard to guide its processes improvement, CMMI was the standard chosen as reference, for its popularity and background. When this project started, the organization had already partially improved its processes to match the CMMI, so the engineering processes implementation reported here was based on the CMMI.

2.2 CMMI

2.2.1 About CMMI: evolution of CMMI

The CMMI model and appraisal process was inspired in five principal ideas [1, page 5, “CMMI: History and Direction”]:

1. Planning, tracking, and schedule management
2. Requirements definition and configuration control
3. Process assessment
4. Quality measurement and continuous improvement
5. Evolutionary improvement

These ideas were not yet been applied to software and systems development work when CMM development started, but when trying to improve software source selection for the U.S. Air Force, SEI found that organizations that used the best management and technical practices in their development projects seemed likely to do the best work, so they devised an 85-question questionnaire that covered project planning, project tracking, schedule management, requirements management, configuration control, quality measurement and continuous process improvement. SEI then grouped the questions in a maturity framework in order to rank the results, and that became the first version of what ultimately became CMMI. For more on the beginning of CMMI please refer to [1, pages 5–8, “CMMI: History and Direction”].

CMM focus on improving processes in an organization; they describe an improvement path from ad hoc, immature processes to disciplined, mature processes. The SEI created the first CMM designed for software organizations – Capability Maturity Model for Software (later named SW-CMM) – applying the principles introduced by manufacturing industries to this cycle of process improvement. The success of this model inspired others, so two models were produced in 1994: the Systems Engineering CMM, created by the Enterprise Process Improvement Collaboration (EPIC) with SEI participation; and the Systems Engineering Capability and Assessment Method (SECAM), created by the International Council on Systems Engineering (INCOSE). These two models were later merged into Electronic Industries Alliance (EIA) Interim Standard 731 (SECM). Concerned about preserving and enhancing the capabilities of staff, SEI created the People Capability Maturity Model (P-CMM) in 1995. That year SW-CMM was also being updated, and a model was being produced for Integrated Product Development CMM (IPD-CMM). In 1996, the Software Acquisition Capability Maturity Model (SA-CMM)

was created to cover key practices in software acquisition. However, applying multiple models in an organization is expensive in terms of training, appraisals, and improvement activities. Furthermore, SEI saw that software was going in one direction and systems engineering was going in another, and that software problems were still a large cause of program failures, so decided it was important to bring stronger systems engineering into play. The CMM Integration project was formed to sort out these problems. CMMI was first released in 2000, v1.02 in December, and included versions for systems engineering, software engineering and integrated product and process development. Hence, CMMI is a result of the evolution of the SW-CMM, the SECM, and the IPD-CMM. The first model was released as v1.1 in December 2001, after a refinement of the initial version. In April 2002 was added a variant that included some of the acquisition practices as a Supplier Sourcing.

The naming convention used for CMM models led to the vision that the principal activities within the organizations to which CMMI applies are software and systems engineering, but to gain maximum benefit from CMMI adoption and implementation, CMMI must be applied to the entire development structure of the organization. With that purpose, the latest release of CMMI (v1.2) is called CMMI for Development (CMMI-DEV). CMMI for Development is not focused on software or systems engineering, and can be applied to almost every organization. The CMMI models that have been available until 2006 are now considered part of the CMMI for Development constellation.

Constellations are groupings of components used to construct models, and its training and appraisal materials. The “constellations” concept is part of the CMMI framework, used in CMMI v1.2 and beyond, and that accommodates additional content from the inputs of the users. The framework is organized to define model structure and terminology, and to draw on practices that have proven to be valuable in previous models, so that the CMMI models constructed have a familiar placement of model components, thus reducing the effort to write models, train users, and appraise organizations.

“So, [...] CMMI is growing beyond the star practices of the three original source models, and into constellations.” [1, page 18, “The Architecture of the CMMI Framework”]

Currently there are three constellations: CMMI for Services (not yet released), CMMI for Acquisition, and the one that concerns this project, CMMI for Development.

2.2.2 CMMI for Development

CMMI for Development (CMMI-DEV) v1.2 reflects the integration of the systems engineering and software engineering bodies of knowledge, providing an integrated solution for development and maintenance activities applied to products and services. It is a continuation and update of CMMI v1.1 that has been facilitated by the concept of CMMI “constellations”. CMMI-DEV is the first one, it represents the development area of interest and consists of two models: CMMI for Development + IPPD (with the IPPD group of additions) and CMMI for Development (without IPPD). This project was based on CMM for Development (without IPPD), by previous choice of the organization – Qimonda’s Porto Development Center.

CMMI for Development is the successor of the three source models: the SEI has retired SW-CMM and IPD-CMM, and EIA has retired the SECM. CMMI version 1.2 was developed using input from nearly 2000 change requests submitted by CMMI users [1, page 18].

CMMI for Development, produced in August 2006, consists of best practices that address practices that cover the product’s lifecycle from conception through delivery and maintenance, applied to products and services. The emphasis is on the work necessary to build and maintain the total product.

Organizations from many industries like aerospace, banking, computer hardware, software, defense, automobile manufacturing, and telecommunications, use CMMI for Development, for models in the CMMI-DEV constellation contain practices that cover project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes.

CONTINUOUS AND STAGED REPRESENTATIONS

CMMI has two different representations: continuous and staged, that basically differ on the approach to process improvement and appraisals. Both representations have different advantages and disadvantages (summed up in table 2.2), so some organizations use both of them at various times in their improvement programs.

The **continuous representation** allows the organization to select a specific process area (or group of process areas) to improve, improving the processes related to it. Capability levels characterize improvement of an individual process area. You can improve different processes at different rates, with maximum flexibility, although there are some limitations on the choices because of the dependencies among some process areas.

The **staged representation** defines an improvement path through predefined sets of process areas. Each maturity level provides a set of process areas that characterize different organizational behaviors. Through the staged representation,

the organization improves one stage at a time. Achieving each stage ensures a good foundation for the next stage. Process areas are organized by maturity levels, providing an order of implementing process areas, and an improvement path for an organization from the initial level to the optimizing level.

To choose a representation, an organization should consider three categories of factors: business – which business objectives you would like your process improvement program to support and how these business objectives align with the two representations; culture – if your organization has none experience/culture in process improvement, you may choose the staged representation, which provides additional guidance, but if your organization has that experience, you might select the continuous representation; legacy – if an organization has experience with another model that has a staged representation, it may be wise to continue with the staged representation when using CMMI; the same is true for the continuous representation.

Essentially, whether used for process improvement or appraisals, both representations offer equivalent results, so an organization does not need to select one representation over another

“Organizations that are successful in process improvement often define an improvement plan that focuses on the unique needs of that organization and therefore use the principles of both the staged and the continuous representations.” [1, page 25, “Why not both representations”]

Continuous Representation	Staged Representation
Grants explicit freedom to select the order of improvement that best meets the organization’s business objectives and mitigates the organization’s areas of risk. Enables increased visibility of the capability achieved in each individual process area	Enables organizations to have a predefined and proven improvement path Focuses on a set of processes that provide an organization with a specific capability that is characterized by each maturity level.
Allows improvements of different processes to be performed at different rates.	Summarizes process improvement results in a simple form - a single maturity-level number.
Reflects a newer approach that does not yet have the data to demonstrate its ties to return on investment	Builds on a relatively long history of use that includes case studies and data that demonstrate return on investment

Table 2.2: Comparative Advantages of Continuous and Staged Representations

[1, page 24] CAPABILITY AND MATURITY LEVELS

Levels in CMMI describe a path for an organization to improve the processes it uses to develop its products and provide its services. To reach a particular level, the

organization has to satisfy all the goals of the process area or set of process areas that were meant to be improved. Levels can be the result of an appraisal, as a rating, to assess the organizations processes improvement status. The two improvement paths, corresponding to the continuous and staged representations, are associated with two types of levels: capability levels for the continuous representation and maturity levels for the staged representation. I.e., the continuous representation focuses on process area capability, measured by capability levels, and the staged representation focuses on organization maturity, measured by maturity levels.

While there are six capability levels, from 0 through 5, that apply to process improvement achievement in individual process areas; there are only five maturity levels, from 1 through 5, that apply to process improvement achievement across multiple process areas. [1, page 45]

Level	Capability Levels	Maturity Levels
Level 0	Incomplete	N/A
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	Quantitatively Managed	Quantitatively Managed
Level 5	Optimizing	Optimizing

Table 2.3: Capability and Maturity Levels

The organization in which this project was developed – Qimonda’s Porto Development Center – had already started implementing CMMI and chosen the staged representation, for its extra guidance and similarity to rating levels, so that is the representation further explained from this point on. For more on Capability Levels (of the continuous representation) please refer to Appendix B.

MATURITY LEVELS

A maturity level consists of related specific and generic practices for a predefined set of process areas. Through the years, SEI has found that

“organizations do their best when they focus their process improvement efforts on a manageable number of process areas at a time” [1, page 52]

The maturity levels are measured by the achievement of the specific and generic goals associated with each predefined set of process areas.

Maturity Level 1: Initial At this level, processes are usually ad hoc and chaotic.

Success in these organizations depends on the competence and heroics of the people and not on the use of proven processes. They frequently exceed their budgets and do not meet their schedules, although they often produce products

and services that work. These organizations are characterized by a tendency to over-commit, abandonment of processes in a time of crisis, and an inability to repeat their successes.

Maturity Level 2: Managed At this level, the projects of the organization have ensured that processes are planned and executed in accordance with policy; employ skilled people with adequate resources to produce controlled outputs; involve relevant stakeholders; are monitored, controlled, and reviewed; and are evaluated for adherence to their process descriptions. The process discipline existent in this level helps to ensure that existing practices are retained during times of stress, and that projects are performed and managed according to their documented plans. The status of the work products and the delivery of services are visible to management at defined points (as major milestones).

Maturity Level 3: Defined At this level, processes are well characterized and understood, and are described in standards, procedures, tools, and methods. The organization's set of standard processes is established and improved over time. Projects establish their defined processes by tailoring the organization's set of standard processes according to tailoring guidelines. Processes are described more rigorously, clearly stating the purpose, inputs, entry criteria, activities, roles, measures, verification steps, outputs, and exit criteria. Processes are managed more proactively using an understanding of the interrelationships of the process activities and detailed measures of the process, its work products, and its services. Processes are typically qualitatively predictable.

Maturity Level 4: Quantitatively Managed At this level, both the organization and projects establish quantitative objectives for quality and process performance and use them as criteria in managing processes. These objectives are based on the needs of the customer, end users, organization, and process implementers. Quality and process performance is understood in statistical terms and is managed throughout the life of the processes. Quality and process performance measures are incorporated into the organization's measurement repository to support fact-based decision making. Special causes of process variation are identified and, where appropriate, the sources of special causes are corrected to prevent future occurrences. The performance of processes is controlled using statistical and other quantitative techniques, and is quantitatively predictable.

Maturity Level 5: Optimizing At this level, an organization continually improves its processes based on a quantitative understanding of the common

causes of variation inherent in processes. Focus on continually improving process performance through incremental and innovative process and technological improvements. Quantitative process improvement objectives for the organization are established, continually revised to reflect changing business objectives, and used as criteria in managing process improvement. The effects of deployed process improvements are measured and evaluated against the quantitative process improvement objectives.

Descriptions of Maturity Levels adapted from [1, pages 53–55]

As your organization achieves the generic and specific goals for the set of process areas in a maturity level, you are increasing your organizational maturity.

PROCESS AREAS

“A process area is a cluster of related practices in an area that, when implemented collectively, satisfy a set of goals considered important for making improvement in that area.” [1]

There are 22 process areas that can be organized in four categories – Process Management, Project Management, Engineering, and Support – or 4 maturity levels – from 2 through 5 (table 2.4).

In the scope of this project were the Engineering level 3 process areas, except the Requirements Development Process Area.

The processes from the engineering process areas apply to the development of any product or service in the engineering development domain, since they were written using general engineering terminology and support a product-oriented process improvement strategy.

As this project was part of an effort of the organization to improve its processes to achieve a level 3 maturity, the author of this project was responsible for implementing the new engineering process areas. By decision of upper management, the Requirements Development practices were integrated in the organization’s processes related to Requirements Management, thus leaving that particular process area out of the scope of this project.

In the scope of this project were the other four engineering process areas from maturity level 3.

Technical Solution – TS This process area practices include the study of alternative solutions and the establishment of criteria that may differ from product to product, so that the chosen solution will be the best solution according to those criteria.

Categories	Process Areas		Maturity Level
Process Management	OPF	Organizational Process Focus	3
	OPD	Organizational Process Definition	3
	OT	Organizational Training	3
	OPP	Organizational Process Performance	4
	OID	Organizational Innovation and Deployment	5
Project Management	PP	Project Planning	2
	PMC	Project Monitoring and Control	2
	SAM	Supplier Agreement Management	2
	IPM	Integrated Project Management	3
	RSKM	Risk Management	3
	QPM	Quantitative Project Management	4
Engineering	REQM	Requirements Management	2
	RD	Requirements Development	3
	TS	Technical Solution	3
	PI	Product Integration	3
	VER	Verification	3
	VAL	Validation	3
Support	CM	Configuration Management	2
	PPQA	Process and Product Quality Assurance	2
	MA	Measurement and Analysis	2
	DAR	Decision Analysis and Resolution	3
	CAR	Causal Analysis and Resolution	5

Table 2.4: Process Areas by Categories and its Maturity Levels

Product Integration – PI This process area contains the specific practices that will lead to choosing the best possible integration sequence, integrating product components, and delivering the product to the customer.

Verification – VER This process area ensures that selected work products meet the specified requirements. It selects work products and verification methods that will be used to verify work products against the requirements.

Validation – VAL This process area incrementally validates products against the customer’s needs. It includes validation of products, product components, selected intermediate work products, and processes.

Recursion and iteration of these processes enable the organization to assure the quality of the product to be delivered to the customer.

In CMMI-DEV, each process area, as a cluster of practices in that area, has specific goals and practices, but there are also generic goals and practices, that are called generic because the same goal or practice statement applies to multiple process areas.

GENERIC GOALS AND PRACTICES

A generic goal describes the characteristics that must be verified to assure the institutionalization of the processes that implement a process area. It is used in appraisals to determine whether a process area is or is not satisfied.

A generic practice is the description of an activity or step that is part in achieving the associated generic goal.

There are 5 Generic Goals (GG) and 17 Generic Practices (GP), but although all of them are used in the continuous representation, in the staged representation only Generic Goals 2 and 3 are used, along with their Generic Practices. To achieve maturity level 2, one uses the process areas at maturity level 2 as well as generic goal 2 and its generic practices. To achieve maturity level 3 one should not just use the process areas at level 3 and generic goal 3 for these process areas; you must return to level 2 process areas and apply generic goal 3 and its practices as well.

To see the Generic Goals and Practices that applied to the scope of this project, please refer to Appendix [C.1](#).

SPECIFIC GOALS AND PRACTICES

A specific goal describes the unique characteristics that must be verified to implement the process area. It is used in appraisals to help determine whether a process area is satisfied.

A specific practice is the description of an activity or step that is part in achieving the associated specific goal.

To see the Specific Goals and Practices that were within the scope of this project, please refer to Appendix [C.2](#).

2.2.3 Adopting CMMI

Every organization has some unique characteristics – from its size and its products to the organizational attitudes toward quality – that generate its competitive advantage. That should be addressed in organizational processes so that the advantage is applied to every project of the organization – not every organization needs the same set of processes.

Many organizations measure their progress by conducting an appraisal with the possibility of earning a maturity level rating or just to assess the compliance of its processes to CMMI.

Appraisals of organizations using a CMMI model focus on identifying improvement opportunities by comparing the organization's processes to CMMI best practices. The appraisal results are then used (by a process group, for example) to plan improvements for the organization. The appraisals for organizations using a CMMI model must conform to the requirements defined in the Appraisal Requirements for CMMI (ARC) document [9].

The SCAMPI appraisal methods are the methods generally used for conducting appraisals using CMMI models. SCAMPI A is the most rigorous method and the only that can result in a rating. SCAMPI B is performed on implemented practices, with options in the scope of the model or of the organization. SCAMPI C is the one with least depth of investigation, providing a wide range of options. For organizations that wish to appraise multiple functions or groups, one appraisal method can provide separate or combined results for multiple functions.

An initial set of courses is provided by the SEI and its Partners, so an organization can better guide its processes improvement project, like *Introduction to CMMI* – provides a basic overview of the CMMI models – and *Intermediate Concepts of CMMI* – to those who plan to become more deeply involved in CMMI adoption or appraisal.

It was with training on the first course that process improvement on Qimonda's Porto Development Center started, a couple of years ago.

2.3 Qimonda Porto Development Center's Quality Management System (QMS)

2.3.1 Organizational Environment

Qimonda AG carved out of Infineon Technologies AG on May 1st, 2006, and is now a leading global memory supplier, headquartered in Munich (Germany)), with approximately 13.500 employees in nine sites on three continents: in Europe – Dresden and Dresden 300 (Germany), Porto (Portugal); in Asia – SMIC and Suzhou (China), Winbond and Inotera (Taiwan), Mallaca (Malaysia); and in America – Richmond (USA). Qimonda provides DRAM products for a wide variety of applications, including computing, infrastructure, graphics, mobile and consumer areas.

The Porto Development Center is an organizational unit belonging to the Qimonda Information Technologies Product Automation (IT PA).

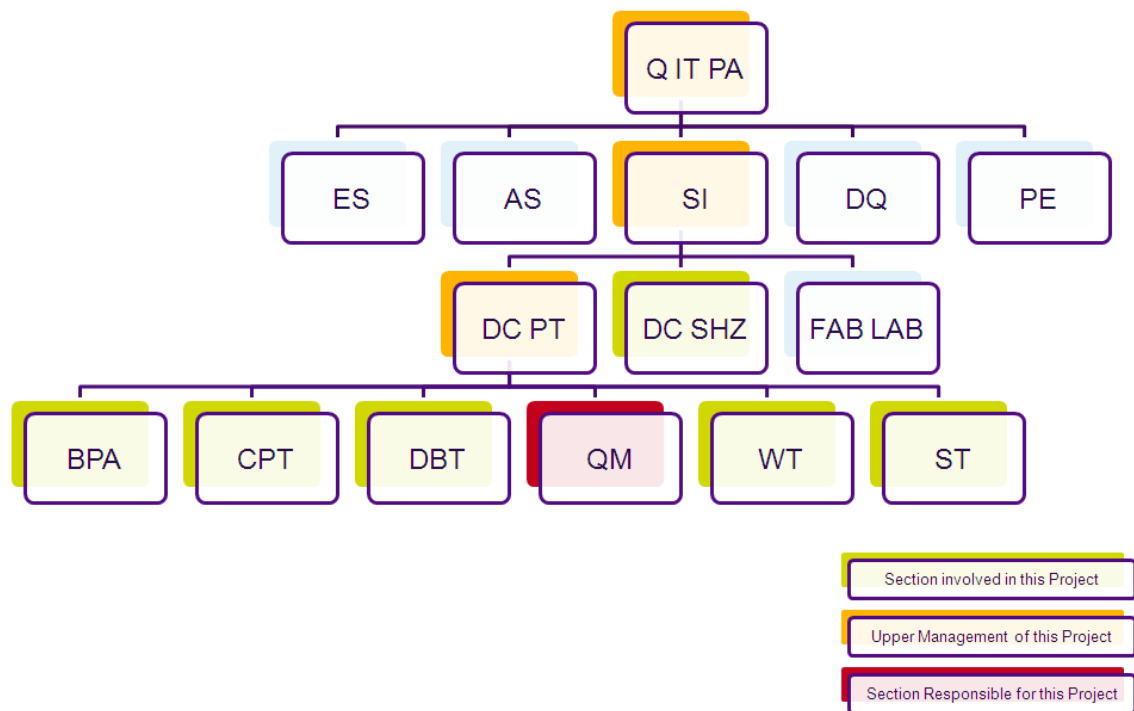


Figure 2.1: Qimonda IT PA organization

PORTO DC (DC PT)

The DC owns the implementation phase within the IT PA, receiving requirements, business specifications and may be high level design from the Domain Function¹. The DC is responsible for the detailed design, implementation, testing and integration of solutions including the release to the “customer”, the Domain Functions.

¹A Domain Function is an expert in the business area, closer to the end user.

The Porto Development Center is organized into six technology oriented sections:

Business Process Automation – BPA Responsible for the automation of business process providing development and support services.

Cross Platform Technologies – CPT Responsible for the development of middleware and web technologies, essentially focused on multi platform support.

DataBase Technologies – DBT Responsible for the Oracle databases and tools: Data Warehouse (as a way for the organization to effectively use digital information for business planning and decision making), and Engineering data analysis Base System (to access and analyze manufacturing data across the globe by using one central solution)

Quality Management – QM Responsible for Product and Process Quality Assurance in all the projects of the DC: artifact testing, Quality Management System and process activities, training, technical activities and Management.

Windows Technologies – WT Responsible for the development of Windows OS based Systems, Frameworks and GUIs;

Software Technology – ST Responsible for the state of the technology and the practice of software architecture in the Porto DC, continuously increasing its competitiveness for the benefit of its customers.

[10]

Porto DC aims to fulfill the functional and non-functional customer requirements at adequate quality levels, within the agreed schedules and within the agreed budget. To prosecute its mission, the Porto DC focus on software solutions delivery, commitment to quality, endeavor to state-of-the-art knowledge, commitment to continuous improvement, and uses the Quality Management System as a tool to meet its objectives.

2.3.2 Quality Management System

The software quality improvement in Porto DC started as a project in May 2006, followed by the release of QMS 1.0 in September 2006. QMS 1.0 focused on the CMMI level 2 process areas – SDP, PM, CM, PPQA, and REQM. In June 2007 QMS 1.0.1 was release with MA process area and in October 2007 was released the QMS 1.1.1, the version currently in use.

In December 2007 SEI performed a SCAMPI B Appraisal and in January 2008 the QMS 2.0 project started, focusing on the CMMI level 3 areas. Training for all

the organization in QMS 2.0 is planned to be hold from mid August on, so the QMS 2.0 can be released and implemented throughout all the Porto DC projects.

In order to fulfill the Porto DC objectives, QMS aims to provide products and services that meet or exceed the customer functional and non-functional requirements on the agreed timeframes, within the agreed budget, with high quality and using the agreed tools and software development processes. It also aims to achieve a continuous improvement through regular reviews of the performance of the solutions and services against measurable targets.

QMS 1.1.1 STRUCTURE

QMS 1.1.1 is the QMS currently in use in the Porto DC. It was structured in three main categories of processes: primary, support and organizational processes. Within the organizational there are the management processes.

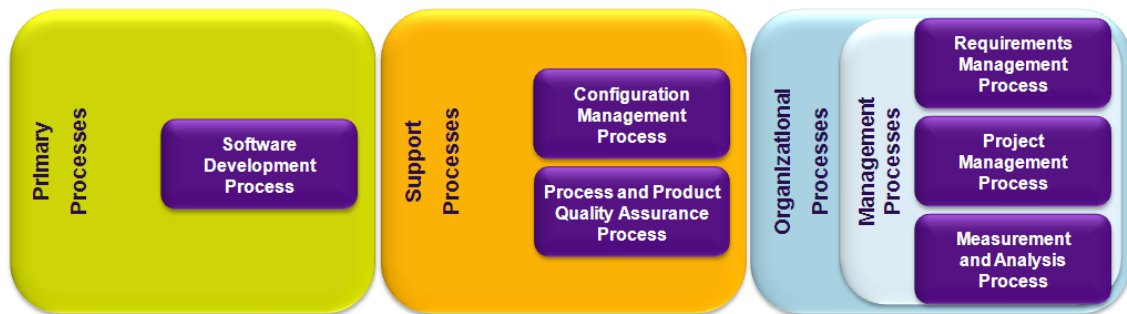


Figure 2.2: QMS 1.1.1 Structure

Each process may have one or more procedures, guidelines and templates associated with each of its activities. There are a few roles in QMS 1.1.1, but the main roles are the ones responsible for each defined process: Configuration Manager, Software Quality Assurance Engineer, and the Project Manager, along with the ones at the level of the organization as the DC Manager and the Quality Manager.

QMS is currently published in the QMS website, within the qShare portal, as a set of text and spreadsheet documents organized in a series of levels:

Level 0 — Organizational level descriptions

Level 1 — Process overviews (comprises the Software Development Process)

Level 2 — Quality Management System process descriptions (e.g. Configuration Management)

Level 3 — Procedures defining activities

Level 4 — Detailed tasks descriptions (performed to accomplish an activity – guide-books, checklists, among others)

There is a Local Software Engineering Process Group (Local SEPG) with representatives of all DC sections, to collect improvement suggestions and assure the

regular update of the QMS processes and documents.

2.4 Porto DC and CMMI

The QMS 1.1.1 processes can be mapped to the process areas in CMMI 1.2

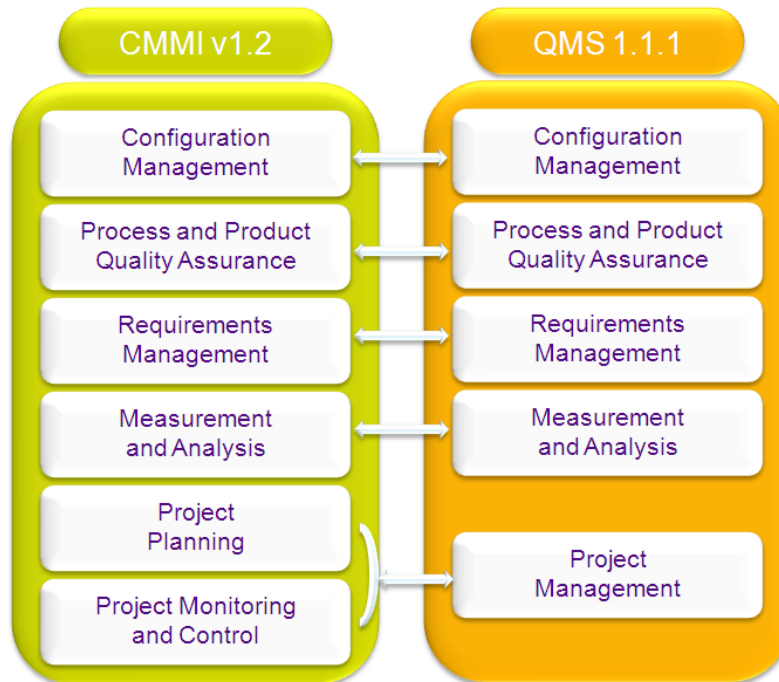


Figure 2.3: Mapping CMMI v1.2 – QMS 1.1.1

The Configuration Management Process Area maps to the Configuration Management Process, the Process and Product Quality Assurance Process Area maps to the Process and Product Quality Assurance Process, The Requirements Management Process Area maps to the Requirements Management Process, the Measurement and Analysis Process Area maps to the Measurement and Analysis Process, and the Project Planning and Project Monitoring and Control Process Areas map to the Project Management Process.

To ensure QMS 1.1.1 was mapping completely to maturity level 2 of the CMMI 1.2, Porto DC asked SEI for a SCAMPI B Appraisal, not only to identify the gaps to CMMI 1.2 Level 2, but also the gaps to CMMI Level 3, in order to create the basis to a road map toward the Level 3 certification.

THE SCAMPI B APPRAISAL

The SCAMPI B appraisal was performed in December 2007 by Colin Benton, the scope being the maturity level 2 and 3 Process Areas. The appraisal findings can be found summed up in [Appendix D](#).

It was time for some more process improvement.

A local Software Engineering Process Group was formed, with at least one representative of each Porto DC Section, to lead the process improvement project toward CMMI Maturity Level 3. The QMS 2.0 project aimed to close the gaps in current level 2 processes and to define the new level 3 processes.

The implementation was of the responsibility of a small team from the QM section, and the local SEPG contributed with process improvement suggestions, analyzed the major improvement suggestions, proposed solutions to address the gaps, and specially contributed with feedback from their sections to the processes definitions and interfaced between the QMS and the DC sections.

The road map for the QMS 2.0 project, with the goal to have the QMS of Porto DC compliant with CMMI Level 3, was then established.

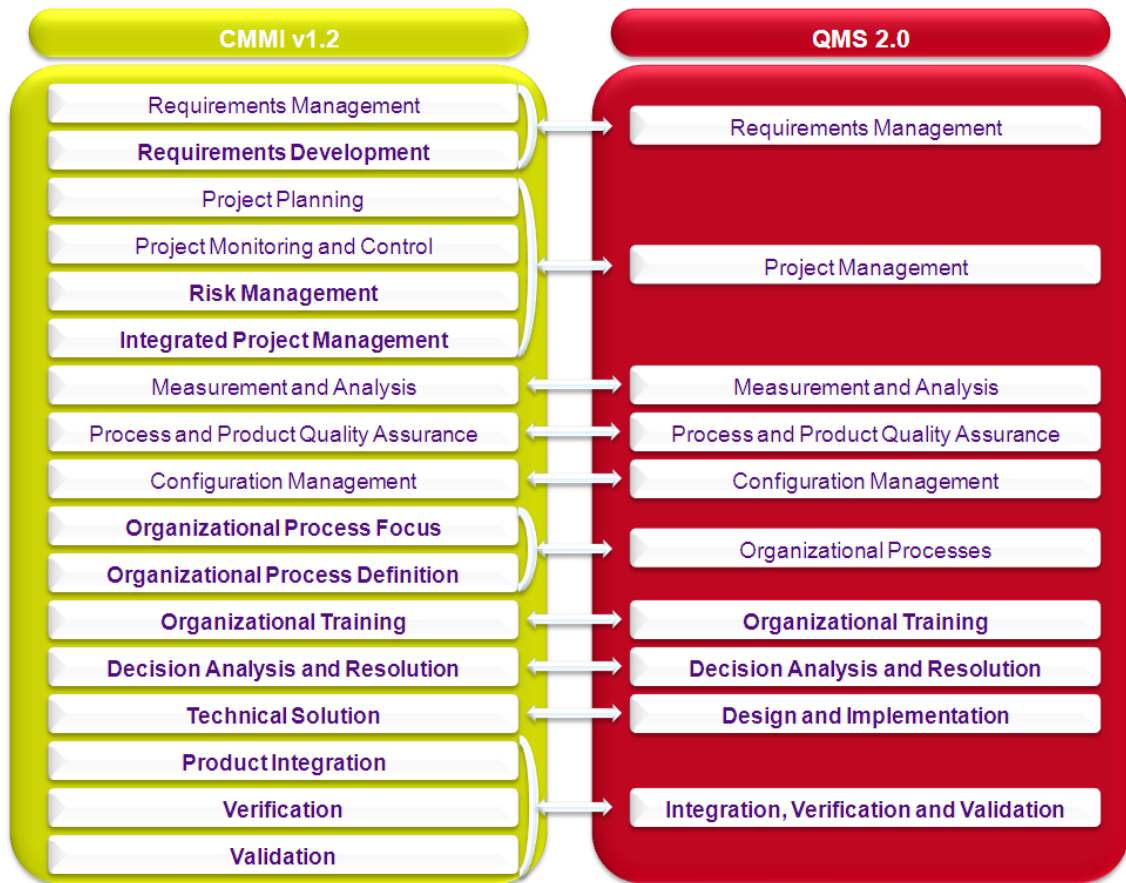


Figure 2.4: Mapping CMMI v1.2 – QMS 2.0

Qimonda has another Development Center, SuZhou DC in China, so a global SEPG put together the efforts of both Development Centers so both could aim for a CMMI Maturity Level 3 certification. Porto DC had already an appraised QMS, so it focused its efforts on enhancing the level 2 processes to close the identified

gaps and satisfy the Generic Goal of Maturity Level 3 – GG3 (Requirements Management, Software Development Process, Configuration Management, Measurement and Analysis, Process and Product Quality Assurance, and Project management). SuZhou focused its efforts on defining the new processes (Design and Implementation, Integration Verification and Validation, Organizational Processes, Organizational Training, and Decision Analysis and Resolution). To correct a deviation on the QMS 2.0 project timelines, the responsibility of Integration, Verification and Validation process definition was transferred to Porto DC.

2.5 Conclusions

As stated in the beginning of this chapter, the quality of a system or product is highly influenced by the quality of the process used to develop and maintain it, so the focus on processes enables maximization of people and technology productivity. It has been proven that process implementation reduces cost and schedule and increases productivity, quality and customer satisfaction.

CMMI is a process model that can be used to assess the implementation of processes in an organization and/or to provide guidance for improvement. When it was available as a technical report, SPICE could be considered an alternative to CMMI but the latest was the obvious choice to the organization in which this project was developed, for its availability, background and popularity.

CMMI for Development is one of the constellations of the CMMI framework, and the one that contains the CMMI for Development model used in this project. From the two possible representations – continuous and staged – the organization chose the staged one, since this one provided more guidance to the process improvement effort. This representation focuses on organization maturity, measured by maturity levels. The organization in which this project was developed aims for a level 3 maturity: at this level, processes are well characterized, understood and fully defined.

This project focused on the Engineering Process Areas: Technical Solution, Product Integration, Verification and Validation. Qimonda's process improvement effort toward maturity level 3 began with a SCAMPI B appraisal that identified the existing gaps and helped established a process improvement road map.

The Porto DC is working with the SuZhou DC to establish a Quality Management System compliant with CMMI level 3. In order to do that, it was established, besides a global Software Engineering Process Group (SEPG) to coordinate the efforts, a local SEPG with representatives of all the Porto DC Sections.

The QMS 2.0 project aims to close the gaps in current level 2 processes and to define the new level 3 processes. The Porto DC, that had the appraised version of QMS in place was to improve the level 2 processes while the SuZhou DC was to

State of the Art

create the new level 3 ones. Due to a schedule deviation, the responsibility of the PI, VER and VAL process areas was transferred from SuZhou DC to Porto DC.

State of the Art

Chapter 3

Problem and Current Situation Analysis

3.1 The Problem

Porto Development Center, already using a process framework built to cover CMMI Maturity Level 2 practices, had planned a process improvement effort so the Quality Management System in place would cover the Maturity Level 3 practices. There were enhancements to be done within the processes already in place, but there were also new processes to be implemented, namely to cover the Engineering Process Areas: Technical Solution, Product Integration, Verification, and Validation. The Technical Solution process area was going to be covered by the *Design and Implementation* process, responsibility of the SuZhou DC, and the last three process areas were going to be covered by the *Integration, Verification and Validation* process, responsibility of the Porto DC.

In the scope of this project are the processes that correspond to the engineering process areas, but with two different approaches: the Integration, Verification and Validation process was to be defined from scratch whereas the Design and Implementation, as it was going to be defined by SuZhou DC, was to be verified in terms of compliance with CMMI maturity level 3 and with Porto DC's reality.

The main goal of this project was to define the Integration, Verification and Validation process (IVV), to be compliant with the CMMI Maturity Level 3 Generic Goals and Practices, with the Specific Goals and Practices of the Product Integration, Verification and Validation process areas and also to reflect the Porto DC reality. It was also a project goal to monitor the Design and Implementation process definition, trying to make sure it was also CMMI Maturity Level 3 compliant and

adapted to the current practices in use in Porto DC. In order to define the IVV process it was also expected the definition of procedures, guidelines and templates to support the implementation details of the process.

Design, Implementation, Integration, Verification and Validation were already being performed in Porto DC, but not in a mindful way: most of the corresponding activities were being performed but were not defined, were not being performed consistently across the organization, and the established best practices in these areas were not being used. Some project teams within the same area, section or technology had some sort of best practices that passed down to team members as new people entered the team. However, the organization as a whole was not aware of these practices.

The said purpose of this project was to help Porto DC improve its QMS to achieve Maturity Level 3 compliance, focusing on the Engineering Process Areas, thus meaning this project should assess the current practices within these areas and define the processes to reflect the current practices with enhancements to embody the level 3 of CMMI.

3.2 The Approach

In order to better fulfill the purpose of this project, an analysis of the CMMI engineering process areas, along with the study of the QMS in place was in order. After a combined analysis of CMMI, QMS and of the current practices in Porto DC, a new process was devised, along with all its support documents (e.g., procedures, guidelines, templates). A parallel study was conducted in order to analyze whether or not the organization should change the publishing of the QMS from a set of documents available in the intranet to a more user-friendly framework.

This project started with the study of the CMMI v1.2, focused on the Engineering Process Areas. The author presented the engineering goals and practices to the local SEPG, focusing on the specific practices and work products. This presentation could then be a basis to establish a road map to define the QMS processes that would implement the practices of the Engineering Process Areas.

With this in mind, a study was conducted to identify the CMMI level 3 practices already existent within the Quality Management System in place. None of the design activities were featured in QMS, as it was assumed that the direct customer (Domain Function) provided the design. When that would not suffice, the developer would implement the solution in the way he felt was better. The implementation activities were not reflected in the QMS in place, but almost all the sections had best practices, code guidelines and other documentation, to ease the understanding of previously developed components. The integration activities were not really planned

Problem and Current Situation Analysis

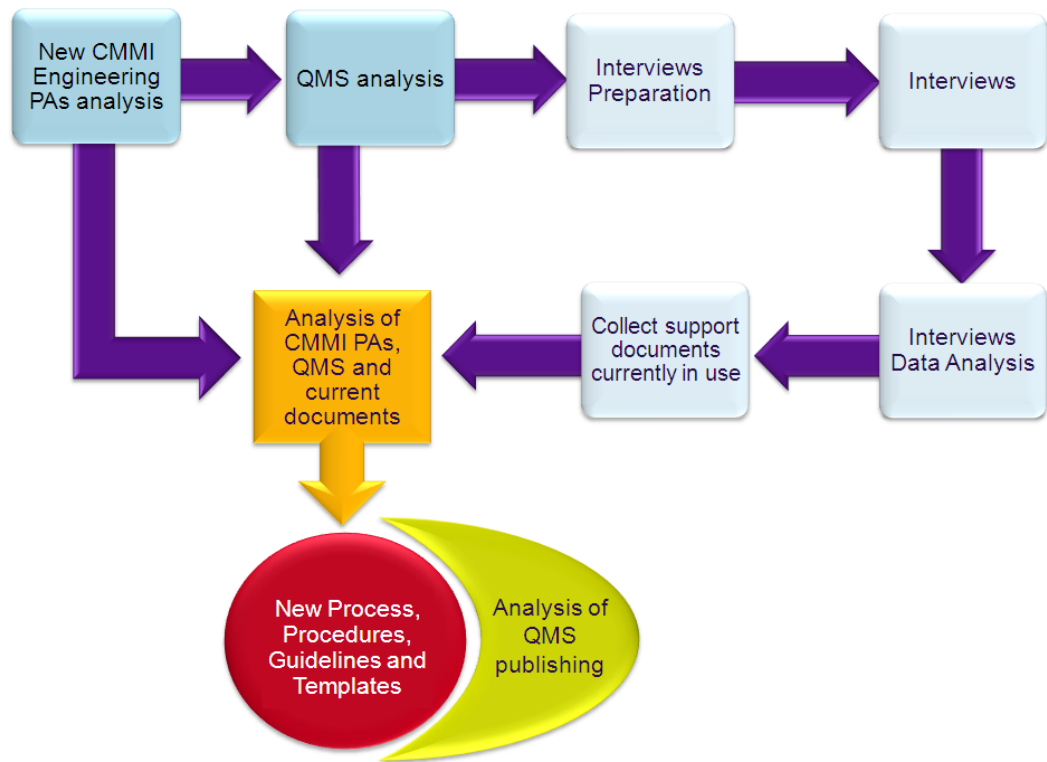


Figure 3.1: Method followed in the approach to the problem

before they were carried out, as the project team concern was to “make the product work” from the very beginning; products that required the involvement of more than one DC section would, in general, only be integrated before the testing phase. Some of the verification and of the validation activities were respectively being done as part of the Process and Product Quality Assurance (through a test plan, test logs and test reports) and of the Requirements Management processes (validation of the requirements). The validation of the product could be difficult since usually the Porto DC customer is not the end-user, although the DF would provide user acceptance test cases in order to ease this task.

After the study of the QMS, there was still a need to better understand the current practices in Porto DC, so some interviews with people directly involved in the projects were planned. The section representative members of the Porto DC SEPG appointed people that were involved in projects that could be used as a sample of each section’s work.

While planning for the interviews, a set of questions based on CMMI practices was compiled to guide the data collection. The set of questions covered some design, implementation, integration, verification and validation specifics, as to how the requirements or the design arrived, if the team designed the solution and kept traceability from the requirements to the code (and vice-versa), if there was interface

Problem and Current Situation Analysis

definition to ease the integration, if the integration was planned, if the environment in which the verification and validation activities took place was thought through, how could the team assure the customer would be satisfied with the solution they developed, among others. The questions were devised to serve as an interview guide, so they were open questions, and there were only 20 questions. The interviewers were free to ask more questions besides the ones planned, if they felt there was need to clarify some aspects of the answers. Each interview was made by two interviewers (one was the author of this project) and was planned to last about half an hour. Some took twice as much.

Number	Question
1	How do you obtain the requirements/requests?
2	How do you define how you will implement a request/requirement? Why?
3	How do you make sure your code implements all the requirements / requests? And how do you know what part of the code implements each requirement / request?
4	Do you design the solution? Always in the same way? In what way?
5	How do you define the interaction between every product component? (interfaces)
6	How and when are those components integrated? Is this planned / defined?
7	Are the components tested before and after the integration? How?
8	The developers test what they implement (at component level)?
9	What kind of verifications do you perform? (e.g., tests, code reviews) When? To what? Is there a procedure/criteria?
10	How do you know the verifications were successful?
11	Is there need for any kind of preparation/planning to perform those verifications? (e.g., hardware, software, rooms)
12	The client defines the test cases? For the product as a whole or just for some components?
13	Is there any (other) way to assure the customer satisfaction?
14	How do you know your customer was satisfied with the product?
15	What if he isn't satisfied? What do you do?
16	What characteristics of the solution does the customer value more?
17	The test cases defined by the customer are executed in an environment similar to the production environment?
18	Do you use prototyping in the early stages of the project to validate the requirements/design?
19	What goes on record (what stays in the organization after the project ends?)
20	Suggestions/Comments to the QMS and the Porto DC practices?

Table 3.1: Interviews Questions

There were 12 interviews, with a total of 16 interviewees: 2 from DBT, 4 from

BPA, 2 from WT, 2 from CPT and 6 from QM. There were none from ST since at that time ST was not directly involved with Porto DC's projects. Today ST has exclusivity on the new role of "Design Advisor" in each project.

After the analysis of the data from the interviews, each interviewee was asked to send, to the author of this project, an example of each support document he had mentioned in the interview. Afterwards, an analysis was conducted on the documents sent, to identify the differences between the documents and the existent QMS templates, and among documents that should be similar in content. This analysis was conducted also bearing in mind the previous analysis – the CMMI Engineering Process Areas analysis and the QMS analysis. Some quick meetings with the senders of the documents took place in order to understand the most significant differences of the content of documents that should be similar, so to incorporate the needed changes in the new process.

The new process "Integration, Verification and Validation" was then defined, along with the Release procedure, the Release Notes templates, the Release Mail guidelines and template, the Integration, Verification and Validation Plan template, the Test Report template, and the Release Checklists. The Inspection, Email-pass-around Review and Over the Shoulder Review procedures already defined were integrated in this process. At the same time, as the "Design and Implementation" process was becoming available, it would be reviewed based on the analysis performed in order to identify gaps towards CMMI level 3 and the current practices of Porto DC.

Simultaneously, a study to the publishing of QMS 1.1.1 was conducted. QMS 1.1.1 was available in the internal website as a set of documents, text and spreadsheets, but there was a framework – the WebViewer, from João Cortez – that could be used to ease the navigation through all the QMS information. However, its implementation was never really studied. Within this project, the advantages and disadvantages of the framework versus the set of documents were analyzed, along with the effort needed to migrate the QMS 2.0 to the framework (since it was already in a set of documents).

3.3 Conclusions

Porto DC had a process framework in place, built to cover CMMI Maturity Level 2 practices but had planned a process improvement effort so the QMS would cover the Maturity Level 3 practices. Besides the enhancements that were to be done to the existent processes, there were new processes to create, and among them the Design and Implementation process (DI) – to cover the Technical Solution process

area – and the Integration, Verification and Validation process (IVV), to cover the Product Integration, the Verification and the Validation process areas.

This project goals were to successfully define the IVV process and related support documentation, and to help SuZhou DC with the DI process – by reviewing the process definition and trying to make sure it was CMMI level 3 compliant and also adapted to Porto DC reality.

Design, Implementation, Integration, Verification, and Validation were already being performed in Porto DC but not in a mindful way, so, besides studying the mentioned CMMI process areas and the QMS in place, it was necessary to assess the current practices in Porto DC. This was done through a series of interviews with project team members. From the effort of analyzing the CMMI PAs, the QMS in place and the current practices in use, was defined the new IVV process.

Simultaneously a study was performed to analyze an alternative to the way QMS was currently published. It is published as a set of documents but there was a framework available that could improve the navigability of the QMS information, enabling the project team member to consult the QMS on a “as needed” basis.

Chapter 4

The New QMS Engineering Processes

4.1 Solution Development

As presented in the previous chapter, besides the Integration, Verification and Validation process, there were other documents, support documents, that were produced within the scope of this process. The Integration, Verification and Validation Plan was completely new and produced from scratch, but the Release Notes template, the Test Report template and the Release Mail were already part of the QMS in place. In order to improve these templates and guidelines not just to meet CMMI Maturity Level 3 requirements but also to better portrait the current practices in Porto DC, the examples sent by the interviewees were compared to the existent template and quick meetings were held to understand the reason of the differences. Finally the conclusions from the differences of the documents were compiled and from that knowledge base, a new template was created.

4.1.1 Release Notes

The Release Notes example documents were compared to the QMS 1.1.1 template (that can be found in [appendix E](#)) and to a content list of the QMS 2.0. Most of the notes were taken from the quick meetings held to understand the differences.

1. Items included in the release
2. Change requests per item
3. Items version

4. Other info according to the projects complexity and needs

PROJECT: BPA – MASTER DATA

In the second chapter the Master Data project has created three sections – 2.1 Modules, 2.2 Scripts SQL, and 2.3 Workflows.

In the third chapter, this project has created two sub-chapters – 3.1 MasterData component functional changes and 3.2 MDS GUI Functional Changes, and modified the table.

Artifact Name	Functionality (Added/Changed/Removed)	UTP (if applicable)
<artifact name>	<LDAP Authentication functionality added>	<151872>

Figure 4.1: Table from Chapter 3 of the Release Notes template

The Master Data project team did not use this table because it was too complicated to map the UTPs to the functionalities, specially when the one creating the release notes is from QM and is not so familiar with the code. The interviewee said “it would be a day’s work and it would be necessary to bother many people”.

So, instead, they use the following tables.

Functionality	CR Type
F30000121233 - Copy of experiment has original as reference in CS	Code Bug
F30000120812 - Please remove automatic Consistency Check of Integration CR	Design Bug

Figure 4.2: Master Data section 3.1 table

Functionality (Added/Changed/Removed)	Scope			CR Type
	GUI	Administration	Internal	
F30000120986 - Error message during a selection of a step	✓			Code Bug
F30000120729 - No possible to integrate PCR B CRs with e...	✓			Design Bug

Figure 4.3: Master Data section 3.2 table

Finally, the Master Data project team added a new chapter, Chapter 10 - Configurations, which comprised: 10.1 ClearCase specifications, with 10.1.1 QCIM Master Data Integration: table with Project Name, VOB, ClearCase Components and Stream, and 10.1.2 QCIM Master Data ClearCase Components Content: tables, each one with title (component name) and content (brief description of each component content); and 10.2 Master Data Component Installation with 10.2.1 Installation Guide.

And this project team’s release notes were compliant with the QMS 2.0:

1. yes, in chapter 2
2. yes, in chapter 3
3. yes, in chapter 2
4. yes, in chapters 8 (Compatibility) and 10 (Configurations)

PROJECT: BPA – COMPONENTS

In the second chapter, the table was modified.

Artifact Name	Version	N – New E – Enhanced F – Fixed
<artifact name>	<1.4>	</>

Figure 4.4: Table from Chapter 2 of the Release Notes template

Work Product Name	Version	Changed (Yes/No)
Qimonda CRE	3.1.0	Yes
DebugCRERules (Microsoft Visual Studio Project)	3.1.0	Yes

Figure 4.5: Components chapter 2 table

The only difference is the last column, so they agreed they could use the New / Enhanced / Fixed if they could use as many as they would like for each “Artifact/WorkProduct” (i.e., an artifact may be enhanced and fixed at the same time).

In the third chapter, this project team also modified the table. The table on the template was the table 4.1.

Work Product Name	Functionality (Added/Changed/Removed)
CRE Engine	Removed the EvaluateScenario API. The scenarios are evaluated by calling the EvaluateRule. The EvaluateRule both evaluate rules and scenarios.
CRE Engine	Added the flag RuleTrace to activate the publishing messages to Yoda.

Figure 4.6: Components chapter 3 table

The UTPs are presented only in the Solved Problems chapter, because the team feels there is no interest in mapping the UTPs to the functionalities “it is complicated, and a different detail level”.

The team changed the content of the fourth chapter – Restrictions – although the name of the chapter was kept. The content with which the team filled this chapter was similar to the content of MasterData’s tenth chapter – Configurations.

When offered this option, the team agreed that the information that was being included in this fourth chapter should be in a chapter called Configurations.

In the sixth chapter, the team modified the table and created a section – 6.1 Test Results Summary.

UTP (if applicable)	Description
<115288>	<Bug calculating the final grade.>

Figure 4.7: Table from Chapter 6 of the Release Notes template

PR/CR	Id	Description
PR	3	Increase the size of rule editor on CRE GUI 3.0.1
PR	4	The wildcard '*' doesn't work on field Scenario Name
CR	5	Add maps for the <u>EvaluateRule</u> CRE method

Figure 4.8: Components chapter 6 table

With this change, Id can be the UTP number or an id from a list of requirements, and PR stands for Problem Reported while CR stands for Change Request.

The new section contains a table with the test results summary.

6.1 Test Results Summary

Description	Total	%
Unit Test Cases	44	100.00%
Unit Tests Executed	44	100.00%
Unit Tests Succeeded	44	100.00%
Unit Tests Failed	0	0.00%
Unit Tests Inconclusive	0	0.00%
Unit Test Coverage	Not Available	Not Available

Table 5 – Test Result Summary

Please find the TestResults logs for this release at the CRE QShare page.

Figure 4.9: Components section 6.1 table

The team feels that there should be a chapter just for the test results summary.

The name of chapter 7 was changed from Known Problems to Known Problems and Enhancements, and a table was created to display the content of the chapter.

7. Known problems and Enhancements

PR/CR	Id	Component	Description
CR	2	CRE Engine	Creation and maintenance of CRE rules should be secure and controlled.
PR	21	CRE GUI	Copy / Paste inside a rule tend to insert many blank lines. Pasting multiple lines of code puts the code all on one line.
CR	23	CRE Engine	Implement integration to the Data Integration Data Sink from CRE
CR	24	calling patterns	Add the ability to Parameterize Rules within CRE
CR	28	CRE Engine	Access to FAB300 Parameter Object from within CRE Rule.
CR	29	CRE Engine	Ability to access Entity Sub-Components within CRE Rule.

Table 6 - Known problems and Enhancements

These issues and enhancements will be fixed and implemented on next releases.

Figure 4.10: Components chapter 7 table

Actually this was not the intent of this chapter. This chapter indicates to the customer – or whatever stakeholder receives the release notes – that there are some known problems with the product being released, that will be resolved in a near future. Enhancements are out of the scope of the present release, so they should not be mentioned in the release notes.

This project team’s release notes were compliant with the QMS 2.0:

1. yes, but sometimes per package, sometimes deeper (per module, workproduct or something)
2. yes, per package
3. yes, per module or workproduct
4. yes, in chapter 4 (Restrictions but should be “Configurations”), in chapter 6 (solved problems), in chapter 7 (known problems and enhancements) and in chapter 8 (Compatibility)

PROJECT: BPA – WORKFLOWS

The release notes example from this team does not follow the QMS template at all, because the person who created the release notes was in a hurry to create them and when he opened the QMS template he got scared with all the sections and no immediate understanding, so he searched for a simpler one. That is how an old component’s release notes was made into a template and from that point on every version of this workflow that is released goes with this kind of release notes.

There is so much information missing that there was no point on comparing these release notes with the QMS template.

1. yes, but only the component (really high-level view)
2. not really, it just mentions the UTPs that were addressed (not really what was addressed)
3. yes, per DLL
4. not really, just a bit of an installation manual

PROJECT: CPT – YODA

This project team uses the Release Sign-Off Form template instead of the Release Notes one, so there was no point on comparing the example document to the QMS template.

1. yes, in the Major Changes section
2. yes, in the appendix (requirements, improvements and bug fixes sections)
3. yes, in the overview of each appendix
4. yes, in the appendixes

PROJECT: QM – PROFAB – USER TEST SCENARIOS

In the second chapter, besides a general description there are Clear Case configurations. Once again is felt the need of a Configurations chapter.

In the fourth chapter, section 4.1, in the template the restrictions were per work-products but in this example the restrictions are presented per type (e.g., Pre, Post).

In the sixth chapter the content was modified for instead of referring to the solved problems, this chapter states what was implemented.

These release notes contents were not fully compliant with the contents expected in the QMS 2.0.

1. yes, in the General Description Chapter
2. no (don't know if it's applicable)
3. no, it says NA
4. no

PROJECT: QM – PROFAB

This team's example was a spreadsheet and not a text document as the template, so there was no point on comparing it to the QMS template.

These release notes contents were not fully compliant with the contents expected in the QMS 2.0.

1. yes, in the FrontPage per Work Package and in the Overview per Artifact
2. not really, just the references to the UTP addressed (per Artifact) (in the overview sheet)
3. yes, per artifact (in the overview sheet)
4. yes, in the sheets clearcase configuration, test results, and code coverage

PROJECT: WT

This section proposed a new template, with some chapters removed and other added.

Chapter 2 was changed from General Description to Changes in this Release.

2. Changes in this release

Changes	Scope			Issue Type
	GUI	Administration	Internal	
F30000120360 - Remove <ALL> from <u>LotsAtStep</u> and <u>ToSite</u> in <u>RemoteShipGUI</u>	✓			Enhancement
F30000119972 - Material Traceability	✓			New
F30000119141 - BEMESGUI--Batch function of adjust lot	✓			Enhancement
F30000120335 - Subject subscription error in SZV when BEGUI starts			✓	<u>CodeBug</u>

Figure 4.11: WT chapter 2 table

Each section of chapter 2 matches the reference in each line of the table [4.11](#).

Chapter 3 was changed from Functional Changes to Changed Dependencies, but most of the content are configurations. Each sub-chapter matches a work-package.

Chapter 4 was changed into Language Support, chapter 5 is the Known Problems chapter, with a section further explaining each problem.

Chapter 6 was changed into Compatibility but it actually contains a description of the external dependencies of the product being released.

6. Compatibility

The dependencies for the current version of BEMESGUI are:

Version	Name
2.6	MAC GUI Framework
4.5	YODA

Figure 4.12: WT chapter 6 table

The template proposed does not fully comply with the intended contents for the release notes of the QMS 2.0.

1. not really, only the changes done
2. not really, only changed dependencies per item
3. The items version is the release's version
4. yes, in the changes in this release, changed dependencies, known problems, and compatibility chapters

PROJECT: WT – MAC

In the third chapter the used table does not mention if the functionality was added, changed or removed, and, according to the team, if there are 30 or 40 artifacts affected, it is complicated to read the table; added/changed/removed does not belong to the description of the functionality but if it was a column where you could see all the added, all the changed, all the removed, then it would be helpful. In chapter 3 sections the UTPs are mentioned in the table of the functionalities changed.

In the sixth chapter, the sections mention the UTPs in the table of the solved problems. This team suggested to merge the chapters Functionality Changes and Solved Problems into one chapter, with a table with a column to indicate the kind of change. The team also complains that at the beginning of the document the Scope and the Audience are always the same so they could be omitted.

This team also mentioned the new template proposal from the WT section.

These release notes example was compliant with QMS 2.0

1. yes, in the general description chapter
2. yes, in the functional changes chapter
3. yes, in the general description chapter
4. yes, in the solved problems chapter

4.1.2 Release Mail

Although the release mail was sent in every project release to the stakeholders of the project, including the customer, it was not really defined in QMS 1.1.1 so this was more a gathering of needed mail content instead of a comparison.

The From field was always filled with DC QM Support (QPT IT MFG DC QM) to the Distribution List of the project (e.g., DL-IT-MFG-MES-FAB300-Release, DL-QPT-IT-PA-MES-MasterData-Release).

The Subject field was always filled with the Release Name (e.g., ProFAB 5.1 BETA2) and the body of the mail differed from release mail to release mail, so there was a gathering of opinions and the new Release Mail template reflects these opinions. The body field always ended with the QM Support Signature:

Best Regards,
DC QM Support
QPT IT PA DC QM
Porto Development Center
Qimonda Portugal S.A.
Mailto:DcQmSupport.R-POR@qimonda.com
Visit us at http://goto.qimonda.com/porto_dc

4.1.3 Release Checklists

It was felt the need to gather some “how to” information in the form of Release Checklists, in order to guide the person that was actually doing the release. As it was supposed to be as detailed as possible, each checklist was very specific, so the section representatives in the local SEPG were in charge of the collection – and sometimes, creation – of these release checklists. This project’s author provided an example file based on existing checklists, and the section representatives added their checklists to that base spreadsheet. The Release checklists file can be found in Appendix [F](#).

4.2 The New Processes and Support Documents

The new processes, Design and Implementation, and Integration, Verification and Validation, as well as related support documents, are presented as they were defined in the QMS 2.0 documents.

4.2.1 Design and Implementation

1. Introduction

1.1 Overview

This document details the Design and Implementation process used by Qimonda Development Centers.

1.2 Purpose

The purpose of this process is to define how to design, develop, and implement solutions to requirements.

1.3 Audience

The audience of this document is all project managers, design advisor, lead developer/developer, testers and other members that participate in QMS compliant projects developed within the organization in one of the roles referred in [AD-3].

The content of this document is property of Qimonda AG and distribution to external entities (such as customer) is not allowed unless there is a written authorization from the Global SEPG.

1.4 Definitions and acronyms

Please find the full description on the glossary [AD-2].

1.4.1 Acronyms For the purpose of this document the following acronyms are applicable:

AD	Applicable Document
CDR	Critical Design Review (Milestone)
CUST	Customer (Role)
DEV	Developer (Role)
KOM	Kick-Off Meeting (Milestone)
LDEV	Lead Developer (Role)
MAN	Manager (Role)
NA	Not Applicable
PCM	Project Close-Down Meeting (Milestone)
PDR	Preliminary Design Review (Milestone)
PM	Project Manager (Role)
QM	Quality Manager (Role)
QMS	Quality Management system
QR	Qualification Review (Milestone)
RM	Release Manager (Role)
SEPG	Software Engineering Process Group (Role)
SRM	Senior Manager (Role)
SQA	Software Quality Assurance (Role)

1.5 Document structure

Section 1, Introduction - provides information about the document content including the overview, purpose, audience, definitions and acronyms, structure and references list.

Section 2, Process overview - provides a general description of the process.

Section 3, Resources - identifies the resources needed to implement the process activities, including human and non-human resources.

Section 4, Process inputs - identifies the inputs considered in the process activities.

Section 5, Process outputs - identifies the outputs resulting from the process activities.

Section 6, Roles and responsibilities - defines the roles, responsibilities, and authorities of the person that somehow are involved in the process.

Section 7, Process initiation and termination events - indicates the process start trigger conditions (start event) and process end trigger conditions (end event).

Section 8, Success criteria - indicates the success criteria that should be accomplished by the process, includes partial implementations criteria.

Section 9, Design and Implementation Process activities, identifies the activities performed in the scope of the process.

1.6 References

1.6.1 Applicable documents

- [AD – 1] Software Development Process, Qimonda, QAG-SQI-PRC-2006-0010.
- [AD – 2] DC Quality Management System, Glossary, Qimonda, QAG-SQI-PRC-2006-0001.
- [AD – 3] Roles and Responsibilities, Qimonda, QAG-SQI-GDK-2006-0053
- [AD – 4] Quality Management System Website, Qimonda,
http://qShare.qimonda.com/sites/porto_dc/qms.
- [AD – 5] QAG-QM-MMO-2008-0979-Detailed-Design-Specification (Detailed Design example), Qimonda,
http://qshare.qimonda.com/sites/ITPADQ/processes/QMS_CMMI_L3/D-I/QAG-SQI-DSP-2008-1013-Detailed-Design-Example.ppt
- [AD – 6] Code Review User Manual, Qimonda,
AG-SQI-MAN-2007-0838-CodeReviewForm-UserManual

1.6.2 Reference documents

- [RD – 1] Detailed Design, Qimonda,
QAG-SQI-DSP-2008-1013-Detailed-Design
- [RD – 2] Administration Manual, Qimonda,
QAG-SQI-TPL-2006-0311-Administration-Manual
- [RD – 3] Installation Manual, Qimonda,
QAG-SQI-TPL-2006-0312-Installation-Manual
- [RD – 4] User Manual, Qimonda, QAG-SQI-TPL-2006-0313-User-Manual
- [RD – 5] Detailed Design Specification, Qimonda,
QAG-QM-MMO-2008-0979-Detailed-Design-Specification

2. Process overview

The process of Design and Implementation is to design, develop, and implement solutions to requirements. It corresponds to the phase of software development in which the requirements are turned into a software system.

3. Resources

Main resources are:

Application	Description
Rational Rose	Design Modeler
Development IDE	Visual Studio .Net, Eclipse 3.x, etc...
Test Case Editor	Test case design tool.

Table 4.1: Table1- Process Resources

4. Process inputs

The New QMS Engineering Processes

The inputs to the Design and Implementation process are:

- Requirements and associated test cases;
- Use Cases, and associated System Test Cases when available;
- High Level Design when available.

5. Process outputs

The outputs of Design and Implementation process are:

- Software products and/or software product components
- Technical data package

6. Roles and responsibilities

Roles and responsibilities are generally described in the DC's QMS Intranet page (http://qshare.qimonda.com/sites/porto_dc/qms/default.aspx).

Additionally, the following roles have specific responsibilities within the scope of the current process definition:

Project Manager – Manage all the development activities.

Design Advisor – Define technical solutions and its selection criteria, and select the most applicable solution to meet the requirement against the selection criteria.

Lead Developer – Develop the selected solution into high level design.

Developer – Define the detailed design according to high level design. Implement detailed design into product or product components.

SQA – Monitor metrics and their targets defined in Quality Assurance Plan.]

7. Process initiation and termination events

The Design and Implementation Process starts when:

- Software requirement is reviewed and approved by customer, project manager, and senior manager/managers (when necessary).

The Design and Implementation Process ends when:

- CDR (Critical Design Review) meeting.

8. Success criteria

The success criteria of the Design and Implementation process are:

- Product or product components are generated against all allocated requirements

9. Design and Implementation Process Activities

The Design and Implementation Process includes three different activities: High Level Design – DI.A1 –, Detailed Design – DI.A2 –, and Implement the design – DI.A3.

DI.A1, Solution Definition and High Level Design – Identify and analyze alternative solutions against predefined selection criteria (to evaluate at what level the solution satisfies the software requirements), and select the most applicable solution via a formal evaluation process (refer to DAR about formal evaluation processes) or other evaluation method. In Qimonda IT PA context, in some projects, Domain Functions take over solution definition and selection responsibility. In such cases, the output of this activity is called High Level Design, and includes requirements, use cases, associated system test cases when necessary. During the solution definition and selection, a “make-or-reuse analysis” should be done to determine which products or product components should be reused, which should be built internally or out sourced to business partner.

DI.A2, Detailed Design – Develop the selected solution into a design, and establish/maintain the interfaces and technical data package.

DI.A3, Implement the design – After detailed design is completed, it is implemented into product components with necessary data package and interface descriptions, unit tests are run and passed, necessary documents are generated.

In the image below (Figure 1) is possible to overview the relationship between inputs and outputs in Design and Implementation process activities.

In the next images are presented the Integration, Verification and Validation activities’ detailed descriptions.

4.2.1.1 Support Documents

To support the Design and Implementation process were created a set of templates, some directly related to the process – Solution Specification template, Detailed Design template – and some indirectly related – Administration Manual template, Installation Manual template, User Manual template.

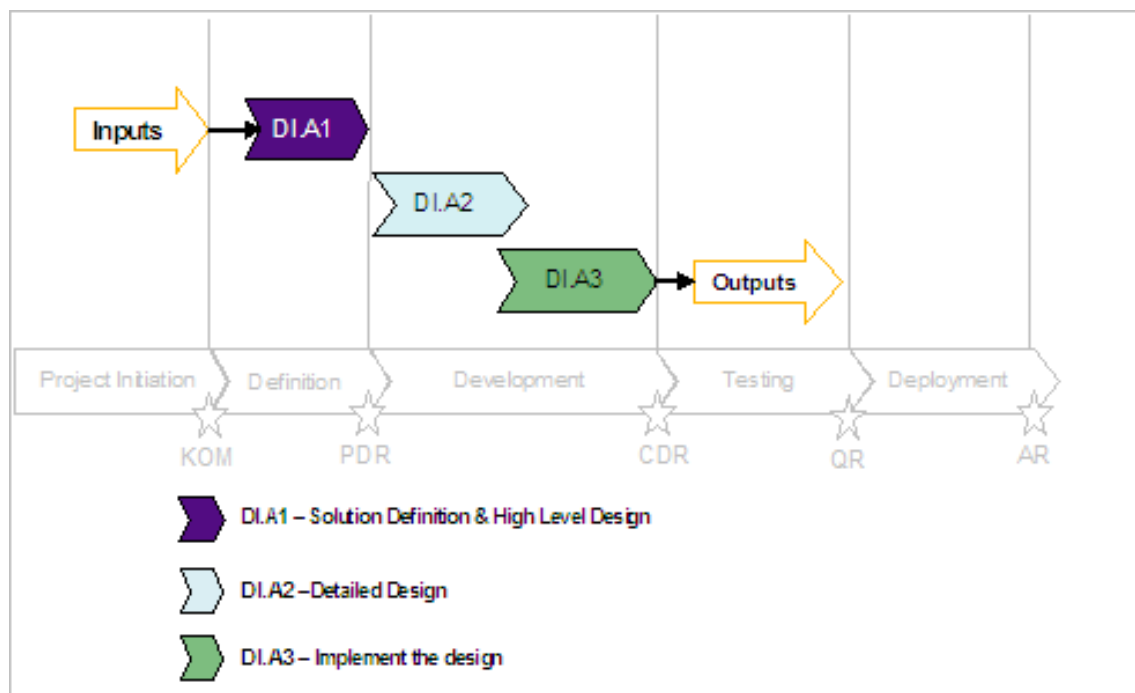


Figure 4.13: Figure 1 – Relationship between the Design and Implementation process activities and the Software Development Process phases

The New QMS Engineering Processes

ACTIVITY DESCRIPTION				
Process:	DESIGN AND IMPLEMENTATION PROCESS			
Activity Title:	Solution Definition & High Level Design		Activity Id:	D&I.A1
Start Event(s):	Requirement is reviewed and approved			
End Event(s):	PDR			
Roles:	<i>Responsible</i> Lead Developer	<i>Approval</i> PM	<i>Consulted</i> Project team/Customer	<i>Informed</i> Project team/Customer
Purpose: Define the most applicable solutions from alternative solutions defined to answer the requirements against selection criteria. Generate the High Level Design so called architecture design to according to the most applicable solution.				
Inputs: <ul style="list-style-type: none"> - Software requirements and associated test cases - Use Cases User Acceptance Test Cases 				
Outputs: <ul style="list-style-type: none"> - Alternative solutions and its selection criteria refer to [RD-1] for template - Selected solution which is the most applicable for the software requirement - refer to [RD-1] - High Level Design 				
Implementation details: <ol style="list-style-type: none"> Identify selection criteria <ol style="list-style-type: none"> Identify the architecture requirements, drivers and constraints Develop the screening criteria for selecting the best alternative solution. Identify alternative solutions <ol style="list-style-type: none"> Identify and resolve issues with the alternative solutions and requirements Identify technologies currently in use and new product technologies for competitive advantage Identify the product component solutions that will be reused or acquired. Generate alternative solutions, and obtain a complete requirements allocation for each alternative. Evaluate alternative solutions <ol style="list-style-type: none"> Evaluate each alternative solution/set of solutions against the selection criteria. Based on the evaluation of alternatives, assess the adequacy of the selection criteria and update these criteria as necessary. Select the best set of alternative solutions that satisfy the established selection criteria via formal evaluation (refer DAR process) or other evaluation methods. Establish the requirements allocation of the selected set of solutions as the final allocation of the requirements to those product components. Generate the High Level Design <ol style="list-style-type: none"> Generate the high level design so called architecture design against the most applicable solution defined. Define system test case package for system test verification and validation. 				

Figure 4.14: DI.A1

The New QMS Engineering Processes

ACTIVITY DESCRIPTION				
Process:	DESIGN AND IMPLEMENTATION PROCESS			
Activity Title:	Detailed Design	Activity Id:	D&IA2	
Start Event(s):	High level design passed acceptance criteria			
End Event(s):	Detailed design is achieved			
Roles:	Responsible Developer	Approval Lead Developer	Consulted Project team	Informed Project team
Purpose:	<ul style="list-style-type: none">- Develop the high level design into detailed design, so called detailed design.			
Inputs:	<ul style="list-style-type: none">- High Level Design			
Outputs:	Break down Detailed Design [RD-5] into the top level implementation, which includes: <ul style="list-style-type: none">- Product components design- Interface design- Technical data package- Integration Test case			
Implementation details:				
<ol style="list-style-type: none">1. Establish and maintain design criteria<ol style="list-style-type: none">a. Establish or select design criteria from organizational process asset library.b. Maintain the criteria through the definition of the product designc. Apply the criteria to the detailed design activities.2. Develop product or product components design<ol style="list-style-type: none">a. Define design methods appropriate for the product or product componentsb. Adheres to applicable design standards to the designc. Verify the design adheres to allocated requirements and document the design.3. Establish and maintain interfaces<ol style="list-style-type: none">a. Identify interfaces associated with other product components and external system.b. Document the selected interface designs and the rationale for the selectionc. Ensure the compatibility of the interfaces throughout the life of the productd. Maintain a repository for interface data accessible to projects participantse. Ensure that product components and interfaces are marked to ensure easy and correct connection to the joining product component.4. Identify necessary technical data.<ol style="list-style-type: none">a. Analyze solutions to identify necessary data for product or product component implementation.b. Revise the technical data package as necessary5. Manage interfaces<ol style="list-style-type: none">a. Review interface data for completeness and ensure complete coverage of all interfacesb. Periodically review the adequacy of interface descriptions				

Figure 4.15: D&IA2

The New QMS Engineering Processes

ACTIVITY DESCRIPTION				
Process:	DESIGN AND IMPLEMENTATION PROCESS			
Activity Title:	Implement the Design		Activity Id:	D&I.A3
Start Event(s):	Detailed design is approved by Lead Developer			
End Event(s):	CDR - Critical Design Review			
Roles:	<i>Responsible</i> Developer	<i>Approval</i> Lead Developer	<i>Consulted</i>	<i>Informed</i> Project team
Purpose:	<ul style="list-style-type: none"> - Realize the software design into product or products components 			
Inputs:	<ul style="list-style-type: none"> - Detailed Design [RD-5] which includes <ul style="list-style-type: none"> - Product components design - Interface design - Technical data package - Integration test case 			
Outputs:	<ul style="list-style-type: none"> - Product or product component - Necessary documentation, includes <ul style="list-style-type: none"> - Administration manual [RD-2], Installation manual [RD-3], User manual [RD-4] (when necessary) - Training materials (when necessary) 			
Implementation details:	<ol style="list-style-type: none"> 1. Implement product components design <ol style="list-style-type: none"> a. Use effective methods to implement the design and associated test scripts b. Select applicable standards and criteria from organizational process asset library, and apply it thru design implementation c. Conduct code review [AD-6] for the selected code files/segment d. Perform the unit test. e. Revise the product component as necessary after peer review or unit test. 2. Generate necessary document <ol style="list-style-type: none"> a. Identify and develop necessary documents for administration, installation, operation and maintenance. b. Attach necessary documents for integration/system test, such as integration environment specification, integration sequence, integration criteria and guideline, test cases for integration and system test respectively. c. Select applicable documentation standards and templates [RD-2], [RD-3], [RD-4] from organizational process asset library, and apply it thru documentation. d. Conduct peer review for the selected documents. e. Revise the documents as necessary after peer reviews. 			

Figure 4.16: D&I.A3

4.2.2 Integration, Verification and Validation Process

1. Introduction

1.1 Overview

This document details the Integration, Verification and Validation Process used by Qimonda IT PA.

1.2 Purpose

The purpose of this process is to integrate all product components into one product and to test all functional groups and areas. The emphasis is on planning and verifying that the product was built right and validating that the team has built the right thing. Internal interfaces and critical external interfaces should also be tested.

1.3 Audience

The audience of this document is all project managers, test managers, testers and other members that participate in QMS compliant projects developed within the organization in one of the roles referred in ?[AD-2].

The content of this document is property of Qimonda AG and distribution to external entities (such as customer) is not allowed unless there is a written authorization from the Global SEPG.

1.4 Definitions and acronyms

Full acronyms and definitions are defined in the glossary?[AD-1].

1.4.1 Definitions

For the purpose of this document the following definitions are applicable:

DC	Development Center
NA	Not Applicable

1.4.2 Acronyms

For the purpose of this document the following acronyms are applicable:

CDR	Critical Design Review (Milestone)
DI	Design and Implementation (Process)
IVV	Integration, Verification and Validation (Process)
KOM	Kick-Off Meeting (Milestone)
PCM	Project Close-Down Meeting (Milestone)
PDR	Preliminary Design Review (Milestone)
PM	Project Manager (Role)
QR	Qualification Review (Milestone)
SQA	Software Quality Assurance (Role)
UATs	User Acceptance Tests

1.5 Document structure

Section 1, Introduction, provides information about the document content including the overview, purpose, audience, definitions and acronyms, structure and references list.

Section 2, Process overview, provides a general description of the process.

Section 3, Resources, identifies the resources needed to implement the process activities, including human and non-human resources.

Section 4, Process inputs, identifies the inputs considered in the process activities.

Section 5, Process outputs, identifies the outputs resulting from the process activities.

Section 6, Roles and responsibilities, defines the roles, responsibilities, and authorities of the person that somehow are involved in the process.

Section 7, Process initiation and termination events, indicates the process start trigger conditions (start event) and process end trigger conditions (end event).

Section 8, Success criteria, indicates the success criteria that should be accomplished by the process, includes partial implementations criteria.

Section 9, Integration, Verification and Validation Process , identifies the activities performed in the scope of the process.

1.6 References

1.6.1 Applicable documents

- [AD – 1] DC Glossary, Qimonda, http://qshare.qimonda.com/sites/porto_dc/st/Lists/Glossary/Group%20By%20Context.aspx
- [AD – 2] Roles and Responsibilities, Qimonda, QAG-SQI-GDK-2006-0053
- [AD – 3] Integration, Verification and Validation Plan template, Qimonda, QAG-SQI-TPL-2008-1062
- [AD – 4] Test Report template, Qimonda, QAG-SQI-TSR-2008-1173
- [AD – 5] Release Notes template, Qimonda, QAG-SQI-TPL-2008-1063
- [AD – 6] Release Notes guidelines, Qimonda, QAG-SQI-GDK-2008-1071
- [AD – 7] Release Mail guidelines, Qimonda, QAG-SQI-GDK-2008-1070
- [AD – 8] Release Procedure, Qimonda, QAG-SQI-PRO-2008-1067
- [AD – 9] Project Profile, Qimonda, QAG-SQI-TPL-2006-0004
- [AD – 10] Inspection Review Procedure, Qimonda, QAG-SQI-PRO-2006-0365
- [AD – 11] Email pass around Review Procedure, Qimonda, QAG-SQI-PRO-2008-0762
- [AD – 12] Over the shoulder Review Procedure, Qimonda, QAG-SQI-PRO-2008-0764
- [AD – 13] Change Management Procedure - Project, Qimonda, QAG-SQI-PRO-2008-0788

1.6.2 Reference documents

2. Process overview

This process relates mainly with the phases of software development and testing in which individual software components are combined and verified as a group, and

the software is packaged and delivered when it meets Integration, Verification and Validation successful criteria.

To better understand this process and related activities some concepts should be clarified:

Work product is a useful result of a process. This can include files, documents, products, components, specifications, etc. A key distinction between a work product and a product component is that a work product is not necessarily part of the product (as internal documents).

Product is a work product that is intended for delivery to a customer or end user.

Product Component is a work product that is a lower level component of the product. Product components are integrated (colored as orange in Figure 1) to produce the product.

Verification (colored as violet in Figure 1) confirms that work products properly reflect the requirements specified for them. In other words, verification ensures that “you’ve built it right”.

Validation (colored as blue in Figure 1) confirms that the product, as provided (or as it will be provided), will fulfill its intended use. In other words, validation ensures that “you’ve built the right thing”.

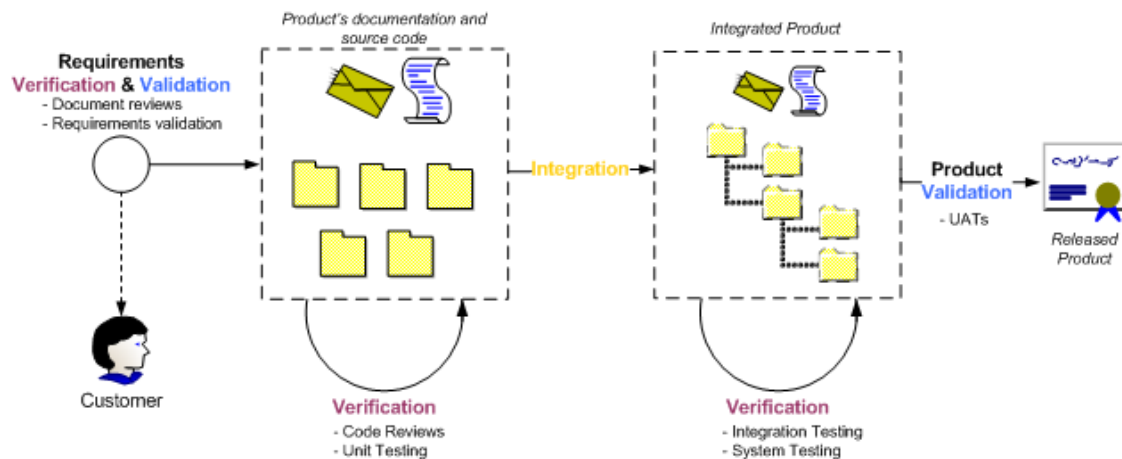


Figure 4.17: Figure 1 - IVV Process Overview

3. Resources

Main resources are:

4. Process inputs

The inputs to the Integration, Verification and Validation process can be:

- Product components (and dependencies)
- Technical Documentation (Configuration Items)

Application	Description
Rational Functional Tester	Tool that supports test case management and test execution
ClearQuest	Tool that supports change management
Test Case Editor	Tool that supports test cases design
Test Case Viewer	Tool that supports the reading of test cases
qShare	Tool used for communication and results sharing
Excel	Tool for filling-out the Integration, Verification and Validation Plan Template
Word	Tool for producing the Integration, Verification and Validation documentation
ClearCase	Repository and version control system

- Test cases (defined by the customer)
- Organizational verification criteria (to support peer reviews, as coding guidelines and checklists)
- Requirements Specification
- Product or product components design model (output from DI.A2)
- Project Plan
- Configuration Management Plan
- Interfaces design and description (output from DI.A4)

5. Process outputs

The outputs of Integration, Verification and Validation process are:

- Integration, Verification and Validation plan (template in [AD-3])
- Peer review database updated (with reviewed artifacts)
- Test reports (template in [AD-4])
- Release notes (signed and approved by the PM and SQA) (template in [AD-5] and guidelines in [AD-6])
- Release Mail (template and guidelines in [AD-7])
- Software package

6. Roles and responsibilities

Roles and responsibilities are generically described in [AD-2].

Additionally, the following roles have specific responsibilities within the scope of the current process definition:

Project Manager – Coordinate with Test Manager to leverage all integration and verification activities and validate and approve deliverables.

Developer – Support Tester to develop test procedures, test scenarios, test scripts, code reviews, and find and fix defects.

Test Manager – Develop Integration, Verification and Validation Plan, and manage all the verification and validation activities.

Tester – Participate in peer reviews and register the results in the peer reviews database, execute test plan, update test result and status, and generate test report.

Release Manager – Package and deliver the software product to customer according to the procedure in [AD-8].

7. Process initiation and termination events

The Integration, Verification and Validation process starts when:

- Project is assigned

The Integration, Verification and Validation process ends when:

- Final software product is released to the customer

8. Success criteria

The success criteria of the Integration, Verification and Validation process are:

- Product components are integrated
- Integrated product is verified against requirements
- Integrated and verified product is validated against customer needs
- Integrated, verified and validated product is released to the customer

9. Integration, Verification and Validation Process Activities

Integration, Verification and Validation Process includes eight different activities, detailed in the following sections:

IVV.A1, Plan the Verification and Validation activities – From the beginning of the project, the verification and validation must be planned. Verification must be planned to define which and how configuration items are going to be verified against requirements. Validation methods should be planned to try to pro-actively evaluate that the product we are developing fulfills its intended use (more than simply checking that it meets the requirements). Estimate the effort required to verify each configuration item and validate the product and plan who will do it and when to do it.

IVV.A2, Review Documentation – Do peer reviews for each relevant document that has been completed (e.g. Requirements Specification document). Record the results (issues list) and report them to the author of the document. After the document has been corrected, ensure the issues were taken care of.

IVV.A3, Validate the Product - From the beginning of the requirements specification you can start to validate your understanding of the product with the customer, through prototypes, low level design, etc. When you have a functional product, you can have the customer to provide test case scenarios, so you can perform user acceptance tests.

IVV.A4, Plan the Integration – At the end of the Definition phase (PDR meeting) you can plan the integration of each product component into the final product. Identify the sequence in which the product components are going to be integrated, the environment requirements and the criteria that each product component has to meet before it can be integrated (entry criteria).

IVV.A5, Perform Code Reviews and Unit Tests – During the Development phase, perform code reviews and unit tests as planned in the Integration, Verification and Validation Plan. Compare the results to the criteria defined in the Integration, Verification and Validation Plan and determine the acceptability of each product component. Plan follow-ups and re-tests so you can verify the identified issues were taken care of. Report the results and acceptability of each work product to the Project Manager.

IVV.A6, Integrate Product – Whenever components are ready for integration, you must ensure that interfaces, both internal and external, are complete and compatible and that each component is properly identified and functions according to its description. Only then the components can be integrated according to the sequence and procedures defined in the Integration, Verification and Validation Plan.

IVV.A7, Perform Integration and System Tests - When the completely integrated product is available, integration test cases are executed according to plan and the test results (fail/pass) are recorded. These results must be compared to defined criteria to determine acceptability. When an issue is found (a test fails) the cause and possible corrective action should be investigated for Developers to initiate corrective action.

IVV.A8, Package and Deliver the Product – After the completely integrated product has met the evaluation criteria, package the product (e.g. source

code, user manual, release notes, etc) and deliver/release it to the appropriate customer.

In the image below (Figure 2) is possible to overview the relationship between the Integration, Verification and Validation process activities and the Software Development Process phases.

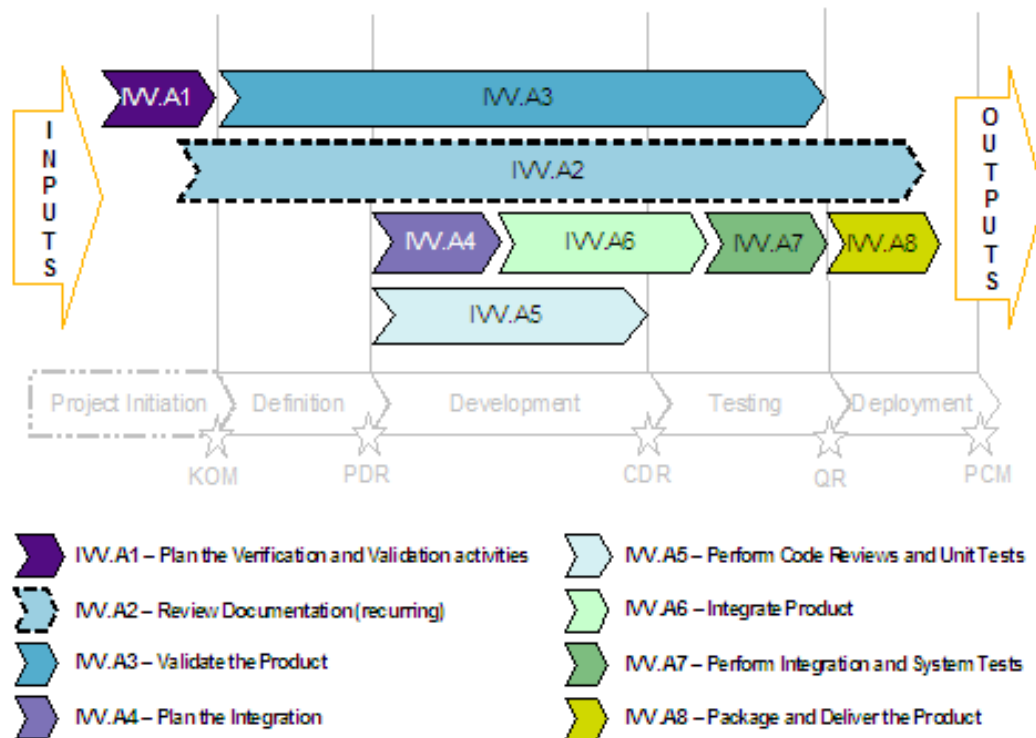


Figure 4.18: Figure 2 - Relationship between the process activities and the Software Development Process phases

In the next images are presented the Integration, Verification and Validation activities' detailed descriptions.

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Plan the Verification and Validation activities		Activity Id:	IWA.1
Start Event(s):	Project assignment			
End Event(s):	KOM (draft version) and PDR (approved version)			
Roles:	Responsible	Approval	Consulted	Informed
	Project Manager	SQA	Project Team, Test Manager	Customer
Purpose:	The purpose of this activity is to plan the verification (peer reviews, unit tests, integration tests, etc) and validation (prototypes, user acceptance tests, manuals, etc) that are going to be performed in the project.			
Inputs:	<ul style="list-style-type: none">- Requirements from Customer (delivered in any format)			
Outputs:	<ul style="list-style-type: none">- Integration, Verification and Validation Plan- Project Plan updated with effort, people responsible and schedule of the verifications and validations- Project Profile updated (Work Environment section)			
Implementation details:	<ol style="list-style-type: none">1. Plan the Verification [AD-3]<ol style="list-style-type: none">a. Identify which configuration items will be verifiedb. Identify the entry criteria that each configuration item has to meet to be verifiedc. Identify which verification methods (which kind of peer reviews, tests, etc) to use for each configuration item and where are going to be recorded the resultsd. Identify the environment requirements for each verification (e.g.: for a peer review, a room, a package of materials and reviewers; for the execution of a test, simulators, emulators, scenario generators, data reduction tools, environmental controls, and interfaces with other systems)e. Identify the exit criteria that each configuration item has to meet to be accepted to fulfill the requirements2. Plan the Validation [AD-3]<ol style="list-style-type: none">a. Identify which validation methods to use (e.g.: User acceptance tests, Prototypes, User Manual, etc).b. Identify the environment requirements (e.g.: an environment that simulates the end-user environment)c. Define when and how to use each of the selected validation methods (entry criteria)d. Identify the exit criteria for the product to be accepted (e.g.: 2 weeks without high or medium severity accepted issues)3. Estimate the effort required to verify and validate each configuration item, plan who will do it and when to do it.			

Figure 4.19: IVV.A1

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Review Documentation	Activity Id:	IVV.A2	
Start Event(s):	Document is ready to be reviewed			
End Event(s):	Document reviewed			
Roles:	Responsible Author	Approval Project Manager	Consulted Reviewer	Informed SQA
Purpose:	The purpose of this activity is to review every planned document produced within the scope of the project to identify possible issues, ensuring that they meet the defined quality standards			
Inputs:	<ul style="list-style-type: none">- Integration, Verification and Validation Plan (identifying the documents that should be reviewed)- Document(s) to be reviewed- Organizational verification criteria for documents (coding guidelines, checklists...)			
Outputs:	<ul style="list-style-type: none">- Document(s) reviewed- Review results (issues list, measurements, participants list...) followed-up until closure			
Implementation details:	<ol style="list-style-type: none">1. Make sure you have the right environment for each document you'll have to review (e.g.: a room, a package of materials and reviewers), according to the Integration, Verification and Validation Plan [AD-3].2. Perform the peer review according to the planned method:<ol style="list-style-type: none">a. Inspection [AD-10]b. Email pass-around [AD-11]c. Over-the-shoulder [AD-12]3. Report should be validated and approved by the Project Manager.			

Figure 4.20: IVV.A2

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Validate the Product	Activity Id:	IWA3	
Start Event(s):	KOM			
End Event(s):	QR meeting			
Roles:	Responsible Project Manager	Approval Customer	Consulted Test Manager, Project Team	Informed SQA
Purpose:	The purpose of this activity is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment, that is, to validate the project team understanding of the product with the customer, to ensure they are building the right thing.			
Inputs:	<ul style="list-style-type: none">- Integration, Verification and Validation Plan- Project Plan- Project Profile- Integrated and verified product User acceptance test cases			
Outputs:	<ul style="list-style-type: none">- Work products that support the validation (prototypes, manuals, ...)- Product validated and ready to be released- Test scripts (from UATs)- Test Report (from UATs)			
Implementation details:	<ol style="list-style-type: none">1. Prepare for validation<ol style="list-style-type: none">a. Make sure you have the right environment for each validation method you'll use, according to the Integration, Verification and Validation Plan [AD-3].b. Ensure that you meet the method's entry criteria defined in the Integration, Verification and Validation Plan from IWA1.2. Validate<ol style="list-style-type: none">a. Perform the validation methods defined in the Integration, Verification and Validation Plan from IWA1 according to the schedule in the Project Plan.b. Identify products and product components that do not meet the specified exit criteria (follow the change management procedure in [AD-13]).3. Deliver the work product that supports the validation to obtain customer's commitment and acceptance.			

Figure 4.21: IVV.A3

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Plan the Integration	Activity Id:	IW.A4	
Start Event(s):	Design model complete (PDR)			
End Event(s):	Integration planned or CDR			
Roles:	Responsible	Approval	Consulted	Informed
	Test Manager	Project Manager	Developer, Tester	SQA
Purpose:	The purpose of this activity is to plan the integration of each product component into the final product, defining an integration sequence and the criteria that each product component has to meet before it can be integrated.			
Inputs:	<ul style="list-style-type: none">- Detailed Design (from DI.A2)<ul style="list-style-type: none">- Product components design- Interface design- Technical data package- Integration, Verification and Validation Plan- Project Plan			
Outputs:	<ul style="list-style-type: none">- Integration, Verification and Validation Plan updated- Project Plan updated			
Implementation details:	<ol style="list-style-type: none">Plan the Integration [AD-3]<ol style="list-style-type: none">Identify the sequence in which the product components are going to be integrated (from the detailed design).Identify the integration environment needs (e.g.: is the test server enough? Do we need software from the productive environment?).Define the (entry) criteria that each product component has to meet before it can be integrated. The entry criteria should state what needs to be verified (tested and reviewed) in which product component. Refer to the Verification sheet in the Integration, Verification and Validation Plan [AD-3]. Make sure the Component meets its interface design.Estimate the effort required to integrate each product component, propose who will do it and when to do it so that the PM can update the Project Plan accordingly.			

Figure 4.22: IVV.A4

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Perform Code Reviews and Unit Tests		Activity Id:	IW.A5
Start Event(s):	Start of the Development phase (PDR)			
End Event(s):	End of the Development phase (CDR)			
Roles:	Responsible	Approval	Consulted	Informed
	Developer	Reviewer	-	Project Manager
Purpose:	The purpose of this activity is to use the planned methods and defined criteria (from Integration, Verification and Validation Plan) to ensure that the selected product components meet their specified requirements.			
Inputs:	<ul style="list-style-type: none">- Product or Product's components.- Integration, Verification and Validation Plan			
Outputs:	<ul style="list-style-type: none">- Test Log (unit tests)- Test Report (Unit tests)- Review results (issues list, measurements, participants list...) followed-up until closure			
Implementation details:	<ol style="list-style-type: none">1. Make sure you have the right environment for each work product you'll have to verify (e.g.: for a peer review, a room, a package of materials and reviewers; for the execution of a unit test, simulators, emulators, scenario generators, data reduction tools, environmental controls, and interfaces with other systems), according to the Integration, Verification and Validation Plan [AD-3].2. Perform the code review according to the planned method:<ol style="list-style-type: none">a. Inspection [AD-10]b. Email pass-around [AD-11]c. Over-the-shoulder [AD-12]3. Perform the unit test as defined in the Integration, Verification and Validation Plan.4. Record the results (fail/pass, issues list) of the unit tests in the Test Log/Report. This can be made manually [AD-4] or using automated tools (e.g. Test Manager).5. Reports (from the code reviews and unit tests) should be validated and approved by the Project Manager.			

Figure 4.23: IVV.A5

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Integrate Product		Activity Id:	IVV.A6
Start Event(s):	Product components developed, reviewed and unit tested or CDR			
End Event(s):	Product integrated (before QR)			
Roles:	Responsible	Approval	Consulted	Informed
	Test Manager	-	Lead Developer, Developer, Analyst	Tester
Purpose: The purpose of this activity is to integrate product components into the final product and have it ready for integration and system testing.				
Inputs: <ul style="list-style-type: none">- Product or Product components- Detailed design- Integration, Verification and Validation Plan- Project Profile				
Outputs: <ul style="list-style-type: none">- Integrated Product				
Implementation details: <ol style="list-style-type: none">1. Verify the integration environment against the integration environment specification in the Integration, Verification and Validation Plan [AD-3].2. Ensure that each component meets its entry criteria defined in the Integration, Verification and Validation Plan from IVV.A1.3. Integrate the product components according to the Integration, Verification and Validation Plan from IVV.A1.				

Figure 4.24: IVV.A6

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Perform Integration and System Tests		Activity Id:	IVV.A7
Start Event(s):	Product integrated (before QR)			
End Event(s):	The product is verified against the requirements (QR)			
Roles:	Responsible	Approval	Consulted	Informed
	Tester	Test Manager	Developer, Analyst	Project Manager
Purpose: The purpose of this activity is to use the planned methods and defined criteria (from Integration, Verification and Validation Plan) to ensure that the integrated product meet their specified requirements.				
Inputs: <ul style="list-style-type: none">- Integrated Product- Integration, Verification and Validation Plan- Project Profile- Project Plan				
Outputs: <ul style="list-style-type: none">- Test Cases- Test Scripts- Test Logs- Test Reports				
Implementation details: <ol style="list-style-type: none">1. Make sure you have the right environment to test the integrated product (e.g.: simulators, emulators, scenario generators, data reduction tools, environmental controls, interfaces with other systems), according to the Integration, Verification and Validation Plan [AD-3].2. Develop the test cases needed to verify each requirement (use Test Manager, rtpar or the IVV Plan template [AD-3])3. Perform the integration and system tests according to what is defined in the Integration, Verification and Validation Plan (develop test scripts for automatic tests).4. Record the results (fail/pass, issues list) of the integration and system tests in the Test Log/Report. This can be made manually[AD-4] or using automated tools (e.g. Test Manager).5. Compare the results to the criteria defined in the Measurement Plan (included in Project Profile [AD-9]) and determine the acceptability of the product6. Test Log/Report should be validated and approved by the Test Manager.				

Figure 4.25: IVV.A7

ACTIVITY DESCRIPTION				
Process:	INTEGRATION, VERIFICATION AND VALIDATION PROCESS			
Activity Title:	Package and Deliver the Product		Activity Id:	IW.A8
Start Event(s):	QR			
End Event(s):	E-mail sent to Customer with released product			
Roles:	Responsible	Approval	Consulted	Informed
	Release Manager	Project Manager	Project Team	Customer
Purpose: The purpose of this activity is to package the integrated and tested product, along with all related relevant documentation, and deliver it to the appropriate Customer.				
Inputs: <ul style="list-style-type: none">- Product validated and ready to be released.- All the technical documentation configuration items to be delivered.				
Outputs: <ul style="list-style-type: none">- Software Package (source code, manuals)- Release Mail- Release Notes				
Implementation details: <ol style="list-style-type: none">1. Check if all the technical documentation configuration items are finished and reviewed as planned2. Check if the product was integrated and tested as planned3. Do the release according to the Release procedure [AD-8]				

Figure 4.26: IVV.A8

4.2.2.1 Support Documents

To support the Integration, Verification and Validation process were created a set of documents to support the process, some were procedures – Release Procedure, other were templates – Integration, Verification and Validation Plan template, Test Report template, Release Notes templates, and other were purely support – Release Checklists.

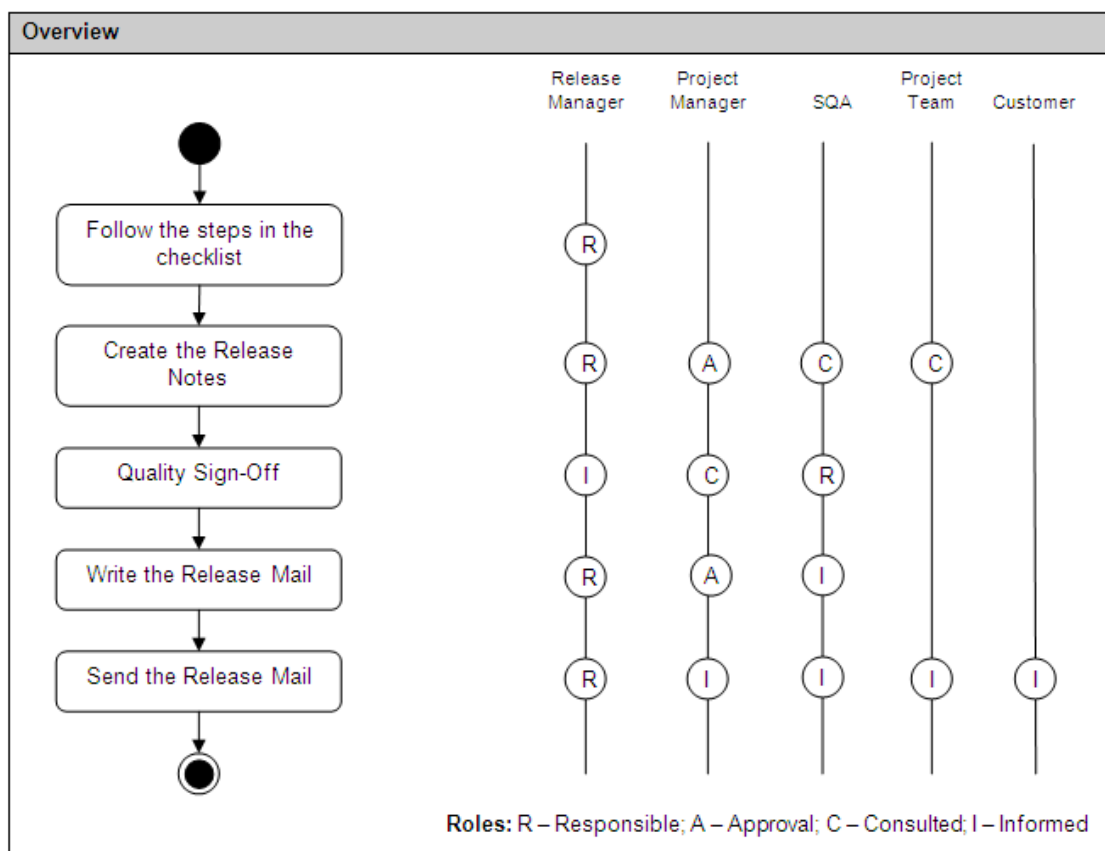
The Integration, Verification and Validation Plan is an important support document to the IVV process and is presented in Appendix [G](#).

As the Releases were thoroughly studied in this project, due to its importance to the organization, since it is an important interface with the client, the Release-related materials are presented in this report.

RELEASE PROCEDURE

I – Involved Objectives

The purpose of this document is to present the procedure that should be followed for every project release (documents, source code, executables, etc.) produced within the Porto Development Center. Project members often consider the release stage as a low priority activity. The use of a release procedure cannot guarantee a perfect release, but not using it almost guarantees a faulty one. This procedure prevents oversights during software release and ensures that all project release stakeholders are informed about the release. Its usage also guarantees that all release deliverables are delivered to the customer. This procedure defines the necessary steps to produce a release, their inputs and expected outputs, as well as the roles and responsibilities of the involved parties.



Roles and Responsibilities

Roles and responsibilities are generally described in [RD-1]. Additionally, the following roles for this procedure should also be considered:

Release Manager – Is responsible for the Release. Is responsible for creating the Release Notes, publishing them on the qShare page (after the approval of the PM), creating the release package, writing the release mail and sending it to the stakeholders (after the approval of the PM).

Project Manager – Ensures the product is ready for the Release. Is responsible for reviewing and approving the release notes and the mail to send to all the stakeholders.

Software Quality Assurance Engineer – Monitors the quality assurance activities in the project. Is responsible for the quality sign-off of the release, making sure the product is released according to the quality standards of the DC.

Inputs

Release Checklist The Release Checklist [RD-2] is a sequence of steps that the Release Manager has to follow to make the release.

Distribution List The list to which the release mail will be sent.

Deliverables List List of all items that are part of the release (e.g. available in project's master plan).

Configuration Baseline The configuration baseline is used as input for the release creation. It contains all configuration items that are part of the release, which were meet the quality requirements defined for the product. The Configuration Baseline corresponds to the label/tag generated in ClearCase for the release creation.

Outputs

Release Package The final package of Work Products (Artifacts /Components / ...) to be released.

Release Mail Mail sent to the stakeholders with the information of the Release [RD-5].

Release Notes Document that contains all relevant release information. It will usually follow the QMS template for Release Notes [RD-4].

Tasks	
Follow the Steps in the Checklist The Release Manager has to make the release package following the checklist [RD-2] step-by-step.	RM
Create the Release Notes The Release Manager has to create the Release Notes following the Release Notes Template [RD-4] or, if the templates aren't suitable, making sure they have the mandatory content (according to [RD-3]). The Project Manager has to review the Release Notes (referring to the Release Notes Template and organizational quality standards), and inform the Release Manager of the changes that are to be done (if any), and then approve it.	RM (PM)
Quality Sign-Off After the PM approval of the Release Notes, the SQA has to sign-off the Release Notes to ensure that the product is released according to the quality standards of the DC. This is not an official approval, is just a quality certificate. A product can be released without the SQA quality sign-off, but a good explanation should be given to Management.	SQA
Write the Release Mail The Release Manager has to write the Release Mail following the Release Mail Template [RD-5], and show/send it to the PM for approval. The Project Manager has to review the Mail (referring to the Release Mail Template and organizational quality standards), checking the content and the links, and after reviewing, have to inform the Release Manager of the changes that are to be done (if any), and then approve it.	RM (PM)
Send the Release Mail The release mail is to be sent to the distribution list (with the relevant stakeholders), with the sender and signature as indicated in the Release Mail template [RD-5], after the approval of the PM.	RM

Guidebooks / Templates / Checklists / Other Sources

- [RD – 1] Roles and Responsibilities, Qimonda, QAG-SQI-GDK-2006-0053
- [RD – 2] Release Checklist, Qimonda, QAG-SQI-CHK-2008-1068
- [RD – 3] Release-Notes-Guidelines, Qimonda, QAG-SQI-GDK-2008-1071
- [RD – 4] Release Notes Templates, Qimonda, QAG-SQI-TPL-2008-1063 and QAG-SQI-TPL-2008-1065
- [RD – 5] Release-Mail-Guidelines, Qimonda, QAG-SQI-GDK-2008-1070

The Release Notes templates can be found in Appendix H of this report.

4.3 The WebViewer

The WebViewer is a framework, developed by João Cortez, that would allow publishing the QMS with more navegability through the information. All the other QMS versions were published as a set of documents in the intranet, which made it difficult to consult the information in an “as needed” basis and made it difficult to read all the documents with a logical order, since the documents were not prioritize, just categorized according to the main QMS area they belonged to or to its type – templates were grouped together.

In order to assess the advantages and disadvantages of publishing QMS 2.0 in this framework instead of the usual set of documents, a study was conducted. The

vantages and disadvantages of the framework against the set of documents were identified and the migration effort estimated (through publication of the IVV process and examples of procedures and templates).

ADVANTAGES AND DISADVANTAGES

Advantages

- More Navigability than a set of documents in the intranet (links from processes to procedures and other processes, and etc)
- Contents more Uniform and Coherent than in a set of documents
- Less effort on keeping QMS coherent (less risk of incoherence and redundancy)
- Lighter (to read) than a set of documents
- Less intimidating and easier to consult on a regular basis

Disadvantages

- Some loss of information
 - With the homogenization of the inputs/outputs, we refer only the artifact itself while in a document we can refer a section or a paragraph of the artifact (either we choose to lose this information or we choose to consider an “artifact” the section we want to refer... but then we increase both the effort of keeping QMS coherent and the redundancy)
 - With the homogenization/simplification of the responsibilities in the procedures, there is some information lost about the responsibilities of each role in each step of the procedure... we just have the overall information... (we manage to overcome this by inserting the image with the steps and the responsibilities)
- There cannot be more than one template per artifact (Release Notes have 2 - one in a document, other in a spreadsheet, and whenever we mention the release notes they can be in either format... to resolve this we'd have to mention two release notes as input/output of activities.)
- The Software Development Process has a totally different structure than the other processes... its migration has to be planned in a different way... There's risk of losing information, having a great effort to adapt the process to the framework (or the other way around) and of confusing the user because it will be totally different information shown in the same way.

Process	Number of Activities	Number of Procedures	Number of Guidelines	Number of Templates
IVV	8	4	4	6
REQM	5	0	1	2
MA	6	3	0	4
PM	5	3	5	10
DI	3	0	0	1
OT	4	2	0	2
CM	6	7	1	2
PPQA	5	1	1	5
DAR	5	0	1	1
OPM	5	0	1	0

Table 4.2: Process Data

Besides these, there is the info that is general (the roles, the milestones, the tools, the artifact homogenization, the revision of the history, concepts, glossary and references, and the CMMI mapping).

It is estimated to spend 14.25 days to migrate the data. Adding the setup to make the data available to everyone, this migration is expected to last approximately 15 days (with the possibility to be extended for a week due to problems that may occur and were not expected).

4.4 Conclusions

Besides the definition of the processes itself, it was felt a need to produce documentation – procedures, guidelines, templates – to support the processes’ implementation details. In order to do so, the interviewees were asked to provide examples of the current documentation in use, that were compared between themselves and to the QMS template, to identify the actual needs of the templates. As a way of better understanding the existent differences and needs, quick meetings were held and as conclusions were drawn, a new template was created.

This process focused on the Release related documentation since the organization consider these to be an important interface with the customer.

This analysis produced two Release Notes templates (similar in content), a Release Mail template and a set of Releases Checklists.

The new DI process was comprised by three activities: Solution Definition and High Level Design, Detailed Design, and Implement the Design. To support this process were created the Solution Specification template and the Detailed Design template.

The new IVV process was comprised by eight activities: Plan the Verification and Validation activities, Review Documentation, Validate the Product, Plan the

The New QMS Engineering Processes

Process	Information Data	Hours	Total (Hours)	Total (Days)
IVV	Processes	8	16	2
	Procedures	4		
	Guidelines	2		
	Templates	2		
MA	Processes	6	10.5	1.30
	Procedures	3		
	Guidelines	0		
	Templates	1.5		
CM	Processes	6	14	1.75
	Procedures	7		
	Guidelines	0.5		
	Templates	0.5		
REQM	Processes	5	6	0.75
	Procedures	0		
	Guidelines	0.5		
	Templates	0.5		
PM	Processes	5	14	1.75
	Procedures	3		
	Guidelines	2.5		
	Templates	3.5		
PPQA	Processes	5	8	1
	Procedures	1		
	Guidelines	0.5		
	Templates	1.5		
DAR	Processes	5	6	0.75
	Procedures	0		
	Guidelines	0.5		
	Templates	0.5		
OPM	Processes	5	5.5	0.70
	Procedures	0		
	Guidelines	0.5		
	Templates	0		
OT	Processes	4	6.5	0.80
	Procedures	2		
	Guidelines	0		
	Templates	0.5		
DI	Processes	3	3.5	0.45
	Procedures	0		
	Guidelines	0		
	Templates	0.5		

Table 4.3: Migration Effort Estimates (part 1)

The New QMS Engineering Processes

Item	Total (Hours)	Total (Days)
Roles (27 roles)	4	0.5
Milestones	1	0.125
Tools	1	0.125
Artifacts	4	0.5
History	1	0.125
Concepts	2	0.25
Glossary	2	0.25
References	1	0.125
CMMI mapping	8	1

Table 4.4: Migration Effort Estimates (part 2)

Integration, Perform Code Reviews and Unit Tests, Integrate Product, Perform Integration and System Tests, and Package and Deliver the product. To support these activities were created the Integration, Verification and Validation Plan template, the Test Report template, the Release Procedure, the Release Checklists, the Release Notes templates and the Release Mail template.

Finally a study took place to analyze the advantages and disadvantages of two different ways to publish the QMS. To mitigate the disadvantages found, actions were planned, and so the QMS will be published in the WebViewer.

TOTAL	90	14.25
-------	----	-------

Table 4.5: Total Estimates

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This project was part of an organization's effort to improve its processes, in order to become a maturity level 3 organization. The QMS 2.0 project aimed to close the gaps in the level 2 processes and to implement the new level 3 process areas practices.

The main focus of this project were the four engineering level 3 process areas: Technical Solution, Product Integration, Verification and Validation. The road map toward level 3 certification was previously established based on a SCAMPI B appraisal and Porto DC joined efforts with SuZhou DC to define the new processes.

The processes that were going to cover the four new engineering process areas were the Design and Implementation process (DI), and the Integration, Verification and Validation process (IVV). The former was under the responsibility of the SuZhou DC while the definition of the latter was the main goal of this project.

One of the main challenges was to overcome the dependencies of the two processes: since the IVV process logically follows the DI process, there should have been a previous plan and clear definition to identify what were going to be the outputs of the first, so the second could reflect those choices in its inputs and activities. As that preparation did not took place, a great deal of rework was needed, on both processes, from both DCs, in order to have the two processes aligned. They are still not perfectly aligned but the final details will be adjusted after the release of QMS 2.0.

Design, Implementation, Integration, Verification, and Validation were already being performed in Porto DC but not in a mindful way, so, besides studying the mentioned CMMI process areas and the QMS in place, it was necessary to assess

the current practices in Porto DC. This was done through a series of interviews with project team members. The effort of analyzing the CMMI PAs, the QMS in place and the current practices in use, was defined the new IVV process.

The new DI process was comprised by three activities: Solution Definition and High Level Design, Detailed Design, and Implement the Design. The new IVV process was comprised by eight activities: Plan the Verification and Validation activities, Review Documentation, Validate the Product, Plan the Integration, Perform Code Reviews and Unit Tests, Integrate Product, Perform Integration and System Tests, and Package and Deliver the product. The DI process has still a lot of room for improvement and Porto DC expects to better adjust this process to its reality as soon as the QMS 2.0 is released. The IVV process was more solidly built so there are no major improvements foreseen, just minor adjustments to the Porto DC projects.

During this project, a study was performed to analyze an alternative to the way QMS was currently published. It is published as a set of documents but there was a framework available that could improve the navigability of the QMS information, enabling the project team member to consult the QMS on a “as needed” basis. There is now an action plan to mitigate the disadvantages of this framework, so the QMS 2.0 will be published in the WebViewer.

The author of this project had planned to be able to run a trial of the two new processes, along with another for the usability of the WebViewer, but the project faced delays and that proved impossible to be done, so it is not yet possible to measure the performance of these processes in the field. The major setback was managing the distance and the different timezones. The expected delay between Request and Response was aggravated by these two factors and the process improvement project deadlines were postponed a couple of times.

No proof was, until the writing of this report, produced to backup the definition of these processes, however, the Qimonda Development Centers will use the processes defined in this project and so these will be implemented, tested and reworked until they are fully aligned with the practices in the Development Centers. Although they can not be defined as successfully implemented, it is the author’s belief, based on the studies that led to the definitions of the processes and support documentation, that they will be a cornerstone to the Development Center’s CMMI Maturity Level 3 certification.

5.2 Future Work

To further evolve the processes defined within this project, a project team could be trained in the new processes, so they could implement their practices in a trial project. From this trial we could draw conclusions and enhancements to the new

Conclusions and Future Work

processes before the QMS 2.0 was released. The WebViewer could also be the subject of a trial, to study real usability and navigability through the QMS information.

As these processes cover the main activities within Porto DC development efforts, they could be possible candidates to be quantitatively managed, as Porto DC evolves towards a CMMI Maturity Level 4 organization.

Conclusions and Future Work

References

- [1] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum. *CMMI: Guidelines for Process Integration and Product Improvement*. Addison-Wesley Professional, Second edition, November 2006.
- [2] Watts S. Humphrey. *Managing the Software Process*. Addison-Wesley Professional, January 1989.
- [3] Walter A. Shewhart. *Economic Control of Quality of Manufactured Product*. D. Van Nostrand, 1931.
- [4] Software Engineering Institute. Capability maturity model integration version 1.2 overview, 2007. <http://www.sei.cmu.edu/cmmi/adoption/pdf/cmmi-overview07.pdf>.
- [5] CMU/SEI-2006-TR-004. *Performance Results of CMMI-Based Process Improvement*. Software Engineering Institute, August 2006.
- [6] 3H Technology. 3h technology – empowering business solutions, 2008. <http://www.3ht.com/> – last visited on 5th July 2008.
- [7] Robins Air Force Base. 402nd maintenance wing, 2008. <http://www.robins.af.mil/units/402mw.asp> – last visited on 5th July 2008.
- [8] Wikipedia, the free encyclopedia. Iso 15504, 2008. http://en.wikipedia.org/wiki/ISO_15504 – last visited on 6th July 2008.
- [9] CMU/SEI-2006-TR-011. *Appraisal Requirements for CMMI, Version 1.2*. Software Engineering Institute, 2006.
- [10] Qimonda AG. qshare, 2008.
- [11] Wikipedia, the free encyclopedia. Iso 12207, 2008. http://en.wikipedia.org/wiki/ISO_12207 – last visited on 6th July 2008.
- [12] Volkmar Haase, Richard Messnarz, Günter Koch, Hans Jürgen Kugler, and Paul Decrinis. *Bootstrap: Fine-Tuning Process Assessment*. IEEE Computer Society Press, 2006.
- [13] Bell Canada. Trillium model description, 2006. <http://www2.umassd.edu/swpi/BellCanada/trillium-html/trillium.html> – last visited on 6th July 2008.

REFERENCES

- [14] Wikipedia, the free encyclopedia. Iso 15288, 2008. http://en.wikipedia.org/wiki/ISO_15288 – last visited on 6th July 2008.
- [15] Alec Dorling. Enterprise spice, 2008. <http://www.enterprisespice.com> – last visited on 6th July 2008.
- [16] Alec Dorling. The spice user group, 2008. <http://www.spiceusergroup.org/> – last visited on 6th July 2008.

Appendix A

ISO/IEC 15504

ISO/IEC TR 15504 [8], also known as SPICE – Software Process Improvement and Capability dEtermination – first appeared on 1997, as a framework for the assessment of processes, developed by the Joint Technical Subcommittee between ISO – International Organization for Standardization – and IEC – International Electro technical Commission.

It has derived from ISO 12207 [11] – an international standard for software lifecycle processes created on 1995, with the purpose of supplying a common structure so that the buyers, suppliers, developers, maintainers, operators, managers and technicians involved with the software development used a common language established in the form of well defined processes – and the ideas of many maturity models like Bootstrap [12] – a project done with the goal to develop a method for software-process assessment, quantitative measurement and improvement –, Trillium [13] – a model that covers all aspects of the software development lifecycle, most system and product development, support activities, and a significant number of related marketing activities – and the CMM – Capability Maturity Model.

ISO/IEC TR 15504 was created because of the many national maturity model proposals existent, establishing an international standard in this area. It is the reference model for maturity models so that the assessors can give an overall determination of the organization’s capabilities for delivering products (as software, systems, and IT services).

When it first appeared, this standard contained a reference model that defined a process dimension and a capability dimension. It also referred to external process lifecycle standards as ISO 12207 mentioned above and ISO/IEC 15288 [14] – a Systems Engineering standard covering processes and life cycle stages that defines processes divided into 4 categories: technical, project, agreement, and enterprise processes.

The process dimension of the reference model in the ISO/IEC TR 15504 defined processes divided into 5 categories: customer-supplier, engineering, supporting, management, and organization.

For each process the standard defined a capability level (table A.1). The capability of processes was measured using process attributes. Each process attribute consisted of one or more generic practices, elaborated into practice indicators.

In a major revision of the draft standard in 2004, the process reference model was removed, so that the standard now specifies the measurement framework and

Capability Level	Meaning
0	Incomplete
1	Performed
2	Managed
3	Established
4	Predictable
5	Optimizing

Table A.1: Capability Levels

can use different process reference models.

For more on this standard please visit [\[15\]](#) and [\[16\]](#).

Appendix B

Capability Levels

A capability level consists of a generic goal and its related generic practices, which can improve the organization's processes associated with those goals and practices process area.

Capability Level 0: Incomplete — An "incomplete process" is a process that either is not performed or partially performed.

Capability Level 1: Performed — A "performed process" is a process that satisfies the specific goals of the process area. It supports and enables the work needed to produce work products.

Capability Level 2: Managed — A "managed process" is a performed process that has the basic infrastructure in place to support the process. It is planned and executed in accordance with policy; employs skilled people who have adequate resources to produce controlled outputs; involves relevant stakeholders; is monitored, controlled, and reviewed; and is evaluated for adherence to its process description.

Capability Level 3: Defined — A "defined process" is a managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures and other process improvement information to the organizational process assets. A major difference between levels 2 and 3 is the scope of standards, process descriptions, and procedures. At level 2 these may be quite different in each instance of the process (e.g., on a particular project); at level 3 these are tailored from the organization's set of standard processes to suit a particular project or organizational unit and therefore are more consistent, except for the differences allowed by the tailoring guidelines.

Capability Level 4: Quantitatively Managed — A "quantitatively managed process" is a defined process that is controlled using statistical and other quantitative techniques. Quantitative objectives for quality and process performance are established and used as criteria in managing the process.

Capability Level 5: Optimizing — An "optimizing process" is a quantitatively managed process that is improved based on an understanding of the common

Capability Levels

causes of variation inherent in the process. The focus is on continually improving the range of process performance through both incremental and innovative improvements.

Descriptions of Capability Levels adapted from [insert ref cmmi book page 47-48].

The capability levels of a process area are achieved through the application of the generic practices associated with that process area: reaching capability level 1 for a process area means that the processes associated with that process area are performed; reaching capability level 2 means there is a policy that indicates you will perform the process: there is a plan, resources, responsibilities assigned, training as needed, selected work products are controlled, ... i.e., the process can be planned and monitored; reaching capability level 3 means an organizational standard process exists associated with that process area, which can be tailored to the needs of the project; reaching capability level 4 means this process area is a key business driver that the organization wants to manage using quantitative and statistical techniques; finally, reaching capability level 5 means that you have stabilized the selected sub-processes and that you want to reduce the common causes of variation within that process.

Appendix C

Goals and Practices

This project was part of an organization's effort to implement level 3 process areas, so the implemented processes were supposed to respond to Generic Goals 2 and 3, as well as their practices.

As the scope of this project were the Technical Solution, Product Integration, Verification and Validation process areas, the processes implemented were supposed to respond to each process area Specific Goals and practices.

C.1 Generic Goals and Practices

- GG 2 – Institutionalize a Managed Process
 - GP 2.1 – Establish an Organizational Policy
 - GP 2.2 – Plan the Process
 - GP 2.3 – Provide Resources
 - GP 2.4 – Assign Responsibility
 - GP 2.5 – Train People
 - GP 2.6 – Manage Configurations
 - GP 2.7 – Identify and Involve Relevant Stakeholders
 - GP 2.8 – Monitor and Control the Process
 - GP 2.9 – Objectively Evaluate Adherence
 - GP 2.10 – Review Status with Higher Level Management
- GG 3 – Institutionalize a Defined Process
 - GP 3.1 – Establish a Defined Process
 - GP 3.2 – Collect Improvement Information

C.2 Specific Goals and Practices

- Technical Solution
 - SG 1 – Select Product Component Solutions

Goals and Practices

- * SP 1.1 – Develop Alternative Solutions and Selection Criteria
 - * SP 1.2 – Select Product Component Solutions
- SG 2 – Develop the Design
 - * SP 2.1 – Design the Product or Product Component
 - * SP 2.2 – Establish a Technical Data Package
 - * SP 2.3 – Design Interfaces Using Criteria
 - * SP 2.4 – Perform Make, Buy, or Reuse Analyses
- SG 3 – Implement the Product Design
 - * SP 3.1 – Implement the Design
 - * SP 3.2 – Develop Product Support Documentation
- Product Integration
 - SG 1 – Prepare for Product Integration
 - * SP 1.1 – Determine Integration Sequence
 - * SP 1.2 – Establish the Product Integration Environment
 - * SP 1.3 – Establish Product Integration Procedures and Criteria
 - SG 2 – Ensure Interface Compatibility
 - * SP 2.1 – Review Interface Descriptions for Completeness
 - * SP 2.2 – Manage Interfaces
 - SG 3 – Assemble Product Components and Deliver the Product
 - * SP 3.1 – Confirm Readiness of Product Components for Integration
 - * SP 3.2 – Assemble Product Components
 - * SP 3.3 – Evaluate Assembled Product Components
 - * SP 3.4 – Package and Deliver the Product or Product Component
- Verification
 - SG 1 – Prepare for Verification
 - * SP 1.1 – Select Work Products for Verification
 - * SP 1.2 – Establish the Verification Environment
 - * SP 1.3 – Establish Verification Procedures and Criteria
 - SG 2 – Perform Peer Reviews
 - * SP 2.1 – Prepare for Peer Reviews
 - * SP 2.2 – Conduct Peer Reviews
 - * SP 2.3 – Analyze Peer Review Data
 - SG 3 – Verify Selected Work Products
 - * SP 3.1 – Perform Verification
 - * SP 3.2 – Analyze Verification Results
- Validation
 - SG 1 – Prepare for Validation

Goals and Practices

- * SP 1.1 – Select Products for Validation
- * SP 1.2 Establish the Validation Environment
- * SP 1.3 Establish Validation Procedures and Criteria
- SG 2 – Validate Product or Product Components
 - * SP 2.1 – Perform Validation
 - * SP 2.2 – Analyze Validation Results

Goals and Practices

Appendix D

SCAMPI B Findings

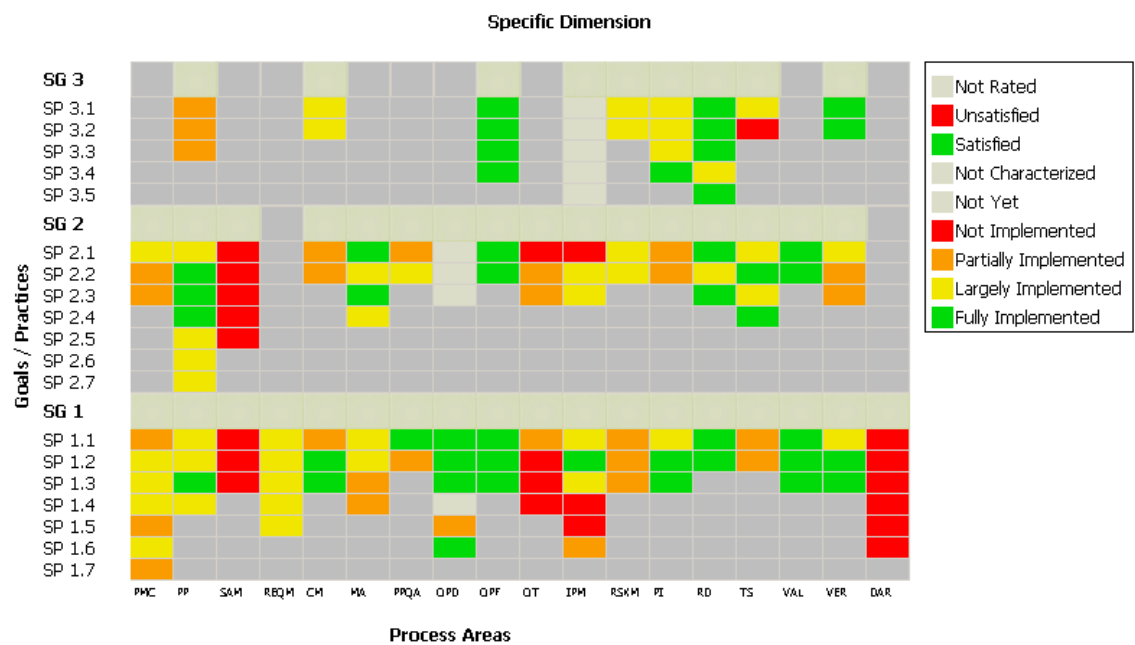


Figure D.1: SCAMPI B findings by *Colin Benton*

SCAMPI B Findings

Appendix E

Release Notes QMS 1.1.1 Template

1. Introduction

1.1 Objective

1.2 Scope

1.3 Audience

1.4 Definitions and acronyms

All the definitions and acronyms used in this document are described in [AD-1].

1.5 References

1.5.1 Applicable documents

[AD-1] DC Quality Management System glossary, Porto DC, QAG-SQI-PRC-2006-0001

[AD-2] < applicable documents >, owner, < code-ref >.

1.5.2 Reference documents

[RD-1] < Applicable documents >.

2. General description

In this section should be described the work products belonging to the product or software package, their current versions and the differences to the last released version.

Artifact Name	Version	N – New E – Enhanced F - Fixed
<artifact name>	<1.4>	<>

Figure E.1: Table 1 - Status

3. Functional changes

In this section should be described all functional changes to the work products referred in section 2. It should also be specified if these changes were made due to problems encountered during the test phase or other implementation decisions.

Artifact Name	Functionality (Added/Changed/Removed)	UTP (if applicable)
<artifact name>	<LDAP Authentication functionality added>	<151872>

Figure E.2: Table 2 - Functional changes

4. Restrictions

In this section should be described all restrictions to every work product considering, among others, configuration and installation issues. As an example, you should describe the steps to be followed to reinstall or upgrade the software.

4.1 Work Product 1

5. Language support

In this section should be described all supported languages.

Languages	Remarks

Figure E.3: Table 3 - Language support

6. Solved problems

In this section should be described all solved problems reported for the work products belonging to the product or software package. For every problem, if applicable, a UTP number should be assigned.

UTP (if applicable)	Description
<115288>	<Bug calculating the final grade.>

Figure E.4: Table 4 - Solved problems

7. Known problems

In this section should be described all known problems that were not solved in this version of the product or software package. Description should include an explanation of how the problem can be recognized by the user and the expected impact or effect.

8. Compatibility

In this section should be described all the compatibility problems (installation, configuration, databases, etc.) with the early version of the product or software package. First state any compatibility requirements or restrictions for current release (if not mentioned in Sect 4)

Appendix F

Release Checklists

Action	Description	Comments
1 Code Release		
1.1 Sub-activity A.1		This is a sub-activity or a step of the activity
1.2 Sub-activity A.2		
1.3 Sub-activity A.3		
2 Documentation Release		
2.1 Update the documentation (User Manual, Admin Manual, Installation Manual)		A typical step of the Documentation Release
2.2 Finish the release notes		A typical step of the Documentation Release
3 Release Finalization		
3.1 For each UTP in the release, re-check its type and artifact(s)		A typical step of the Release Finalization
3.2 Release the UTPs		A typical step of the Release Finalization
3.3 Notify the client that the release is ready		A typical step of the Release Finalization

Figure F.1: Release Checklist Template

Release Checklists

Appendix G

Integration, Verification and Validation Plan



INTEGRATION, VERIFICATION AND VALIDATION PLAN

<PROJECT NAME>

DISCLAIMER
***This document is part of the Quality Management
System, which is issued and controlled by the
global SPEG.***

Date: <yyyy-mm-dd>
Pages: 9
Access: For internal Use Only
Reference: <Document reference>
Clearcase <ClearCase version>
Version :

VERIFY THAT THIS IS THE CORRECT DOCUMENT REVISION BEFORE USE.

Figure G.1: IVV Plan - Page 1

Integration, Verification and Validation Plan



INTEGRATION, VERIFICATION AND VALIDATION PLAN

<PROJECT NAME>

Approval:				
ClearCase Version	Name	Function	Signature	Date
<ClearCase version>	<name>	<function>		<yyyy-mm-dd>
<ClearCase version>	<name>	<function>		<yyyy-mm-dd>

Authors and Contributors:			
Name	Contact	Description	Date
<name>	<mail>	<Author/ Contributor>	<yyyy-mm-dd>

Access List:	
Internal Access	
DL-QPT-IT-PA-DC-ALL	
External Access	
None	

Revision History:			
ClearCase Version	Date	Description	Author
<ClearCase version>	yyyy-mm-dd	<describe document changes, regarding this version>	<name>

Figure G.2: IVV Plan - Page 2

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

TABLE OF CONTENTS

1. Introduction	4
1.1 Overview	4
1.2 Definitions and acronyms	4
1.2.1 Definitions	4
1.2.2 Acronyms	4
1.3 Document structure	4
1.4 References	4
1.4.1 Applicable documents	4
1.4.2 Reference documents	5
2. Integration	5
2.1 Product Components	5
2.2 Integration Sequence and Verifications to be performed during product integration	5
2.3 Product Integration Procedures	6
2.4 Integration Environment Requirements	6
3. Verification	7
3.1 Work Products and methods for the verification of each one	7
3.1.1 Peer Reviews	7
3.1.2 Tests	7
3.2 Test Procedures	7
3.2.1 Unit Tests	7
3.2.2 Integration Tests	8
3.2.3 System Tests	8
3.2.4 UATs	8
3.3 Test Cases	8
3.4 Verification Environment Requirements	8
4. Validation	8
4.1 Overview (user needs to be addressed)	8
4.2 Product or Product components to be validated	8
4.3 Validation methods	9
4.3.1 Discussions	9
4.3.2 Prototypes	9
4.3.3 Documentation	9
4.3.4 User Acceptance Tests	9
4.3.5 Others	9
4.4 Validation Environment Requirements	9

PRINTED ON 08/07/2008

3/9

<DOCUMENT REFERENCE>

Figure G.3: IVV Plan - Page 3

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

1. Introduction

1.1 Overview

<Describe here a brief overview about this document, stating the objective and scope of this plan.>

1.2 Definitions and acronyms

1.2.1 Definitions

<Insert here the most significant definitions within the scope of this document.>

1.2.2 Acronyms

For the purpose of this document the following acronyms are applicable.

<Insert acronym>

<Description>

1.3 Document structure

Section 1, Introduction - provides information about the document content including the overview, definitions and acronyms and structure.

Section 2, Integration – provides a plan for the integration of the product components into the final product, with the identification of the product components, a study of the possible integration sequences and the chosen sequence with a reference to the verifications to be performed during the integration, the product integration procedures to be followed by the team and the integration environment specification.

Section 3, Verification – provides a plan for the verifications to be performed on the product components, work products and final product, identifying with verification methods are going to be used in each work product and the verification environments specification.

Section 4, Validation – provides a plan for the validations to be performed, with an overview of the user needs to be addressed, the identification of the product and product components to be validated and the methods for each validation, along with the validation environments specification.

1.4 References

1.4.1 Applicable documents

[AD-1] *Software Development Process*, Qimonda, <document reference >.

[AD-2] *Glossary*, Qimonda, <document reference >.

[AD-3] *DC Quality Management System*, Qimonda, QAG-SQI-PRC-2006-0001.

[AD-4] *Quality Management System Website*, Qimonda,
http://qShare.qimonda.com/sites/porto_dc/qms.

[AD-5] *Roles and Responsibilities*, Qimonda, QAG-SQI-GDK-2006-0053

[AD-6] *Integration and Test Process*, Qimonda, QAG-SQI-PRC-2008-0967.

PRINTED ON 08/07/2008

4/9

<DOCUMENT REFERENCE>

Figure G.4: IVV Plan - Page 4

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

1.4.2 Reference documents

N/A.

2. Integration

2.1 Product Components

Product components are all the sub-products, with specific functionalities, that are going to be part of a more complex product. All the integrated product components constitute the final product.

They can be identified in a list, a design picture (with the components well identified) or through a reference to where the product components are well identified.

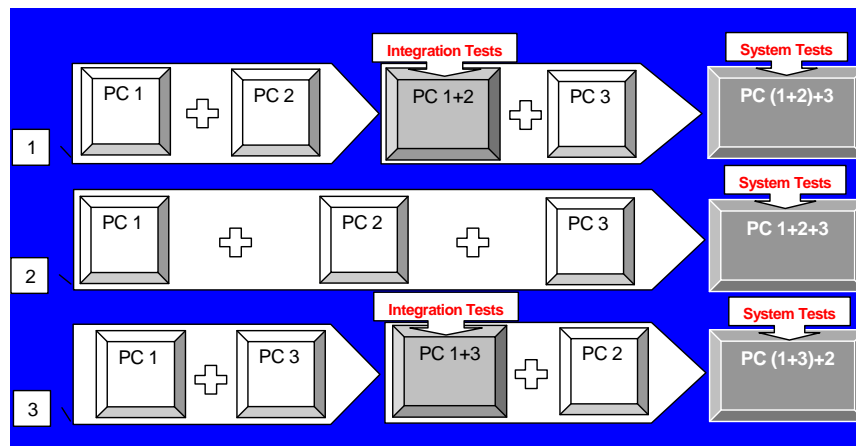
2.2 Integration Sequence and Verifications to be performed during product integration

An integration sequence should be identified (e.g.: in a picture or in a written sequence) with needed specific tools and test equipment, advantages and disadvantages and the verifications timings.

The sequence that is chosen should be reported/recorded along with the explanation of why it was chosen over the others.

Decision Analysis and Resolution (DAR) process can and should be used to evaluate different alternatives to choose the best one.

Example 1 (with picture):



- Sequence 1 has as an advantage the fact that you can verify each component as soon as it is assembled, but it takes more time and more effort.
- Sequence 2 is less time and effort consuming in the integration process but the final product has to be verified as it had never been verified, and is bound to have more bugs than the final product from sequence 1.

PRINTED ON 08/07/2008

5/9

<DOCUMENT REFERENCE>

Figure G.5: IVV Plan - Page 5

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

- *Sequence 3 is similar to sequence 1 but the pc1 and pc3 are more complex to join, so joining them first will facilitate the finding of bugs from the two working together, while in the sequence 1 that could be more difficult and the bugs more difficult to trace to the source.*

*This project will follow the **sequence 2** because the major concern of the customer is the time-frame of the project, and not really the extreme quality of the first version of the product.*

Example 2 (with text):

Sequence 1:

1. $PC1 + PC2 = PC1+2$ (Integration tests)
2. $PC1+2 + PC3 = PC(1+2)+3$ (final product: System tests)
 - Advantages: each component can be verified as soon as it is assembled
 - Disadvantages: takes more time and more effort; as PC1 and PC3 are more complex to join, the bugs could be more difficult to trace to the source in this sequence.

Sequence 2:

1. $PC1 + PC2 + PC3 = PC1+2+3$ (final product: System tests)
 - Advantages: less time and effort consuming in the integration process
 - Disadvantages: the final product has to be verified as it had never been verified, and is bound to have more bugs than the final product from sequence 1

Sequence 3:

1. $PC1 + PC3 = PC1+3$ (Integration tests)
2. $PC1+3 + PC2 = PC(1+3)+2$ (final product: System tests)
 - Advantages: the pc1 and pc3 are more complex to join, so joining them first will facilitate the finding of bugs from the two working together
 - Disadvantages: similar to sequence 1, so takes more time and more effort.

*This project will follow the **sequence 2** because the major concern of the customer is the time-frame of the project, and not really the extreme quality of the first version of the product.*

2.3 Product Integration Procedures

State the procedures to follow to integrate the product components into the final product.

Define the criteria for the delivery of the integrated product.

Note: Refer to the Interface Matrix from DI process.

When joining PC1 and PC2 please change the call to the funtion $f(xpto)$ to $f(xxpto)$.

Product is ready to be released after all system tests have been run and passed their acceptance criteria..

(...)

2.4 Integration Environment Requirements

State the hardware, software and data needs for the product integration.

Note: If the integration environment requirements are quite simple, they can be identified in the Project Profile's Resources and Environment section.

PRINTED ON 08/07/2008

6/9

<DOCUMENT REFERENCE>

Figure G.6: IVV Plan - Page 6

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

3. Verification

3.1 Work Products and methods for the verification of each one

3.1.1 Peer Reviews

Plan the configuration items/work products that should be reviewed.

Work Products	Review Type	Purpose	When?	Responsible
Identify only the configuration items that need to be reviewed	Inspection or Email pass-around or Over-the-shoulder	Explain what you want to achieve with the peer review	Can be a specific date or be related to a phase or milestone	Indicate who will be the initiator of the peer review (author, other?)

3.1.2 Tests

Test object	Work products (title and version) to be tested
Test type	Components, Integration, System, Acceptance, Regression, ...
Test objective	Verify detail specifications, verify system specifications, verify requirement specifications, ... <i>Note: Use the traceability matrix to identify the requirements to be satisfied by each artifact/work product.</i>
Test team	Tester Name 1, Tester Name 2, ...
Qualified approver	Test Manager, SQA, ...
Test strategy	Identify or define overall rules and processes that will be followed in the project
Test procedures	List the test procedures that will be used in the project (their description are detailed on the next section): - Unit Tests - Integration Tests - System Tests - UATs
Restriction	Degree of automation: automatic or manual Constraint on the volume of data
Test location	Path to where the scripts will be saved
Test Report	Path to where the report/log will be saved

3.2 Test Procedures

Identify which types of tests are going to be performed in the project and define the procedures that need to be followed by the project team.

Note: Generic test procedures can and should be available in the Process Asset Library (PAL). Only project specific test procedures should be described in this section.

3.2.1 Unit Tests

After having the work environment ready, after the code has the label "xpto" in ClearCase, test scripts must be created and recorded in folder "xpty" and run using the application "programx", recording the test logs in folder "xptz" and filling-out the test report and update it in ClearCase

PRINTED ON 08/07/2008

7/9

<DOCUMENT REFERENCE>

Figure G.7: IVV Plan - Page 7

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

3.2.2 Integration Tests

3.2.3 System Tests

3.2.4 UATs

See UATs Procedure in Process Asset Library (PAL).

3.3 Test Cases

Test cases can be defined directly in Test Manager or using rtpar files (if so, please identify in this section the path to those files).

If preferable, the following table can be used (one table per test case):

Test Case ID	Version	Status
Details	Product Version	Severity
Description	Please describe here in general words what will be tested here.	
Acceptance Criteria	List here all results from the test case, which will mark it as "Pass".	
Step	Action	Expected behavior

3.4 Verification Environment Requirements

State the specific hardware, software and data needs for the product verification.

Note: If the verification environment requirements are quite simple, they can be identified in the Project Profile's Resources section.

4. Validation

4.1 Overview (user needs to be addressed)

Identify which categories of user needs (operational, maintenance, training, support) are to be validated, and why.

This project's major concern is with the functionality of the product, with the operational needs of the customer and the training needed so the product can be useful to the customer. As so, nothing will be done to validate the maintenance or support needs of the customer.

4.2 Product or Product components to be validated

The Product components that are to be validated with the customer can be the user manual (serves as a non-functional prototype), a functional prototype, the detailed design, the design and the requirements specification, etc.

Work Products to be validated	Issue to be investigated	Related requirements and constraints	Validation Methods
E.g.:User Manual,	NA, clicking on "Cancel"	NA, REQx, Customer is	Refer to next section for

PRINTED ON 08/07/2008

8/9

<DOCUMENT REFERENCE>

Figure G.8: IVV Plan - Page 8

Integration, Verification and Validation Plan

<PROJECT NAME>

INTEGRATION, VERIFICATION AND VALIDATION PLAN

<i>Prototype, Design, Requirements, Others...</i>	<i>should show a confirmation window or not? ...</i>	<i>only available at Monday, from 3p.m. to 5p.m. GMT...</i>	<i>examples.</i>

4.3 Validation methods

Plan which methods or procedures will be followed to communicate with customer to try to assess if our understanding of the requirements is consistent with the customer's needs.

Notes: Refer to the Communication Plan (from Project Profile). Use "NA" if not applicable.

4.3.1 Discussions

Discussions with users (using phone calls, communicator, e-mails or meetings) can and should be used to clarify any doubts that need quick feedback.

It's important to record all decisions or action items that may come up from this discussions using minutes (sent by mail to the customer, for his knowledge and validation) or updating existent project documents and submitting them for review and approval.

4.3.2 Prototypes

Using an iterative lifecycle is a good validation method as the product is progressively delivered to the customer (as BETA releases), with increasing complexity and functionality, so that it can be validated in an early stage.

4.3.3 Documentation

Documentation is a good validation tool as it is a record of how we are planning to actually do things and, when presented to the customer, it can reveal many misinterpretation issues and improvement opportunities.

E.g: User Manuals, Requirements Specifications, Designs, Training Materials, etc.

4.3.4 User Acceptance Tests

Customer supplied products are the easiest validation methods as they reflect directly the users' needs and how the product will be used in the intended environment.

4.3.5 Others

Describe any other validation method that will be used in the project to ensure that "you've built the right thing".

4.4 Validation Environment Requirements

State the specific hardware, software and data needs for the product validation.

Note: If the validation environment requirements are quite simple, they can be identified in the Project Profile's Resources section.

PRINTED ON 08/07/2008

9/9

<DOCUMENT REFERENCE>

Figure G.9: IVV Plan - Page 9

Appendix H

Release Templates

H.1 Word Template



SOFTWARE RELEASE NOTES

<PROJECT NAME>


DISCLAIMER
The Release Notes Guidelines is part of the Quality Management System, which is issued and controlled by the global SEPG.

Date: 0000/00/00
Pages: 8
Access: For Internal Use Only
Reference: <document code>
Clearcase <ClearCase version>
Version:

VERIFY THAT THIS IS THE CORRECT DOCUMENT REVISION BEFORE USE.

Figure H.1: Release Notes template - Page 1

Release Templates



SOFTWARE RELEASE NOTES

<PROJECT NAME>

Approval:				
Clearcase Version	Name	Function	Signature	Date
<ClearCase version>	<name>	Project Manager		<yyyy-mm-dd>

Authors and Contributors:			
Name	Contact	Description	Date
<Name>	<e-mail address>	Author	<date>
<Name>	<e-mail address>	Contributor	<date>
<Name>	<e-mail address>	Reviewer	<date>

Access List:	
Internal Access	
<All / Distribution list>	
External Access	
<None/ Distribution list>	

Revision History:			
Clearcase Version	Date	Description	Author
<ClearCase version>	<yyyy-mm-dd>	<description>	<name>

Figure H.2: Release Notes template - Page 2

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

TABLE OF CONTENTS

1. INTRODUCTION.....	4
1.1 OBJECTIVE.....	4
1.2 SCOPE	4
1.3 AUDIENCE	4
1.4 LANGUAGE SUPPORT	4
1.5 DEFINITIONS AND ACRONYMS	5
1.6 REFERENCES.....	5
1.6.1 <i>Applicable documents</i>	5
1.6.2 <i>Reference documents</i>	5
2. RELEASE ITEMS	5
3. CHANGES	6
4. KNOWN ISSUES	7
5. CONFIGURATIONS	7
6. COMPATIBILITY	8
7. SIGN-OFFS.....	8
7.1 TESTS SUMMARY	7
7.2 SIGN-OFF.....	ERRO! MARCADOR NÃO DEFINIDO.

PRINTED ON 2008-07-08

3/8

<DOCUMENT CODE>

Figure H.3: Release Notes template - Page 3

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

General Guidelines

If there is a section which you feel doesn't apply to the release you're making, fill it with NA (Not Applicable).

Don't forget to fill-in all the headers and footers.

(If you feel the template doesn't suit your project, you may not follow the template but all the content that is in the sections marked as mandatory in the tailoring matrix will have to be present in every release notes you make).

In the end, update the date field, the number of pages field and the table of contents.

1. Introduction

1.1 Objective

Keep the text as it is, but you can add information as needed.

This document provides information about what items are being delivered and all the changes and enhancements that occurred for this release.

1.2 Scope

Substitute <application release version> for the actual name of the application, release and version. You can add information as needed.

e.g.1: The scope of this document is the changes and enhancements present in the BE MES GUI 2.10.0.

e.g.2: The scope of this document is the changes and enhancements present in the twelfth hot fix for QCIM Master Data project iteration 7.

The scope of this document is the changes and enhancements present in the <application release version>.

1.3 Audience

Just write the distribution list or "<name of the application> users".

e.g.1: DL-QPT-IT-PA-MES-MasterData-Release (distribution list)

e.g.2: BE MES GUI users (application users)

Distribution List / Application users / ...

1.4 Language Support

Fill-in the table with the languages supported and the remarks you believe are necessary.

Languages	Remarks
< English>	< Other languages are possible, but only English is included in the release package>

PRINTED ON 2008-07-08

4/8

<DOCUMENT CODE>

Figure H.4: Release Notes template - Page 4

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

Table 1 - Language Support

1.5 Definitions and acronyms

If you want you can add other acronyms, using the following text and table:

"For the purpose of this document, the following acronyms are applicable: "

Acronym	Definition
<acronym>	<definition>

Please verify if there isn't any acronym already defined that you may use.

All the definitions and acronyms used in this document are described in [AD-1].

1.6 References

1.6.1 Applicable documents

An applicable document is an internal document capable of being applied, having relevance for the purpose of this document.

The reference should have the format: applicable document, owner, code-ref.

[AD-1] DC Quality Management System glossary, Porto DC, QAG-SQI-PRC-2006-0001

[AD-2] <Applicable document>, <owner>, <code-ref>.

1.6.2 Reference documents

A referenced document is an external document that was consulted and/or was a source of relevant information to the writing of this document.

If the reference is for a document, it should have the format: Referenced document, author, publisher, (version if applicable), date of publication.

If the reference is for a webpage, it should have the format: Referenced webpage, author, URL, date of the last consultation.

[RD-1] <Referenced document>, <author>, <publisher>, <(version if applicable)>, <date of publication>

[RD-2] <Referenced webpage>, <author>, <URL>, <date of the last consultation>

2. Release Date

yyyy-mm-dd

3. Release Items

This section is an overview of the release. Here should be a list of all the work products (items / artifacts / ...) of the release. The table helps to list the items, and allows the reader

PRINTED ON 2008-07-08

5/8

<DOCUMENT CODE>

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

to see the version of the artifact included and the status of each work product regarding the last released version (new, enhanced and/or fixed).

You may divide this chapter into sub-chapters according to work packages.

Component/Module Name	Version	N – New/E – Enhanced/F – Fixed
<artifact name>	<1.4>	<E/F>

Table 2 - Release Items

4. Changes

List all the UTPs (or requirements) addressed in this release, identified by an UTP number or an ID of the requirement. In the “Description” field you may write the title of the UTP or requirement or, if you feel that isn’t enough, a short description instead. In the “Artifact Affected” field, you should list, for each UTP or requirement, the artifacts affected by it. The Issue Type field may be filled with the CR type or with one of the following:

Possible issue types: Code Bug, Deployment, Design Bug, Enhancement, Modification, New, Refactoring

You may divide this chapter in sub-chapters, one for each listed change, in order to further explain the change:

ID	Description	Artifacts Affected	Issue Type
F30000118330	CheckListModel Dialog shall read PartNumbersQty	CheckListModel	Design Bug
F30000119140	Translate messages to Chinese		New

3.1 F30000118330 - CheckListModel Dialog shall read PartNumbersQty

MMSGUI is now considering the UDA name PartNumbersQty instead of PartNumbers_Qty as previously. This is related with CheckListModel wizard.

3.2 F30000119140 - Translate messages to Chinese

Now it is possible for the user choose Chinese language once all error, warning and information messages are translate do Chinese.

ID	Description	Components/Modules Affected	Issue Type
<Id or UTP number>	<UTP title or a short description >	<list of artifacts affected, separated by a semi-colon (;)>	<CR Type or a type from the list on the guidelines>

PRINTED ON 2008-07-08

6/8

<DOCUMENT CODE>

Figure H.6: Release Notes template - Page 6

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

Table 3 - Changes

5. Known Issues

List all known problems that were not solved in this version of the product or software package. The description of the known issues should include an explanation of how the problem can be recognized by the user and the expected impact or effect.

To list the problems you may present first an overview in a table:

ID (if applicable)	Description	Components/Modules Affected	Issue Type
<ID/UTPhr>	<UTPtitle/short description of the problem>	<list of components/modules affected, separated by a semi-colon (;)>	<CRtype or one of the issue types list>

and then create sub-chapters, one for each problem, in order to explain the issue.

Possible issue types: Code Bug, Deployment, Design Bug, Enhancement, Modification, New, Refactoring

Or you can skip the table and just create the sub-chapters. This is similar in format to the chapter Changes.

6. Test Results

6.1 Tests Summary

Fill-out the table with the information related to the tests. If you have an excel summary table you can import it as long as you format it to look like the one in this chapter and as long as it has the same fields. If you do not have an excel summary table, fill the table from this template with the data of your test results. The fields with the letters of a different color are example of non mandatory fields that you may add to the table (if you don't use them, delete them); but if you do use them, change the color of the letters to "automatic".

WorkPackage	Integration Test Cases					Code Coverage	
	Available	Valid	Executed	Passed	Execution %	Pass %	By Test Cases
<Work Package 1>	9	4	4	4	100.00%	100.00%	58.55%

6.2 Test Results

Add a reference to where the test results/report can be found or, for simple test results, fill-out the table:

Test Case Name	WorkPackage	Component / Module	Available	Valid	Run Status	Run Date	Comments
ApproveCheckList	<Work Package 1>	IFXApi	yes	Updated	Pass	2/6/2008	
CMSE_Validate_AlternateProcedure	<Work Package 1>	Workflow	yes	Outdated	Inconclusive	2/13/2008	Testcase outdated because in the case specified the testcase the workflow returns success.plnAlternateProcedure = "OR doesn't exist" > flow continues to CreateProcedure.plnAlternateProcedure = > continue normal flow

PRINTED ON 2008-07-08

7/8

<DOCUMENT CODE>-

Figure H.7: Release Notes template - Page 7

Release Templates

<PROJECT NAME>

SOFTWARE RELEASE NOTES

7. Configurations

Describe all the configurations needed, in ClearCase, for the installation of the application, and others.

You should use the following chapters as applicable:

9.1 ClearCase specifications

List all the configurations needed for ClearCase (Views, VOBs ...).

9.2 Installation

List all the installation related configurations (Databases installations, on application servers, on terminal servers ...).

8. Compatibility

Describe all the compatibility issues with the early version of the product or software package or with other applications or frameworks.

e.g.: QlApi changes, lfxApi changes...

9. Sign-Off


Insert the names of the Project Manager, the CM and the SQA and the date of their approval.

This software and/or documentation have been thoroughly tested and/or reviewed, are considered to be a quality product, and are authorized for release.

Project Manager	<name>	<date>
CM	<name>	<date>
SQA	<name>	<date>

Figure H.8: Release Notes template - Page 8

H.2 Excel Template

Version:	ProFAB <number of version>	
Description:	<type of version> <i>Incremental baseline</i>	
Release date:	<yyyy-mm-dd>	
Content:	This version includes the following capabilities: Artifacts released included in the Overview sheet	

Work Package	Available	Valid	Integration Test Cases			Execution %	Pass %	Code Coverage	
			Executed	Passed				by Testcases	Total
<Work Package 1>	9	4	4	4	100.00%	100.00%	58.55%	72.24%	
<Work Package 2>	9	3	3	3	100.00%	100.00%	58.55%	72.24%	
<Work Package 3>	9	1	1	1	100.00%	100.00%	NA	NA	

These Release Notes include the following sheets:	Sign-Off						
FrontPage ClearCase view configuration Overview Test Results Code Coverage	<table border="1"> <tr> <td>PM</td> <td><name></td> </tr> <tr> <td>CM</td> <td><name></td> </tr> <tr> <td>SQA</td> <td><name></td> </tr> </table>	PM	<name>	CM	<name>	SQA	<name>
PM	<name>						
CM	<name>						
SQA	<name>						

Figure H.9: Release Notes template - FrontPage

ClearCase VOB:	<i>VFAB300_BaseModel_VOB</i>
To see only the artifacts that were changed (new/updated) for this version use this configuration in a ClearCase view:	
<pre>element -directory * ... \main\common\LATEST -nocheckout element * ProFAB_5.1_BETA1 -nocheckout</pre>	
To see all artifacts that are part of the ProFAB/Base Model releases until this version use this configuration in a ClearCase view:	
<pre>element -directory * ... \main\common\LATEST -nocheckout element * ProFAB_5.1_BETA1 -nocheckout</pre>	
Important Notes	
<p><i>Please be aware that the deployment of this ProFAB 5.1 BETA 1 release into a test system requires previous deployment of ProFAB 5.0.1 BETA2 in that system (and all ProFAB former releases after ProFAB 5.0).</i></p>	

Figure H.10: Release Notes template - ClearCase view Configuration

Release Templates

Capability	Type	CC Path	Artifact	Version	Change Requests	Notes
<Work Package 1>	Workflow	EquipmentManagement\Equipm ApproveCheckList.xml		1.65	UTP 119278	
<Work Package 1>	Workflow	EquipmentManagement\Equipm CheckForPendingCheckLists.xml		1.38	UTP 119284	
<Work Package 1>	Workflow	EquipmentManagement\Equipm CreateCheckListModel.xml		19.7	UTP 119288	
<Work Package 2>	Workflow	EquipmentManagement\LogMaterialTrackingEvent.xml		4.1	UTP 119635	
<Work Package 3>	Template	ProductManagement\CSIM_LO1CSIM_LOT_TPL_FOR_UDA.xml		190	UTP 119766, UTP 119859, UTP 120003, UTP 120103	
<Work Package 2>	IFXAPI	IFX-API\MES\ProductManagem LogMaterialTrackingEvent.api		121	UTP 118810	
<Work Package 2>	CRM	DurablesManagement\PanelCou PanelTargetAndComplexity.xml			UTP 119636	
NA	CRM	GlobalDefinitions\GlobalService WorkflowErrorCodeBase.xml			Archive and Delete existing CRM version before importing!	
NA	CRM	GlobalDefinitions\GlobalService WorkflowErrorCodes.xml			Archive and Delete existing CRM version before importing!	

Figure H.11: Release Notes template - Overview

Description					Status		
Test Case Group	Test Case Name	Work Package	Artifact	Validat	Run Status	Run Date	Comments
ApproveCheckList	ApproveCheckList	<Work Package 1>	IFXApi	Updated	Pass	08-02-2008	
ApproveCheckList	ApproveCKL_Approve	<Work Package 1>	Workflow	Updated	Pass	13-02-2008	
ApproveCheckList	ApproveCKL_ModeNotValid	<Work Package 1>	Workflow	Updated	Pass	13-02-2008	
ApproveCheckList	ApproveCKL_Request	<Work Package 1>	Workflow	Updated	Pass	13-02-2008	
CreateMaintenanceScheduleForEquipmen	CMSE_Validate_AlternateProcedur	<Work Package 1>	Workflow	Outdated	Inconclusive	13-02-2008	
CSIM_LOT_TPL_FOR_UDA	CSIM_LOT_TPL_FOR_UDA	<Work Package 3>	Template	NotVerified	Pass	13-02-2008	
LogMaterialTrackingEvent	LogMaterialTrackingEvent	<Work Package 2>	IFXApi	Updated	Pass	14-02-2008	
LogMaterialTrackingEvent	LogMaterialTrackingEvent_Fail	<Work Package 2>	Workflow	Updated	Pass	13-02-2008	
LogMaterialTrackingEvent	LogMaterialTrackingEvent_Fail2	<Work Package 2>	Workflow	Updated	Pass	13-02-2008	

Figure H.12: Release Notes template - Test Results

ID (if applicable)	Description	Artifacts Affected	Issue Type
<ID/UTPnr>	<UTPtitle/short description of the problem>	<list of artifacts affected, separated by a semi-colon (;)>	Code Bug
			Deployment
			Design Bug
			Enhancement
			Modification
			New
			Refactoring

Figure H.13: Release Notes template - Known Issues

Code Coverage - ProFAB <version>			
This Code Coverage sheet represents the percentage of Workflow code executed during testing in Porto DC, distinguished by official Testcases code coverage and total code execution with additional testing by Porto DC.			
Capability	Workflow	Code Coverage by Testcases	Total Code Coverage
<Work Package 1>	ApproveCheckList	61%	64%
<Work Package 1>	CheckForPendingCheckLists	69%	69%
<Work Package 1>	CreateCheckListModel	60%	69%
<Work Package 2>	LogMaterialTrackingEvent	44%	86%

Figure H.14: Release Notes template - Code Coverage