SPL-CMM: SOFTWARE PRODUCT LINE CAPABILITY MATURITY MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY
RECEP BORA ÇALIŞKANBAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INFORMATION SYSTEMS

JULY 2019

Approval of the thesis:

SPL-CMM: SOFTWARE PRODUCT LINE CAPABILITY MATURITY MODEL

submitted by **RECEP BORA ÇALIŞKANBAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, Graduate School of **Informatics** __________________

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department, **Information Systems** __________________

Assoc. Prof. Dr. Altan Koçyiğit
Supervisor, **Information Systems Dept., METU** __________________

Assist. Prof. Dr. Ebru Gökalp
Co-Advisor, **Tech. and Knowledge Man. Dept., Baskent University** __________________

Examining Committee Members:

Assoc. Prof. Dr. Aysu Betin CAN
Information Systems Dept., METU __________________

Assoc. Prof. Dr. Altan KOÇYİĞİT
Information Systems Dept., METU __________________

Assist. Prof. Dr. Ebru GÖKALP
Technology and Knowledge Man. Dept., Baskent University __________________

Assoc. Prof. Dr. P. Erhan EREN
Information Systems Dept., METU __________________

Assist. Prof. Dr. Tülin ERÇELEBİ AYYILDIZ
Computer Engineering Dept., Baskent University __________________

Date: **01-07-2019**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Recep Bora Çalışkanbaş

Signature:

ABSTRACT

SPL-CMM: SOFTWARE PRODUCT LINE CAPABILITY MATURITY MODEL

Çalışkanbaş, Recep Bora
M.S., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Altan Koçyiğit
Co-Advisor: Assist. Prof. Dr. Ebru Gökalp
July 2019, 102 pages

Software companies show increasing attention to Software Product Line (SPL) approach. SPL provides cost reduction, quality improvements, and reduced delivery time. Because of these benefits, companies want to assess their current level of SPL and to achieve a guidance for improvement. It is a relatively new area of research in which, so far, limited work has been done. These limited studies are based on Business, Architecture, Process and Organization (BAPO) model. In order to check the applicability, usefulness and completeness of the BAPO model, a pilot study and an exploratory case study have been conducted. Based on the feedbacks from these case studies, the Modified BAPO has been achieved. For the validation of the Modified BAPO, an explanatory case study has been conducted. The results indicated that the Modified BAPO is still insufficient, incomplete and not useful. In order to provide a solution that satisfy these requirements, a SPL Capability Maturity Model (SPL-CMM) has been developed based on SPICE-ISO/IEC TR 15504. SPL-CMM includes SPL-Process Reference Model (SPL-PRM) consisting of 16 SPL specific process definitions under four process areas of business, architecture, technical and organization as well as a measurement framework providing objective ratings. It is a structured and standardized approach that enables assessment of the SPL specific processes in a consistent, repeatable manner. It is assisted by adequate measures with guidance on actions for improvement. The validation of the proposed model has been performed through conducting a case study. The case study results show that the SPL-CMM is applicable for identifying the current state of the SPL process capability and the gaps for process improvement to the next capability level.
Keywords: Software Product Line, Capability, Maturity, Assessment Model

ÖZ

YÜH-YOM: YAZILIM ÜRÜN HATTI YETENEK OLGUNLUK MODELİ

Çalışkanbaş, Recep Bora

Yüksek Lisans, Bilişim Sistemleri

Tez Yöneticisi: Doç. Dr. Altan Koçyiğit

Tez Eş Danışmanı: Dr. Öğretim Üyesi Ebru Gökalp

Temmuz 2019, 102 Sayfa

Yazılım şirketleri, belirli bir müşteri bölümü için belirlenmiş ortak bir temel altyapı üzerinde geliştirilmiş ortak özellikleri paylaşan yazılımlar geliştirmek için Yazılım Ürün Hattı (YÜH) yaklaşımına artan şekilde ilgi göstermektedir. YÜH, maliyet düşüşü, kalite artışı ve kısa teslim süresi gibi faydalar sağlamaktadır. Bu nedenle, şirketler mevcut YÜH seviyelerini değerlendirerek, iyileştirme için rehberlik istemektedir. Bu alan, şimdiye kadar sınırlı çalışma yapılmış, nispeten yeni bir araştırma alanıdır. Bu sınırlı çalışmaların çoğu BAPO (İş Mimari Süreç Kuruluş – Business Architecture Process Organization) çatısına dayanmaktadır. Bu tez kapsamında, BAPO'nun uygulanabilirliğini, kullanışlılığını ve eksiksizliğini kontrol etmek için, bir pilot çalışma ve bir keşif amaçlı durum çalışması yapılmıştır. Bu çalışmalardan elde edilen geri bildirimlere dayanarak yapılan değişikliklerle, Değiştirilmiş BAPO elde edilmiştir. Değiştirilmiş BAPO'nun onaylanması için bir vaka çalışması yapılmıştır. Sonuç, Değiştirilmiş BAPO'nun hala yetersiz, eksik ve kullanışsız olduğunu göstermiştir. Bu gereklilikleri karşılayan bir çözüm sağlamak için, bu tez çalışması kapsamında SPICE-ISO / IEC TR 15504'e dayanarak geliştirilen YÜH-Yetenek Olgunluk Modeli (SPL-CMM) önerilmektedir. SPL-CMM, İş, Mimari, Teknik ve Kuruluş Süreç Alanları altında gruplanmış toplam 16 YÜH özgü süreç tanımını kapsayan YÜH Süreç Referans Modelinin yanı sıra nesnel değerlendirme sağlayan bir ölçüm çerçevesi içermektedir. YÜH'na özgü süreçlerin tutarlı ve tekrarlanabilir bir şekilde değerlendirilmesini sağlayan yapısal ve standart bir yaklaşımdır. Ayrıca, iyileştirme faaliyetlerinde rehberlik edecek doyurucu ölçütler ile de desteklenmektedir. Önerilen modelin geçerliliği bir durum çalışması ile incelenmiş olup, sonuçlar, SPL-CMM'nin, YÜH mevcut durumunu ve bir üst seviyeye ulaşmak için yol haritasını belirlemek için uygulanabilir olduğunu göstermiştir.

Anahtar Kelimeler: Yazalım Ürün Hattı, Yeterlilik, Olgunluk, Değerlendirme Modeli

Dedicated to my beloved family and my wife…

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors Assist. Prof. Dr. Ebru Gökalp and Assoc. Prof. Dr. Altan Koçyiğit for their inexhaustible support throughout my thesis. I am very grateful for their inspiring ideas and insightful discussions about the study. Without their contributions and guidance this would not be possible.

I would like to thank the organizations in which case studies were conducted and the employees working in this organizations for their contributions to case studies.

I would like to express my very special gratitude to my wife, Burcu Mert Çalışkanbaş, for her endless patience and support and for always being there for me.

Finally, I wish to thank to my beloved family. I am very grateful to my parents Sıdıka and Selçuk Çalışkanbaş for supporting me my whole life at each and every aspect. I am very grateful to my sister, Güneş Çalışkanbaş, for her support and always cheering me up.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BP: Base Practice

CMMI: Capability Maturity Model Integrated

F.A.: Fully Achieved

GP: Generic Practice

GPI: Generic Practice Indicator

ISO: International Organization for Standardization

L.A.: Largely Achieved

METU: Middle East Technical University

N.A.: Not Achieved

P.A.: Partially Achieved

PA: Process Attribute

SPICE: Software Process Improvement and Capability Determination

SPL: Software Product Line

SPLA: Software Product Line Architecture

SPL-CMM: Software Product Line Capability Maturity Model

CHAPTER I

INTRODUCTION

High product variety increases production and distribution costs. Modular product design and exploiting commonality are some of the utilized approaches to maximize the benefits while offering a variety of products[1]. By exploiting the commonality with approaches like product platforms and product families, companies tried to tackle these costs. The term product family is defined as *"a set of similar products that are derived from a common platform and yet possess specific features/functionality to meet particular customer requirements"*[2]. Like the existence of product families and product platforms in product development and production, Software Product Line (SPL) and software product families exist in software development. SPL can be described as *"a set of software intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment and developed from a common set of core assets in a prescribed way"*[3]. The benefits of commonalities and reuse are also highly utilized in software development. Similar rules in product line engineering also apply to software. SPL have commonalities which in turn implies the existence of the variabilities and variation points which is a crucial design concept in software development. This, in turn, enables the realization of configurable software. This configurability is important since it supports reuse and lets developers easily adopt services related to the needs of their applications [4].

The SPL is receiving an increasing amount of attention from software development organizations because of the promising results in cost reduction, quality improvements, and reduced delivery time. SPL assessment is a relatively new area of research. Hence, limited work has been done in this field. Currently, researchers from both academia and industry are attempting to develop a prescribed and a systematic way of measuring the capability/maturity of a SPL processes[5].

A process capability maturity model (PCMM) provides a roadmap for implementing the vital practices for one or more domains of processes performed in the organization.

It covers the key elements of effective processes of disciplines. It represents stages or levels of process capability [6]. As a result of determining current capability level of the process based on the assessment results, a road-map of improvements is achieved. By performing the actions determined in the road-map, the process will be improved. There are various well-accepted PCMMs, such as Software Process Improvement and Capability Determination (SPICE) which is published as the standard of ISO/IEC TR 15504, CMMI (Capability Maturity Model Integration). These models are used as an evaluative and comparative basis for process improvement and assessment. Higher process capability and organization maturity is associated with better performance. Observed benefits of these models include cost savings, more involved employees, improved and predictable quality as well as increased productivity, generating a consistency of process capture and use [7]. Customizing SPICE to different sector is subject of growing interest in the literature, as MEDI-SPICE, Automotive-SPICE, Enterprise-SPICE, etc.

Although there are limited number of studies related to measuring SPL process capability/maturity are in the literature, none of existing studies is developed based on a well-known PCMM. The purpose of this study is to develop the SPL Capability Maturity Model (SPL-CMM) based on SPICE by developing SPL Process Reference Model (SPL-PRM) including process definitions for SPL engineering domain. The aim of SPL-CMM is to provide a basis for improving the SPL engineering processes.

SPL-CMM consists of two dimensions, process and capability. Process dimension includes SPL-PRM covering four main process areas of Business, Architecture, Technical and Organization, and 16 critical SPL processes defined in these four process areas. The process definitions include measurable objectives of a process: Process Outcomes, Base Practices (BPs), and Work Products which are constructed based on SPICE. The capability dimension includes Capability levels and Process Attributes (PAs) adapted from SPICE. The PAs representing measurable characteristics necessary to improve the process are applicable to all processes.

The significance of the study is that the SPL-CMM including SPL specific process definitions and a measurement framework enables objective rating. It enables determinıng the current process capability level and provides a road-map for process improvement. SPL-CMM pursues to provide a structured and standardized approach by assessing the SPL specific processes. The motive is to be able to perform improvement initiatives in a consistent and repeatable way, supported by adequate measures with guidance for improvement.

As described in Figure 1, the followed research methodology starts with the problem identification. The necessity of a capability/maturity model for making an assessment of the current state and providing a structured road map for improvement in SPL domain has been identified by the author as a software architect working in the industry. Then, the literature was reviewed if there is a model satisfying the requirements and it was figured out that there are a limited number of studies in this domain, and most of the existing SPL Maturity Models depend on the model of BAPO[5], [8], [9].We conducted a pilot case study in Organization-A to check understandability, completeness and applicability of BAPO. The received feedbacks from process owners indicated that the model is insufficient, hard to understand and incomplete. For instance, testing process is omitted in the model but it is important for SPL. After collecting feedback from the pilot study, we conducted an exploratory case study to measure SPL Maturity Level of Organization-A. The results were correlated with pilot study results for insufficiency, incompleteness and understandability, and these results guided us in determining improvement areas of the BAPO. Based on these improvement areas, we modified BAPO to achieve the model called as Modified BAPO. Then, we conducted an explanatory case to check the applicability, completeness and usefulness of Modified BAPO. The results indicated that the Modified BAPO is still insufficient and not useful. After this experience, we concluded that modifying BAPO will not solve our problem. There is a need for a PCMM specifically developed for SPL domain based on a well-known PCMM. By taking into consideration of this necessity, we have developed SPL-CMM by customizing SPICE. After development of SPL-CMM, a case study in Organization-B was conducted to check the applicability, usefulness and completeness of SPL-CMM by determining the capability level of 16 SPL specific processes and to providing a guideline for process improvement. The results show that the SPL-CMM is applicable for identifying the current state of the process capability and the gaps with the assessed capability level of the SPL processes. We answered the following research questions in the light of the case studies:

RQ1: How suitable it is to use the SPL-CMM with the purpose of identifying the current SPL specific process capability level and how well it provides a roadmap for the process improvement?

RQ2: What are the strengths and weaknesses of the SPL-CMM?

The rest of this thesis is organized as follows: Chapter two is a review of the literature in SPL process improvement methods, their strengths and weaknesses. Chapter three describes the structure and components of Modified BAPO. Chapter four describes the case studies including pilot and exploratory case studies for BAPO, explanatory case studies for Modified BAPO for validation purposes. Chapter five describes the structure and components of SPL-CMM proposed in this thesis. Process Descriptions of SPL-CMM are provided in Appendix. Chapter six describes study SPL-CMM application via a case study for validation purposes. Chapter seven describes the overall findings, achievements and possible directions for future work.

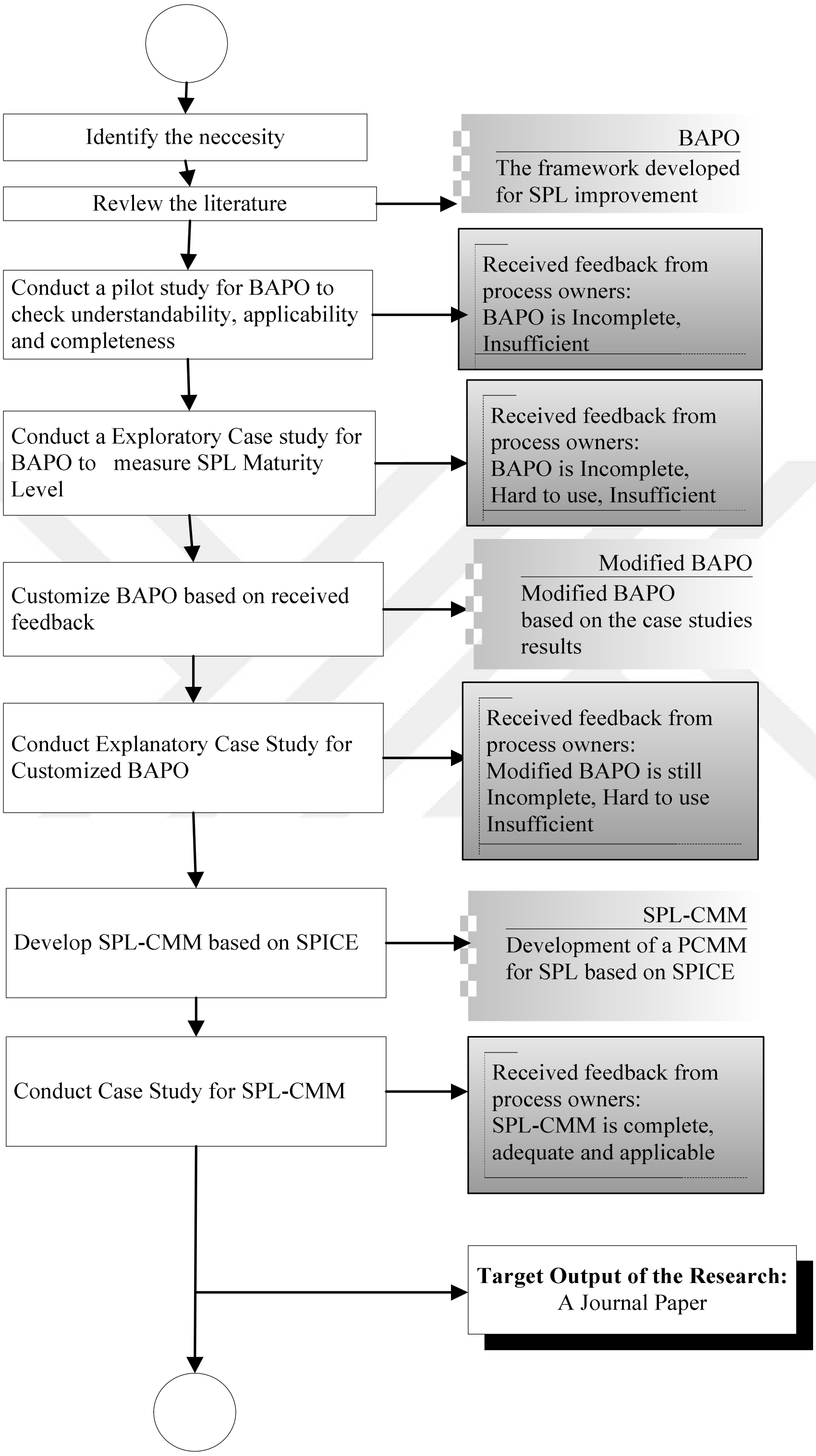

Figure 1 Research Methodology

CHAPTER II

LITERATURE REVIEW

This chapter contains the related literature. The concept of software product line engineering, presented capability and maturity models for software product lines and widely known capability maturity models are investigated. In sections 2.1, 2.2 and 2.3, the concept of software product line engineering is presented. Section 2.4 contains the software product line capability and guideline frameworks. The BAPO model is explored in detail in section 2.5. And in the last sections, 2.6 and 2.7, widely used capability maturity frameworks SPICE and CMMI are presented.

2.1. Product Platforms and Product Families

Before diving into software side of the context, it can be useful to define the terms “Product Platform” and “Product Family”. In a generic manner, why are platforms and families are utilized should be investigated. With the increase in demands and variety of needs, products need to vary, too. This, of course, comes with a cost

To have a better visualization, mobile phones can be given as an example. The variety of mobile phone models of the same manufacturer forms a product family. As defined, they all are similar products with similar specifications, yet they all differ from each other at some respect. The product family as a whole aims to cover the targeted market segment. As in mobile phones example, a manufacturer may try to target all customers, then the targeted segment of the product family is all the possible customers. However, another manufacturer may only produce high end products or just low cost products and target only a portion of the market with their product families. Products themselves on the other hand, aim for the smaller portions of the targeted market segment individually.

The term product platform is defined as “a set of subsystems and interfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced”[2]. There exist specific design rules for each product platform during design and development. Baldwin and Clark divides these rules into 3 categories; platform architectures, interface protocols and standards[10]. Different sorts of product platforms may serve different purposes. A modular platform enables the production of variations by means of configuration, a scalable platform enables the production of variations with the same functionality but with different capacities, a generational platform leverages the product life cycle and enables the rapid next-gen development [11].

Product platforms and families are not stationary assets either. With the advancements in technology and the changes in market needs, product platforms and families also need to evolve. Some initial features and elements may get obsolete while newer ones are introduced. This is the natural life cycle of a product or component. With the so called birth and the death of the products and product elements, product platforms and families evolve. This is visualized with Figure 2 Product Family Evolution p33[12] from El Maraghy.

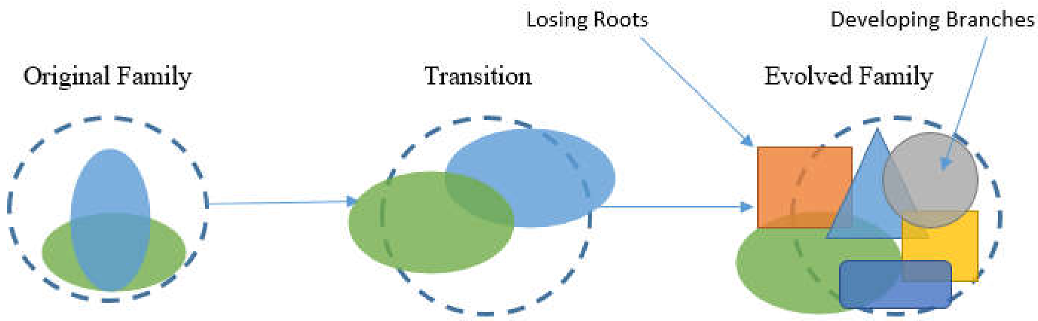

Figure 2 Product Family Evolution p33[12]

The commonality seems to be in the core of these approaches. But still, there is an issue that needs to be mentioned. Variations are as important as the commonalities since there is also the marketing and customer side of this picture. Exploiting commonality does not always come with cost reductions and happy customers. There are always market expectations regarding the prices for their needs and product differentiations. There is a good example to explain this. The memory chips are used for message recording in machines with higher prices whereas cassette tapes are used in lower-price machines. If the same chips are also used in the low-end products, it will increase the costs and shadow the merits of the high and products[1]. This might seem a bit old, but it is self-explanatory. Commonality does not mean using the same element for all the similar purposes and products. There is always cost efficiency side for the needs of the product and the market needs to be convinced why they are paying more for a product than another.

2.2. Software Product Lines

Management is a key feature in SPLs, meaning that it is not simply ad hoc reuse, a concept that can be easily misunderstood. Not every reuse simply means the existence of a SPL. As described by Clements et al “in a SPL approach, the reuse is planned, enabled, and enforced—the opposite of opportunistic"[13]. As in product platforms, SPLs also targets a specific market segment. It should be structured, otherwise it may lead to an unmaintainable code base. And as in the last words, the software is developed from the core assets. This implies the existence of core asset development and application specific development. The application engineering develops different products by configuring the related common parts and implementing product specific extensions where necessary[14]. The distinction between these two concepts plays a key role in SPL engineering.

2.3. SPL Engineering Foundations

2.3.1. ITEA and ESPRIT Projects

The frontier researches on SPL engineering and software product families are done within the consortium of ESPRIT and ITEA, Information Technology for European Advancement. Two projects were held within ESPRIT, namely ARES, Architectural Reasoning for Embedded Software, and Praise. These consortiums were working on the concept of software product family development and evolved into the following ITEA projects afterwards[15]. A series of projects were held by ITEA project groups each lasting a few years. The first project ESAPS, "Engineering Software Architectures, Processes and Platforms for System Families", was carried out between 1999 and 2001. Many project partners like Philips, Bosch, Nokia, Siemens and Thales participated in this project. Right after the end of this project, it was then proceeded by another ITEA project between 2001 and 2003, Café, From Concept to Application in System-Family Engineering. Similarly, after Café, Families projects was initiated, FAct-based Maturity through Institutionalization Lessons-learned and Involved Exploration of Systems, which also was planned for two years as its predecessors. It was carried out between 2003 and 2005. The topic under investigation with ESAPS was the progresses in methods, technology and tools for developing families of products, in the areas of analysis, definition and evolution of system families. In the second study, namely Café, the topic was combining the separate concepts from the ESAPS project into a unified whole, covering the entire lifecycle of a product family. The third of the series was held to consolidate the work done in ESAPS and CAFE in practical businesses, architectures, processes and organizations connected to family development. The details can also be accessed through ITEA website. Figure 2 shows the overview of the participants and Figure 3 shows the project topics of the projects.

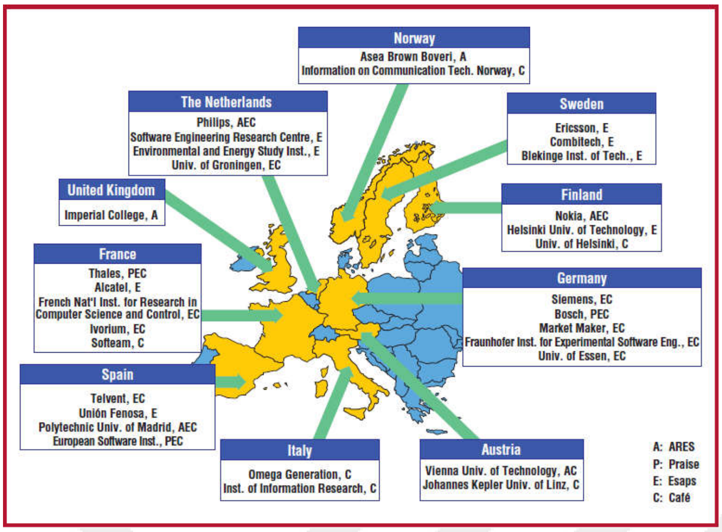

Figure 3 ARES, Preise, Esaps, Café projects participants[15]

Project topics

	Business	Architecture	Process	Organization
ARES		Dealing with variation Architecture description Resource management qualities	Recovery from legacy systems	
Praise		Domain-specific architectures Family architecture Development tools	Family development practices Variability and commonality Traceability between assets Architectural decisions	
Esaps	Scoping	Domain analysis Aspect analysis Family requirements Architecture glossary Commonality and variability Reference architecture Platform and components	Architecture assessment Architecture recovery Domain analysis Aspect analysis Family development process frameworks Requirements modeling and traceability Change management Evolution support Variant configuration and derivation	Platform and component development
Café	Business and market analysis Scoping Family development transition and adoption	Requirements engineering Heterogeneous platforms COTS use Design for quality Development tool support Test modeling Validation	Asset management Traceability Change management and impact analysis Family transition and adoption Configuration and version management Product derivation Family evolution Test strategy and methodology Validation	Asset management Validation and testing Product-line transition and adoption Change management Configuration and version management Product derivation

Figure 4 ARES, Preise, Esaps, Café projects interest areas[15]

2.3.2. BAPO

The acronym BAPO stands for the four aspects of SPL development, namely Business, Architecture, Process and Organization. This Acronym was first introduced by Henk Obbink from Philips Research. These aspects are described as[15]:

- Business: the way the end products make profit

- Architecture: the technology required to build the system
- Process: responsibilities and dependencies of the software development
- Organization: the organization in which the software is developed

These four concerns are actually interdependent rather than orthogonal. Improvements in one, generally leads to improvements in others, too. For example, if your architecture, processes and organization is good, also having a good business value seems quite reasonable. As an opposite example, it would be surprising to have a good business value with bad software architecture, organization and processes.

2.4. SPL Maturity

Researches in the field of SPL field are done to add value to what is being done. It is not only the case for software but the same actually goes for almost everything. It is desired to make improvements at what is done and how it is done. Eventually, the motivation behind all these is to compete better in the market. Surely there are perfectionist reasons in engineering to be better and get satisfied from what you do. Nevertheless, these are funded by companies to add business value to themselves. And when an improvement is introduced, the results are desired to be assessed. With each improvement, the benefits are investigated. This is the way to learn how good you are practicing at something. This leads to the concept of maturity. With the concept of maturity levels, one can assess how well these introduced improvements are utilized.

2.4.1. Staged and Continuous Representations

There are two possible maturity representations in maturity modelling, continuous and staged. In this study, the main focus is on the staged model. Maturity evaluation of an aspect includes several sub-dimensions. Staged model provides a single summarized rating for all the related aspects in the assessment.

In continuous model, it is possible to address more value to a dimension over the others. This allows the organizations to focus on the areas where they see more risk, if required. With the standardization of the evaluation, it is possible to compare different departments and companies dimension-wise. Whereas in staged model, it is a chain of improvements starting from the basics and progressing through levels of maturity. Each level serves as the baseline of the upcoming level. This model also allows the comparison between units through standardized methods, but this is more of an overall comparison rather than dimension-wise. Thanks to this standardized evaluation method, it is possible to see where you are headed among others[16].

2.4.2. Presented SPL Maturity Models in Literature

2.4.2.1. SEI SPL Practice Framework

The four main purposes of the framework are[17]:

- Identification of the concepts and core activities prior to the evolution of the SPL
- Identification of the practice areas that needs to be specialized
- Identification of activities for the practice areas
- Providing guidance for the utilization the SPL approach

Framework defines three essential activities; core asset development, product development and management for the development of the SPLs. Framework describes 29 practice areas under 3 categories. These categories are organization management, technical management, software engineering. Each practice area is addressed in detail including aspects like example practices and practice risks. Their relations with the essential activities are also given.

This framework provides a strong guidance for the utilization and improvement of the product lines. However this framework does not provide an assessment methodology or give maturity values to the practices. Although this framework does not give an explicit idea about the maturity, it is important to understand it to be able to come up with a maturity model, since it provides a quite detailed and structured guidance about the topic.

2.4.2.2. PuLSE

Product Line Software Engineering, PuLSE, is an overall SPL framework[18]. The framework suggests a bottom up methodology and is like a user manual, from the initialization and deployment to evolution and evaluation. The framework is based on three pillars; deployment phases, technical components and support components. Figure 5 visualizes the overview of PuLSE framework.

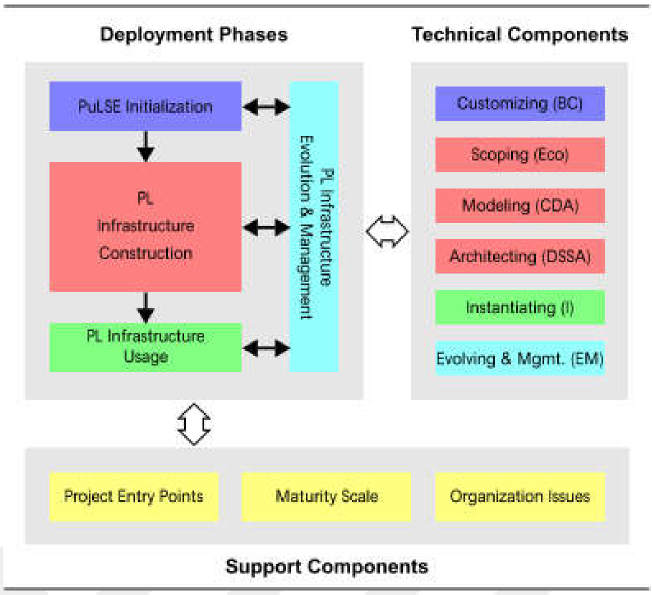

Figure 5 PuLSE Overview[18]

The deployment phases element explains four activities; initialization, infrastructure construction, infrastructure usage, evolution and management. The steps to follow for the utilization of SPLs are explained in this element. These are the phases gone through during the deployment of the product line suggested by the methodology. Technical components construct the technical know-how that is necessary to realize the phases of the deployment. These components are; customizing, scoping, modeling, architecting, instantiating and evolving and management. The support components provide guideline throughout the lifecycle of the SPL. These components are project entry points, maturity scale and organization issues. Each of these components are investigated and explained in detail by the framework.

The framework provides a PuLSE specific maturity evaluation method. This method points out where the framework employer stands though the utilization. Four different maturity levels are defined by the framework. These are initial, full, controlled and optimizing, each implying how further is gone during the utilization of the PuLSE. The assessment provides a good insight while utilizing PuLSE but does not claim to be a generic SPL maturity assessment methodology.

2.4.2.3. Maturity and Evolution in SPLs

Bosch presents five maturity levels in this presented model[19]. The initial level is independent products which actually represents the absence of any kind of maturity or a SPL approach. The second level is the standardized infrastructure. This level also does not actually represent the presence of a SPL but evaluates the signs of it. This is described as the first step while evolving into a mature product line. Commonly in this level, there are some typical components on top of an operating system. Third party software can also be present which is integrated to the framework via some glue code. Next comes the platform. In the platform level, all the functionality is captured on top of the standardized infrastructure. Reuse is exploited. The fourth level is SPL. When the commonality is shared between sufficient components, that functionality belongs to the shared artefacts and commonly used. Shared artefacts concept comes into the play with this level. The final level is called configurable product base. There is a single code base that can be configured for different functionalities. Two other concepts are also introduced in this paper. These are not maturity levels but rather different approaches. When the number of products increases derived from the product line, product population approach is in use. When the number functionality and features are increased, program of product lines approach is in use. The article presents how the product line evolves and which maturity levels it passes through, but it does not actually present a methodology or a guideline about how to assess it.

2.4.2.4. SPLEMM for Small and Medium Sized Organizations

This model[20] seeks to present a methodology to assess the maturity of a SPL in small and medium sized organizations. The focus is on the companies that have less than 250 employees and less turnover than 50 million euros. Overview, adoption and process improvement of SPLs are addressed in the article. Two types of elements are used in the model, process areas and example actions[20]. Example actions serve for a similar purpose as the informative part of CMMI. A process area can be called as a maturity area, process area or a sub process area depending on the position of it in the hierarchical structure. There are four top tier process areas to mature. These are business, domain engineering, application engineering and collaboration maturity areas. Four levels of maturity are defined for all the maturity areas. These levels are called non performed, adaption, sustainable and improving. Maturity areas are divided into process areas. Process areas are divided into sub process areas and sub process areas have different example actions implying for different maturity levels. Although it is not necessarily required for a sub process area to have example actions representing all four possible maturity levels. This also means that, all sub processes do not necessarily have at least four example actions. This article presents a valuable framework for small and medium sized organizations.

2.4.2.5. Family Evaluation Framework

This article[21] presents the BAPO model as a framework to evaluate SPLs. FEF presents five maturity levels for each aspect of BAPO. The BAPO model is explained further in detail in the upcoming section.

2.5. BAPO Aspects of SPL Maturity Evaluation

These four aspects of SPL maturity, BAPO, are all defined in 5 levels of maturity. First level implying the lowest level of maturity and fifth implying the highest. Each four aspect have their own evaluation dimensions to assess the maturity of the aspect itself.

2.5.1. Business

2.5.1.1. Dimensions and Practices of the Business Aspect

As illustrated in figure 6, business aspect is divided into three business dimensions and eight business practices to be evaluated. Each business practice is an assessment area of a certain dimension. The three business dimensions are Marketing Strategy, Portfolio Management and Business Planning. Business practices categorized under the Marketing Strategy are Market Orientation, Relationships Management and Order of Entry to the Market. Financial Management and Asset Management are the practices of the Portfolio Management. And lastly, Business Planning dimension consists of Strategic Planning, Business Vision and Innovation practices[8].

Business Dimensions	Business Practices
Marketing Strategy	Market Orientation
	Relationships Management
	Order of Entry to the Market
Portfolio Management	Financial Management
	Asset Management
Business Planning	Strategic Planning
	Business Vision
	Innovation

Figure 6 Business Maturity Dimensions and Practices

The first business dimension is the market orientation. Narver and Stanley(1990) studied with 113 strategic business units to confirm the effect of market orientation on business profitability and positively associated the market orientation to the business profitability. They also claim that the key features of market orientation are customer orientation, competitor orientation and interfunctional coordination[22]. According to Birk et al market orientation is about the targeted customers, either a specific market segment or individual customer projects[23]. The relationships management deals with the relations between all the stakeholders and the company itself. This includes suppliers, vendors, customers, subcontractors and all other applicable stakeholders. Customer Relationship Management (CRM) and customer satisfaction is in the core of all this since it is the customers who pays you for the end product, in the end. Clark and Fujimoto assert that the key of successful of business managers is addressing the top priority to the satisfaction of customers as a whole and creating attractive product concepts and realizing them through design and production[24]. Markides and Sosa splits the business strategies into three categories according to the order of the entrance of the market and sustaining current position[25]

- the business models that pioneers utilizes to benefit the first-mover advantages (FMAs) due to early entry
- the business models that later entrants adopt to compete with the pioneers
- the business models that the pioneers use to respond to these challenges

The second business dimension, portfolio management, is of high importance in business. Brigham and Houston defines the financial management aims to help managers maximize their firms' values[26]. In a study by Cooper et al, they present the eight key reasons cited by the senior management participated in their study, most of which are linked to the financial or asset management. These are [27]

- to maximize return and productivity and achieve financial goals
- to sustain their position and increase market share
- to properly and efficiently allocate resources
- to forge the link between the organization's strategy and projects
- to resource valuable projects rather than trying to do everything with that limited resource
- to balance the weight of long and short term, high and low risk projects
- to identify the priorities better
- to achieve better project selection and eliminating not worthy ones

The third dimension is the business planning. In a study by Delmar and Shane, where they study over the benefits of business planning on 223 ventures in Sweden, they stress the three benefits of business planning. Business planning [28]

- "Facilitates faster and better decision making by identifying missing information without first requiring the commitment of resources
- Provides tools for managing the supply and demand of resources in a manner that avoids time-consuming bottlenecks
- Identifies action steps to achieve broader goals in a timely manner"

Companies face many challenges throughout their lifetimes. To overcome these challenges, there is always a need for some sort of planning, a strategy. However, what is meant here is not simply giving it a thought on a strategic manner. Strategic planning is the way to face challenges. The strategic planning is a guideline throughout the why and how to do what processes. Strategic planning helps managers to successfully overcome challenges by means of planning and synthesis[29]. Vision states the goal that the company is trying to accomplish. Business vision is about the direction of the company, hence the direction of the scope of the business which results in the scope of the product line. This roadmap needs to be clearly defined to identify the expectations from the product line and therefore, release of the products[30]. The product line is not a static structure, it evolves within its lifecycle. This evolution leads to innovations in products. Without innovations, a company or a product line cannot gain a competitive advantage or even stay in the competition. To solidify the statement, one can look at the example of Apple. Product and business model innovation put Apple at the center of a market approximately 30 times larger than its original market[31].

2.5.1.2. Maturity Levels of the Business Aspect

The five levels of business maturity of product lines are defined as[8], [32].

- **Reactive:** The business reacts upon the situation on an ad hoc basis instead of actively influencing the SPL engineering
- **Awareness:** The business is aware of the SPL engineering but not aware of how to benefit from and utilize it.
- **Extrapolate:** SPL engineering is utilized to achieve business goals.
- **Proactive:** SPL engineering and the related business goals are planned and managed to maximize the.
- **Strategic:** SPL is a strategic asset to reach the business goals.

The first level is the reactive level. In this level, SPL engineering is not yet a part of business planning. Strategic planning and business vision does not recognize SPL and there is no innovation based on SPL engineering. The effects of SPL engineering is not visible on the portfolio management and marketing strategy. There is no recognition of it in practices of all three business dimensions. This level can be considered as the immature stage of business practice in SPL engineering.

In the second level, also called as the awareness level, there is an awareness, as it is in the name of the maturity level, about the benefits of the SPL engineering for the business values. However, there is also lack of awareness in the utilization of the right instruments for it. The business dimensions are trying to be tied with it, but not accomplished, yet. There might be some feedback mechanisms established to be able to enable the utilization of SPLs and the creation of a framework. This level can be considered as the planning stage where the initial steps are taken.

In the third, extrapolate level, the instruments and the framework for the utilization of SPLs are available. Now, it is becoming a planned activity. Market information is used to direct the scope of it. SPL engineering is a part of the strategic planning and used as a parameter in marketing strategy. The effects are getting visible in the asset and financial management. It is now a part of the business vision. This level is where the companies start to enjoy the benefits of SPL engineering.

In the fourth, proactive level, the SPL is a parameter in the financial model. Marketing feedback is used to identify the opportunities, improve the quality and satisfy the needs of the customers with the utilization of SPLs. SPL is highly dynamic and the company is more responsive. The software development lifecycle is shorter and this is visible in time to market. There is an alignment with the business vision and strategic planning. There is management support for innovations.

The strategic level is the fifth and the highest business maturity level of the SPLs. In this level, quantitative feedback mechanisms take over the qualitative feedback mechanisms. The benefits of product line engineering are visible in the market competition, which in turn, comes with a financial strength. The company has its place as a frontier or an early adopter. The role of SPL engineering in business planning is significant.

2.5.2. Architecture

Software Architecture studies has an older history compared to others. It has always been in the scope of software developers and researchers. The software architecture can be defined as the structure or the structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them[33]. In this study, the focus is on the utilization of architecture on the SPL development rather than single product development. Perry and Wolf describes the expected benefits of software architecture as a major discipline in four aspects[34]

- the framework for satisfying requirements
- the underlying technical and managerial bases for design, cost estimation and process management
- the effective basis for reuse
- the framework for dependency and consistency analysis

2.5.2.1. Dimensions and Practices of the Architecture Aspect

As seen in figure 7, architecture aspect is divided into three dimensions and six practices to be evaluated. Each practice is an assessment area of a certain dimension. The three dimensions are Architecture Design, Product Line Management and Documentation. Practices categorized under the Architecture Design are Domain Engineering, Requirements Management and Modeling and Architecture Analysis and Evaluation. Commonality Management and Variability Management are the practices of the Product Line Management. And lastly, Architecture Artifact Management is the activity for Documentation[5].

Architecture Dimensions	Architecture Practices
Architecture Design	Domain Engineering
	Requirements Management and Modeling
	Architecture Analysis and Evaluation
Product Line Management	Commonality Management
	Variability Management
Documentation	Architecture Artifact Management

Figure 7 Architecture Maturity Dimensions and Practices

A lot of researches has been done on architecture design, the first dimension, by a lot of researchers. As a result, there are a lot of approaches, methodologies and frameworks suggested about the subject. There are also a lot of focus groups on architecture design. The designs differ in many ways depending on the goal and the domain of the studies. In this study, the focus is on SPL architecture. In other words, the investigation is on the common software architecture of a product family rather than the architecture of a single product. The goal here is to achieve a generic framework in the scope of product line engineering independent of the domain. As mentioned most designs differ in a lot ways, but they are also quite similar in many ways since the underlying problem is the same as defined by Hofmeister et al(2007); maintaining intellectual control over the design of software. For all, there is the involvement of multiple parties as stakeholders. The software is maintained for long of time. The software is mostly developed by a large group of distributed development teams which also extend with time. Multiple goals and concerns are addressed where conflicts are highly seen[35].

Domain engineering has quite a big impact on other SPL engineering activities. The reasoning is explained by Metzger and Pohl[36]. Domain engineering specifies the scope of commonalities and variations, thus application engineering activities are directly defined by the domain engineering activities. Higher commonality results in lower costs and efforts in product development in SPLs. The SPL needs to be stable and yet flexible at the same time. The requirements usually comes from multiple independent sources in companies that utilize SPLs. These requirements commonly does not fully match with each other but it is the duty of the domain engineering to exploit the commonalities among those requirements. Since there are multiple parties that has different requirements for the products, requirement management is a challenging activity in SPL engineering. Lots of companies utilize tools for the requirements engineering activities. When there is lack of management and modelling in the requirements, a lot of redundancies may pop up. These redundancies result in additional redundant efforts on the product line, inconsistencies between definitions and they are hard to identify[37]. To achieve a high quality in the resulting products, a high quality in software needs to be achieved. This is measured thorough architecture analysis and evaluation. The evaluation is to analyze the architecture to identify risks and verify the quality requirements are satisfied[38].

Product line management is the second architecture dimension. Commonalities and variabilities activities are actually in the scope of domain engineering. However, due to the importance of the activities, they are investigated separately. They are key activities of the SPL engineering. Domain engineering is responsible for the scope of the commonalities and variability points, whereas the application engineering is responsible for the development of the products by exploiting these commonalities and defining the variabilities according to the requirements[36]. There has been various researches and presented models about the topic from various perspectives. Some presented feature oriented models[39]–[41], some took an architecture centric approach[42], [43], some had an configuration based approach[44]–[46] some investigated UML utilization to model variability[47], [48] some focused on the identification of commonality and variability[49], [50] and some scoped the tool support for variability management[45], [51], [52]. More detailed reviews like on the topic are available in the literature[53].

Without proper documentation, it is not quite possible to establish or maintain a successful SPL. Pohl et. al. state that for the successful product realization from a SPL, up to date documentation is a prerequisite[54]. Without documentation, reasoning of the variabilities become ambiguous and inconsistencies become difficult to identify[55]. With the growing code bases, it is not possible to transfer the knowledge without proper documentation. All this said, it is not enough by itself to document what is done, but it also needs to be traceable. With the growing code base, the documentation database also grows and it is highly difficult without a traceable documentation structure. Requirements, designs, tests and performance analysis need to be bound to each other to be able to trace the work. This traceability facilitates the knowledge of where the requirements are implemented[56]. This enables maintaining the knowledge of how it is tested, whether the quality is assured or not and prevents redundant work.

2.5.2.2. Maturity Levels of Architecture Aspect

The five levels of architecture maturity of product lines are defined as[8], [32];

- **Independent Product Development:** Software product are developed individually at this level, there is no software family engineering but ad hoc reuse.
- **Standardized Infrastructure:** The focus at this level is on the standardization of the SPL infrastructure and architecture.
- **Software Platform:** The SPL is being used as the basis for product development.

- **Software Product Family:** The product variations are determined by the family architecture.
- **Configurable Product Base:** There are defined set of rules and the automatic selections of the assets are available to develop products by utilizing configuration management.

In the first level, namely independent product development, the software products are developed individually. There are no introduced SPL engineering activities. There is no specific domain or application engineering activities, development activities occur on an ad hoc basis. The requirements are maintained at individual product level, there is no requirements management as in the scope of product families. The SPL is not formed yet, therefore, architectural activities, commonalities and variabilities do not exist yet. Documentation is also at the individual product level, if any. The reuse of any activities or assets are not planned but done on an ad hoc basis, if any.

The second level, standardized infrasturcure, is the early architectural stage of the SPL engineering. As it is in the name, the first step is to have a standardized infrastructure for the product development. The benefits are seen and the aim is to utilize it efficiently. The organizations at this level try to achieve some level of technical know-how about how to utilize SPL activities. Domain engineering activities are supported. There are some efforts on requirements management. The evaluation and analysis techniques are not defined yet. There are deficiencies in the management of commonalities and variabilities. There are efforts on artifact management but there are defined sets of documentation rules yet.

The third level is the software platform. At this level a software platform that is the basis for the developed products exist. There is sufficient knowledge to utilize SPL and benefit from it. The infrastructure of the SPL is developed. The roles and activities of domain engineering is defined. The requirements are being managed at the product line level. The SPL architecture is developed. There interfaces, components, classes and objects are defined and managed in the scope of product line management. The architecture analysis and evaluation methods are utilized. Commonalities and variability points are defined. The documentation of the architecture is available and traceable.

In the fourt level, software product family, the SPL architecture is fully established. The scope of the product line is well defined. The related documentation is available. The variability is managed and controlled among products. The requirements and the scope of the product line is well aligned. For the evaluation of the architecture, qualitative methods and metrics are used and the feedback from them are used for the improvements. The commonalities are being exploited as much as possible and much of the focus is on the product specific development. Thanks to efforts on requirements management, product specific and common requirements and development is well scoped. At this level, beyond the documentation and traceability, a configuration management system is also utilized.

The configurable product base is the fifth and the highest level of maturity to be achieved for the SPL architecture. The product line is the backbone for the business. There is a collaboration among the organization to manage, maintain and improve the SPL and its processes. The domain engineering activities and organizational plans are effected by each other. Requirements are regularly reviewed. The architectural analysis and evaluation is regular and continuous activity. The organization benefits from past experiences and SPL knowledge. The architecture is less error prone due to this effects. The commonalities are maximized and all development activities are evaluated in the scope of product line engineering. There are established change management methods to maintain the architecture.

2.5.3. Process

Third BAPO aspect, process is not actually in the scope of this study. The reason is that the topic is already well studied and there are already widely accepted frameworks available. Most well-known international standards are known as CMMI of SEI and SPICE of ISO. The general overview of the aspect will be addressed but it will not be thoroughly investigated here.

Software process maturity has a longer history than the other aspects in software. It has been a key research area and the CMM, Capability Maturity Model 1.1, was first introduced in 1991 by the SEI-CMU, Software Engineering Institute - Carnegie Mellon University. In the first version, CMM V1.1, it is stated that the CMM presents sets of recommended practices in a number of key process areas that have been shown to enhance software-development and maintenance capability **(Capability Maturity Model, Version 1.1)**. CMM is presented as a guideline to improve the practices utilized in process areas. In the year 2000, CMM is evolved into CMMI, Capability Maturity Model Integration. CMMI originates from the study that aims to assess the quality and the capabilities of the software contractors of U.S. Department of Defense. Later on, it got widely accepted by the community and became an international standard for assessing the process maturity of the organizations. In 2012 CMMI Institute was founded by the CMU and the studies are continued by the CMMI Institute. CMMI is adopted by more than then thousand organizations located in more than a hundred countries. The appraisals and certifications are organized and authorized by the CMMI Institute. The Standard CMMI Appraisal Method for Process Improvement assessments, SCAMPI in short, are conducted by the assessors certified by the CMMI Institute.

Besides CMMI, there is another widely used model for the process maturity, SPICE, Software Process Improvement and Capability Determination. This is the model presented by the common efforts of ISO and IEC, International Organization for

Standardization and International Electrotechnical Commission. The first maturity family was presented under the ISO/IEC 15504-X family in 1998. Later in 2015, the standards were revised by ISO/IEC 33xxx family and these standards are being used since then. There are still some ISO/IEC 15504 standards that are being actively used but all these are also being revised into ISO/IEC 33xxx family. Unlike CMMI Institute, ISO develop and set the standards but do not take part in the certification, this is done by external certification bodies.

Unlike the other BAPO aspects there are widely accepted and utilized common frameworks for the process capability maturity evaluation. Therefore, this aspect is not in the scope of the study but rather a guideline to assess the maturity of the other aspects. CMMI and ISO/IEC 330xx family process maturity models are based on extensive technical knowledge and experience. This is what is being tried to achieve in the other aspects as well, through various investigations and case studies.

2.5.4. Organization

It would be fair to say the other three aspects of SPL engineering were studied more by the researchers. As also stated by Jan Bosch, during their studies most of the researches assumed the organizational structure is by the book, several domain engineering units on top of a domain engineering unit[57]. This, of course does not mean that the assumed model was insufficient or ineffective, but rather this was not the case for all the companies

Researchers studied how to utilize SPLs, developed processes and models, investigated the profit of it. However, organizational structures and their relation with the success of SPLs did not take interest as much as others. This might not be that surprising, though. In a simple manner of thinking, first, one needs to understand what to do, which is the technical part, the architectural aspect. Then it is necessary to be better at how you do it, may be some set rules or some sort of guidelines, which corresponds to the processes. All of this is for profit, of course. How you align the technique with the business highly matters to maximize the benefits and profit. This relates to the business aspect. Then comes how you adapt your organization to all these. In smaller organization there is a possibility of organizational aspect being more disregarded. To illustrate, it would not be surprising if a company with ten developers do not care about their organizational structure while utilizing SPL approach. Of course some practices are still in the scope like organizational commitment and learning about the SPL approach. Some like the organizational communication and conflict management are also easy to handle since there are too few developers. However this cannot be the case for larger companies. Today, there are companies with thousands of employees scattered all around the world. Developers might even be in quite different time zones. At this point, how you manage and structure it as an organization is highly crucial.

2.5.4.1. Dimensions and Practices of the Organization Aspect

Organization aspect of the BAPO model is divided into two dimensions and seven business practices as visualized in figure 8. Each organizational practice is an assessment area of a certain organization dimension. The organization dimensions are organizational behavior and organizational management. Organizational culture, organizational commitment and organizational learning are practices investigated under the organizational behavior dimension. The practices that are investigated under the organizational management are organizational structure, change management, conflict management and organizational communication[9].

Organizational Dimension	Organizational Practices
Organizational Behaviour	Organizational culture
	Organizational commitment
	Organizational learning
Organizational management	Organizational structure
	Change management
	Conflict management
	Organizational communication

Figure 8 Organizational Maturity Dimensions and Practices

Organizational behavior is the first organizational dimension and consists of three practices. Organizational culture is quite an important factor for the success of SPL utilization. One reason for that is, in some conditions, it can easily lead to the failure of the SPL approach. It actually is so crucial that, it may even end the journey before it starts. Bosch gives a real life example for that. In a company whose name remains to be anonymous, although the teams were quite experienced with the object oriented frameworks and extensive reuse, the management faced with a resistance when they tried to implement product line approach within their software. The resistance was actually so hard that the attempt was unsuccessful and it got cancelled. The reason was actually the organizational culture. The teams had to sacrifice their lead architects for the task and it might have led to delays in the projects. This resulted in the resistance and the initiative got cancelled[58].

Such migrations also require employee commitment to be realized. Koziolek et al. explains that it is quite difficult to implement SPLs in dynamic business areas, since it does not promise immediate benefits. Therefore, such implementations require long term commitments to be successful[59]. Without proper commitment for the implementation of the SPL, it is highly probable that the attempt fill fail. In the successful cases, the transition time is dependent on the success of the organizational learning. How knowledgeable is the organization about the SPLs in various aspects? The better they are, the higher are the chances for a successful implementation.

Levitt and March make remarks on the three sides of organizational learning; learning from direct experience, learning from the experience of others, developing conceptual frameworks and paradigms based on these experiences[60]. In the case of SPLs, as well, the aim is to use this process of learning and experience for building and continuously improving owned product line framework. Nevertheless, the term organizational here, should not be taken wrongly. It is not actually a completely separate concept from individual learning. Organizational learning is achieved through the individual learning as well. A distinction point is, organizational learning is independent from the learning of a specific individual, while it is not independent from the learning of all individual employees[61].

The other organizational dimension, organizational management, has four practices. Organizational structure can be defined as the formal allocation of the roles, responsibilities and the managerial mechanisms to control the work activities and integrate them including also the cross formal organization boundaries[62]. For a successful implementation of the SPLs, organizational structure should be aligned with the purpose. To illustrate in the simplest way, it is not possible to maintain and construct a successful SPL for a complex environment if the development teams act independently from each other. To resolve this, the roles and responsibilities of the organizational units should be clearly identified and followed. When the roles and responsibilities are defined, that leads to a controlled independence among various units. Organizational structure is a key factor to achieve organizational effectiveness. Customers, marketers, asset and application development teams and managers are the key players of the organizational structure[63].

During the transition period towards a SPL approach, urgency and vision of organizational change management is very important. Also after the institutionalization, it is required to improve it continuously[64]. Therefore, organizational change management plays an important role. During the process, it is also necessary to tackle the legacy tacit knowledge with related organizational change processes[65]. Without being dealt with properly, these all may slow the process. While managing the organizational change, communication with the employees is a key factor to success. Every employee needs to have an understanding of the strategy, benefits and goals so that they can be achieved[66].

Conflicts are always part of the job and they are not always bad. Although mostly, the lower the conflicts happen, the better the situation is. The reason they are not all bad are, they can help the employees to have gain an understanding on the subject and improve it. Surely it is important that the conflicts are managed so that the possible negative impacts on efficiency are reduced. While teams with high conflict management skill can work productively, teams that do not have the skills may get demoralized and ineffective[67]. Conflicts are not needed to be reduced, eliminated or suppressed necessarily but they need to be managed instead, to achieve higher efficiency[68].

Organizational communication is a valued aspect by most organizations. Like this method, agile methodologies also highly values communication. The reason is, when done properly, it is capable of speeding up the processes and allow new ideas to hatch. In a sense, communication is the method for transferring, processing and storing the system or environmental data, where the organizations are the information processing systems[69].

2.5.4.2. Maturity Levels of Organization Aspect

The five levels of architecture maturity of product lines are defined as[9];

- **Preliminary:** There is no evidence of organizational acknowledgement of SPLs. As in the name, this is the preliminary stage for the organizational maturity.
- **Consistent:** At this level 2 of the maturity, there is an effort for the migration to SPLs from single product development by introducing organizational changes.
- **Streamlined:** The organization is committed to the SPL concept.
- **Matured:** Organizational strategies are aligned with SPL approach strategy.
- **Institutionalized:** SPL is a strategic core asset for the organization and a vital tool to achieve business objectives.

In the first preliminary level, there is no evidence that SPL approach is a value or objective for the organization. There is no sign of effort for the migration from single product development to SPLs as an organization. There are no guidelines, processes or procedures related to the approach. Due all these reasons, there is no organizational commitment, culture, structure or management activities related to the product line approach.

The second level is the consistent level. At this level, there are efforts or a vision to for the establishment of the SPL approach. There is an awareness related to the potential benefits and new ideas for change are supported and valued. There is managerial support to the employees for the utilization of the approach. There is a commitment for the migration from single product development to SPL approach. The knowledge is tried to be shared among employees and there may be assistance in the means of workshops or trainings. Structural changes are also evident for the utilization of the approach. The change in the organization is merely structured but rather ad hoc.

In the third level, the streamlined level, the organization has established a SPL approach. There are set of rules, guidelines and procedures defined regarding the approach. There is also employee commitment for the vision and the goals of the organization. The required knowledge is acquired by the employees and this knowledge is in use. The experience and the acquired knowledge enables the improvements. SPL concepts are a part of the organizational culture. The organization structure supports the SPLs and application and domain engineering activities have their places in the organizational structure. The information is shared among employees and learning process also continues for further improvements.

The fourth level is called the matured level. The organizational is able to align is strategies with the established SPL framework. There is commitment to the maintenance and improvement of the product lines. The benefits are exploited. Defined roles and procedures keeps the conflict at a healthy rate. SPL is aligned with the organizational goals and objectives and it is enabler for their realization. It is a recognized asset rather than an objective at this level. Employees are confident about the benefits.

In the institutionalized, fifth, maturity level, SPL is a key core asset to realize the business objectives. It plays a major role in the company vision. Management values the benefits of the approach and therefore employee feedback for better optimization of the processes. There is constant support the optimization of the process and improvements. There is commitment to continuous learning and innovations. Conflicts at this level are rather solved easily since there is mutual trust among employees and between employees and management. Conflicts are more likely to end up in improvement ideas rather than problems in efficiency.

2.6. CMMI

CMMI is actually a generic software development process capability maturity model rather than an overall SPL maturity evaluation framework. However, this model has a big impact on maturity evaluation studies, hence it is quite useful to mention it. Capability Maturity Model Integration (CMMI) was developed by Software Engineering Institute (SEI) of Carnegie Mellon University (CMU). Since the introduction of the model, SEI maintains the framework and delegates for the appraisal of it. The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) assessments are conducted by the SEI certified assessors.

CMMI-v1.1 consists of 22 process areas grouped under 4 categories. These four categories are; Process Management, Project Management, Engineering and Support. Each process area consists of specific and generic goals to be satisfied. These are the required components of the CMMI. There are also defined specific and generic practices organized under those goals, respectively. These are the expected components of the CMMI. CMMI also defines sub practices, generic practice elaborations and typical work conducts beneath the generic and specific practices. These are the informative components of the CMMI. Nevertheless, the focus is actually on the "what" part rather than the "how" part. The informative components are just presented as a guideline and are not part of the evaluation method.

CMMI provides six different maturity levels for the continuous representation, the first actually implying the absence of any kind of maturity. The six levels are; Incomplete, Performed, Managed, Defined, Quantitatively Managed and Optimizing, in ascending order. In the staged representation, there are five maturity levels. These are Initial, Managed, Defined, Quantitatively Managed and Optimizing in the ascending order.

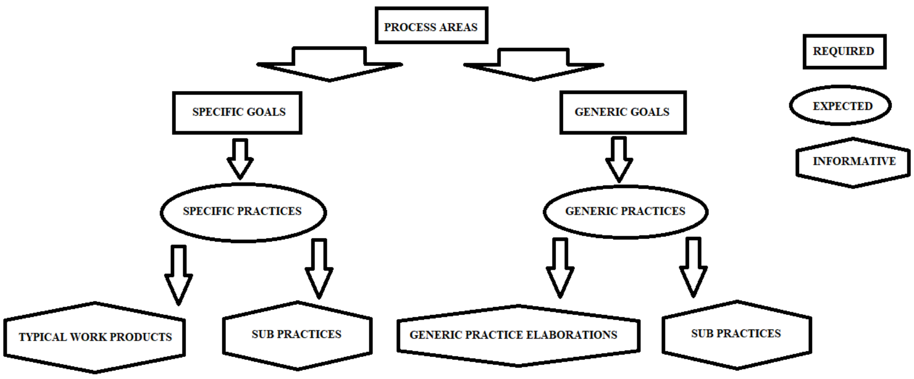

Figure 9 CMMI Structure

Although it is a widely accepted maturity model, this model is not sufficient for the evaluation of the maturity of a SPL as a whole. CMMI is a generic model for the evaluation of software process maturity. Nevertheless, the presented model will use CMMI as is, for the evaluation of the process aspect of SPLs.

2.7. Software Process Improvement and Capability Determination Model

Software Process Improvement and Capability dEtermination Model (SPICE) also known as ISO/IEC TR 15504 standard establishes a structured assessment framework for the software development processes and related business management functions. It enables process assessment and determination of process capabilities. While providing a process capability rating which denotes the current state of the process, it also provides a basis for process improvement. ISO/IEC TR 15504 Part-5 is an informative part which gives a thorough explanation of the structure of the process assessment model. It includes purpose, outcomes, base practices and work products for the software development processes. The purpose explains the high level objective of the process. Outcomes are the achievements of successful implementation of the processes and base practices are the enablers of the successful implementation.

SPICE has two dimensions, one is the capability dimension and the other one is the process dimension. Capability dimension has 6 capability levels ranging from "Incomplete" level to "Optimizing" for each process. Capability of each process is independently measured as shown in the figure below. Each level is characterized by Process Attributes (PAs). Meanwhile, the process dimension consists of processes defined in conformance with ISO/IEC 12207- "Systems and software engineering -- Software life cycle processes".

The PAs are applicable to all processes. Each PA of each capability level describes a specific characteristic of process capability. The achievement levels of the PAs are based on a defined rating scale. Process capability levels of the process are identified by the achievement status of defined grouping of PAs.

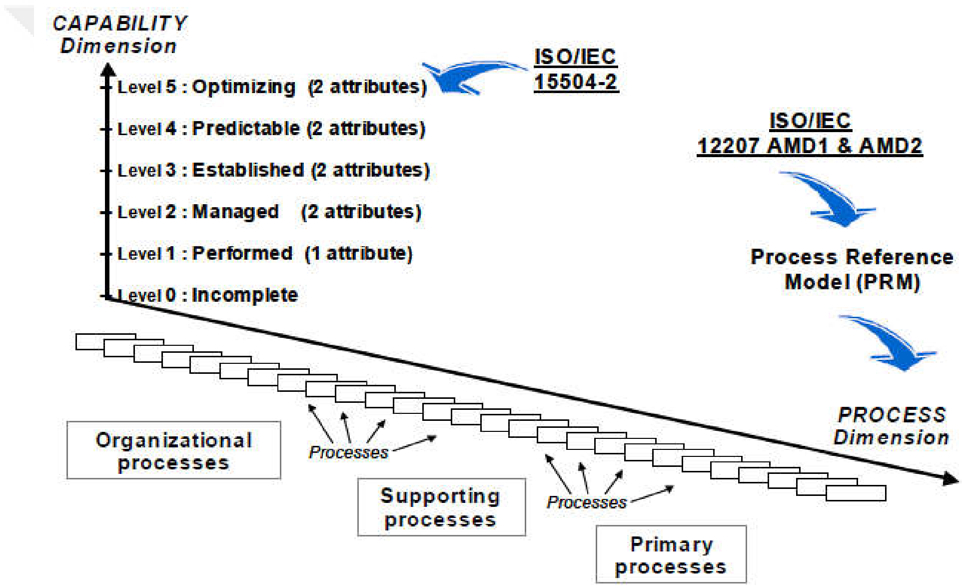

Figure 10 SPICE/ ISO/IEC 15504 Structure

- Level 5 (Optimizing): The process performance is optimized in a continuous fashion to meet current and future business goals.
    - Process Attribute 5.1 Process innovation
    - Process Attribute 5.2 Continuous optimization
- Level 4 (Predictable): The process is performed consistently and operates within defined control limits. Quantitative measures and objectives are established for the quality of work products.
    - Process Attribute 4.1 Process measurement
    - Process Attribute 4.2 Process control
- Level 3 (Established): The process is managed and performed in a defined way.
    - Process Attribute 3.1 Process definition

o Process Attribute 3.2 Process deployment

- Level 2 (Managed): The performance and the work products of the process is managed.
  - o Process Attribute 2.1 Performance management
  - o Process Attribute 2.2 Work product management
- Level 1 (Performed): BPs of the process are performed to achieve the process purpose.
  - o Process Attribute 1.1 Process performance
- Level 0 (Incomplete): The process fails to achieve the purpose and the outcomes.

Process capabilities are determined by the assessments of process attributes. Process attributes have indicators to support their existence and level of achievement. For the process attribute of the first capability level, each process has its own base practices and work products. On the other hand, these are generic for the higher levels of capability and they are identified as generic practices and generic work products.

The achievement of a process attribute is measured in terms of percentage values:

- N.A. (not achieved): 0 – 15%: There is little to no evidence for the process attribute achievement.
- P.A. (partially achieved): 16 – 50%: There is some evidence for the process attribute achievement but it is insufficient to provide the desired results.
- L.A. (largely achieved): 51 – 85%: The PA is achieved and provides the desired results but it is also apparent that there is still room for improvement.
- F.A. (fully achieved): 86 – 100%: All the requirements of the process attribute is fulfilled and there are no shortcomings.

A process is measured to be at a certain capability level if it fully or largely achieves the process attribute(s) of that level and fully achieves all the process attribute(s) of the lower levels.

2.7.1. Reasoning for the Selection of SPICE as a Basis Model

SPICE model is widely accepted and adopted model that is developed by the International Organization for Standardization and International Electrotechnical Commission. It has a well-defined structure that is adoptable to address the needs of various fields.

CHAPTER III

THE INITIAL MODEL OF MODIFIED BAPO

This part explains the initial model used to assess the SPL maturity. The first model is an improved version of the previous studies based on the BAPO framework [5], [8], [9]. After the further investigations of the literature and the case studies, this model is proven to be somewhat insufficient. With the realization of a need for a more holistic and reliable framework, the second model given in Chapter 4 is constructed and presented.

3.1. The Structure of the Assessment Methodology

As already explained in section 2.5 in detail, the structure consists of four aspects and 4 maturity levels for each of the aspects. The result of the assessment is not a single value for all the aspects. Instead, each aspect is rated individually and there are four resulting values in the end. Both studies that are also utilized here actually provides 5 maturity levels. However, 4 levels are utilized here. This is because the first levels do not actually provide any insight about the maturity concept. They represent the lack of any kind of maturity in that area. Thus, there is no use in presenting them as a maturity level. If none of the four levels are achieved, these names can still be used to imply the lack of maturity as level 0. The high leaves structure is presented in table 1 and the details of the dimensions are presented in tables 2,3, 4 and 5.

Table 1 High Level Structure of the Assessment Model

	BUSINESS	ARCHITECTURE	PROCESS	ORGANIZATION
LEVEL 0	Reactive	Independent Product Development	Initial	Unit Oriented
LEVEL 1	Aware	Standardized Infrastructure	Managed	Business Lines Oriented
LEVEL 2	Extrapolate	Software Platform	Defined	Business Group/Division
LEVEL 3	Proactive	Software Product Family	Quantitatively Managed	Inter Division/Companies
LEVEL 4	Strategic	Configurable Product Base	Optimizing	Open Business

Each maturity area is then divided into dimensions. These dimensions indicate the areas that needs to be matured and evolved. Each dimension has different practices and activities to be fulfilled in each maturity level. Process maturity area is out of the scope of the study as mentioned in the earlier chapters. The dimension grouping is the same as in the studies of Ahmed and Capretz [5], [8], [9].

Table 2 Maturity Area Dimensions

BUSINESS	**ARCHITECTURE**	**ORGANIZATION**
Marketing strategy	Architecture Design	Organizational behavior
Portfolio management	Product Line Management	Organizational management
Business planning	Documentation	

Then, the dimensions are divided into practices. Each practice consists of different activities for each maturity level. Practices define the activities to be done to achieve a certain maturity level.

Table 3 Business Practices

Marketing Strategy	**Portfolio Management**	**Business Planning**
Market Orientation	Financial Management	Strategic Planning
Relationships Management	Asset Management	Business Vision
Order of Entry to the Market		Innovation

Table 4 Architecture Practices

Architecture Design	**Product Line Management**	**Documentation**
Domain Engineering	Commonality Management	Documentation
Requirements Management and modeling	Variability Management	
Architecture Analysis and Evaluation		

Almost all dimensions have two or more practices. There is an exception though. In architecture maturity area, documentation dimension does not have multiple practices. Therefore, it serves as both dimensions and practices.

Table 5 Organization Practices

Organizational Behavior	**Organizational Management**
Organizational Culture	Organizational Structure
Organizational Commitment	Change Management

Organizational Learning	Conflict Management
	Organizational Communication

The activities are grouped under practices for each level of the maturity. There are different activities for each maturity level under each practice. Each activity defines the sufficiency for the corresponding maturity level. If a certain percentage of the activities are performed in the same maturity level, the SPL under assessment is said to be sufficient to be on that maturity level on that maturity area. The highest maturity level that is ensured by the SPL with the aforementioned criteria is the maturity level of that SPL.

The overall structure of the model can be thought as a 4 by 4 matrix, rows being maturity levels and columns being maturity areas. Each element of the matrix is a tree structure, top node of the tree being the corresponding maturity area. The nodes spread as maturity areas, dimensions, practices and activities. Activities are the leaves. The depths of the trees are 4.

3.1.1. The Assessment Framework of the Model of Modified BAPO

The maturity scale consists of 4 levels 1 being the lowest and 4 being the highest. It is important to note that even level 1 implies some level of capability in that area. If no capability is found at all after the application of the model, no maturity, and hence no maturity level can be addressed to the assessed structure.

In the previously applied methods, 5 point scale is used but the outcome is actually almost binary. When the activities that are not applicable are removed from the evaluation metrics, the result is either pass or fail. The top two points means pass and the remaining means fail. Therefore, opting for the value 3 or 4 does not bring much value to the study. In this study, the aim is to present the model as simple as possible without losing any valuable data. Since, for each activity, 4 point likert scale is applied. As mentioned, 3 point scale, one of the points being not applicable, is still sufficient for the assessment, but there is a reason for adding the third point. It may not affect the result directly, but it provides insight about how to proceed to achieve that level.

The 4 options are sufficiently performed, performed but insufficient, not performed at all and not applicable. While calculating the capability for a maturity level, both performed but insufficient and not performed at all means fail. The concept here can be explained with the static and kinetic friction analogy. Moving a static object requires more effort than a kinetic object. When the object starts moving, it gets easier to push it. A similar concept is also present here. If an activity is not performed at all, it requires planning and a roadmap to perform it. On the other hand, in the performed but not sufficient case, it generally means keep following and improving the plan to achieve further success. Initiation phase requires more effort and focus. All in all, this distinction provides the implementers some sort of insight about how to proceed.

All this being said, it does not fully mean that number of points in the scale should not be increased. Initially, just to see the results, this study was also applied with a 10 point likert scale. It is observed that this can also be a useful addition as a future work. If the model is extended to provide examples about how to improve in these activities, this can also be a valuable feature. Nevertheless, this is not in the scope of our study for now.

To satisfy a maturity level in a maturity area, same criteria for thresholds are applied as in the base models[5], [8], [9]. To claim a maturity level, 80 percent of the activities of that level should be performed sufficiently. The highest level that is claimed is the actual maturity level of the SPL under assessment. While calculating the required number of sufficiently performed activities, number of activities that are not applicable are subtracted from the total number of activities, if any. To illustrate, if there are 25 activities to achieve some maturity level in a maturity area, 20 of the activities should be sufficiently performed. However, if 5 of these activities are not applicable, then the total number of activities to be performed becomes 20. This will result in the required number of sufficiently performed activities to drop to 16. If the percentage number is not an integer, the number is floored.

In the study of Ahmed and Capretz[5], while assessing the architecture maturity area, variability management has its own pass criteria in addition to the overall score. This rule is not applied in this study. It is seen that all the practices are somehow dependent. Although some practices may seem to have more impact on the maturity of a product line, this distinction is not necessarily required. Practices do not consist of large numbers of activities, therefore, putting such an additional threshold does not seem necessarily required.

For the assessment of the questionnaires, inter rater agreement and reliability calculation methods are often utilized. However, in our case, it is not possible to achieve meaningful results as a standalone study. Moreover, Kendall's coefficient of concordance is not a very suitable pick for this study since we do not apply rankings between options but rather apply a likert scale. Cohen's kappa is also not fitting since we have multiple judges. This is a drawback. Nevertheless, this model is an extension of the previous BAPO model and those values are already calculated for the previously presented models. There is another reason for why this is not evaluated as a critical threat to the study. When the model itself is mature enough, the assessments need to have a SCAMPI like model as in CMMI. Rather than the respondent dependent assessments, proof based assessments should be applied. The results should not depend on the perception of participants on the implementations of activities. These should be evaluated with solid proofs about activities in the end. However, it is beyond the scope of this study and regarded as a future work for now

3.2. Practices and Activities of the Study

In this section, the focus is mostly on the modifications and improvements suggested on the model. There are several omitted and added activities for different maturity levels in different maturity areas.

3.2.1. Modifications of Business Maturity Area

In business maturity area, 4 activities are removed from the base model[8] according to the literature research and participant feedbacks. All the omitted activities belong to the financial management area. The omissions are done on the maturity levels 1, 2 and 3. Omitted ones from the financial management practice within the business maturity area are [8];

- "The organization is able to maintain its debt.
- There is no change in the net profit margin during the last two years.
- The organization is able to reduce its debt.
- The net profits margin increase over a period of time."

First two activities omitted are from maturity level 1, third activity is from level 2 and the last activity is from level 3. These are normally good implications in the sense that the business is mature. However, in the scope of the SPLs, these are too general. It is true that when done right, product lining engineering approach claims to increase efficiency, reduce costs and increase return on investment. However, it is almost always not the case that a company utilizes a single SPL for its all revenue. There might be several product families and therefore several SPLs, which disallows us to evaluate a single product line based on the revenue, debt and profit of the company. Even in a huge amount of cases, the end product is not even a software application. Nevertheless, SPL engineering is an essential part of it. Therefore, there are tremendous amount of other effects on the debt and profit of the company. In the presented evaluation model, it is required to evaluate direct relations to the SPL engineering in the activities. So, in this study these activities are omitted and some of them are replaced with activities which narrows down the scope to the SPLs.

Besides the omitted ones, there are also 5 added activities. Two of these activities are to replace the omitted ones from financial management in maturity levels 1 and 3. Correspondingly, these are;

- There is a neutral or positive trend in the net profit margin related to product line engineering activities during the last two years.
- The net profits margin related to SPL activities increase over a period of time.

Two activities are added for market orientation practice, one in level 2 and one in level 3. Correspondingly, these are;

- A brand name strategy in alignment with the SPL and family is utilized.
- Benefits of SPL approach like high quality, usability and variability for lower costs are promoted as a marketing strategy to gain competitive advantage.

Lastly, there is one more addition in the maturity level 3 activities for the relationships management practice. This is;

- Customer support teams provides service to different products in the family rather than a single product.

Financial management activities are narrowed down to the scope of SPLs. What is desired to assess here is the maturity of the business in the scope of SPL approach. Hence, this was a required modification.

Two activities are added in the market orientation. The first one is for maturity level 2 and the second one is for maturity level 3. As mentioned by Saarlo [20], these are important activities to exploit the benefits of product line approach in the business. There are numerous studies on the brand name strategies and brand equity. Companies try to utilize the benefits of it in every business area. In a study by Cobb-Walgren et al., based on their findings they stated that brand equity increases a customer's intentions of buying the product [70]. In 90s, brand equity was a hot topic on the research list of the Marketing Science Institute. This also had an effect on the increase of studies on brand equity since then. Utilizing product line approach offers the flexibility on products for lower costs. Not all the customers desire the same end product. Each may have their requests based on their needs. This requires modifications on the product. When possible, it is often easier to provide these in software instead of hardware. This also has its costs, but SPL approach provides the companies the benefit of offering these at a lower cost. As in the name of the highest level of architectural maturity, a configurable product base is desired to achieve. This means you can provide variety of functionalities to your customer by configuration management. This is a powerful tool for the business to gain competitive advantage in the market.

One addition to the relationships management is also available in the modified study. When a customer buys a product, it is necessary to provide them with maintenance and support activities. Consider buying a car. The car requires routine maintenance, repair services and spare parts when necessary. You, as the owner, want this process to be as quick as possible and expect a certain quality level. When a spare part is needed, mostly it is already available in the storage of the service provider. Imagine there was no commonality between different models, would it still be possible for the service providers to store such variety of spare parts. The analogy is applicable here as well.

When a problem occurs, customer support teams as in the automotive, help customers. Thanks to these commonalities, it is possible for a support employee to provide service to a variety of products. This enables companies to provide better and faster support at lower costs. The occupancy and efficiency of a customer support employee increases since it offers services to a variety of products. To illustrate, instead of having 10 employees for 10 products, one can provide the same services with 5 employees. In the first case, if an employee needs to attend to different customers for the same fix, one of the customers need to wait. And when there are no issues to fix, the employee becomes idle. However, in the second scenario, idle time is reduced since employees can support various products. Waiting time is reduced for the customers, employee efficiency is increased and costs are reduced all thanks to product line approach and commonalities among products.

3.2.2. Modifications of Architecture Maturity Area

In architecture maturity area, 6 activities are added in total. These are all related to software testing activities. In the architecture maturity evaluation, testing is never mentioned in the base models. Testing is the assurance of quality and functionality. The concepts like test driven development and continuous integration are gaining popularity. There are definite reasons for this as well. It is not possible to overlook the necessity of testing activities in the assessment of the maturity of SPLs. The testing approach should also evolve in alignment with the evolution of the SPL itself. There are a lot of studies done on the software testing area. Mostly, Test Maturity Model Integration, TMMi, is used as a source since it covers the aspects as an overall framework [71]. The extended activities for the first maturity level are;

- The test assets are tried to be reused wherever possible. (Commonality Management)
- The organization is making an effort to acquire technical knowledge and performing test analysis to understand the managing of SPLA test requirements. (Requirements Management And Modeling)

In the second level of the architecture maturity area, 2 additional activities are defined. These are;

- The testing requirements of the SPL are clearly defined, stated, and documented. (Domain Engineering)
- Reuse and commonality is exploited and it is possible to test common parts early in domain testing process. (Commonality Management)

The activity of the third maturity level is;

- The test software architecture is integrated with the product line software architecture. (Architecture Artifact Management)

Finally, the fourth level also consists of 1additional activity which is;

- Testing tasks are automated. (Architecture Analysis and Evaluation)

For the first level of maturity, one activity is added to the commonality management practice and one activity is added to the requirement management and modeling practice. While exploiting commonality in the production code, it also becomes possible to exploit commonality in the test code. This is actually the initiating logic of the whole concept. Same mindset should be applied wherever possible. This will surely result in time efficiency and higher quality. Without utilizing the benefits of SPL engineering in test code, it does not make much sense to claim having a mature SPL. To achieve this, the organization should try to acquire the technical know-how. This activity is mentioned for the generic concept of product line engineering but testing is usually skipped initially. This causes the test framework to lag behind. The earlier this is realized, the sooner the benefits can be realized.

In the second maturity level of the architecture maturity area, an additional activity is presented both in domain engineering and commonality management practices. In TMMI [71] framework, it is expected to have the testing lifecycle and development lifecycle integrated. In each phase of the development, there is a counterpart testing phase. It is necessary in SPLs to involve testing early in development lifecycle. The earlier testing is involved, the sooner defects can be found and fixed. This, of course, results in reduction of costs as well. Although the lifecycle integration is mainly the concern of process area, there are architectural counterparts for it. One of these counterparts is the domain engineering activity added for the second level of maturity. It is also a good idea to involve testing early in the requirements phase. The exploiting of commonality in the test code is a key to achieve this. This will be the grounds where the testing framework and testing architecture evolves. Another benefit with this activity is the prevention of redoes. Development is not the only part benefiting from the SPLs, testing benefits is as well. When common parts are tested early in the domain area development, different applications on top these layers benefit this. This is called as commonality reuse strategy, or CRS by abbreviation.

In the third level of architecture maturity, testing practice is extended with a single activity. This is the second counterpart of the lifecycle integration mentioned earlier in this section. At this level, it is expected to have the test architecture integrated with the product code architecture. This will be the enabler of automation of testing activities. With the binding of the architectures, the structure, process and test way of working will be more visible. This also greatly helps the junior developers. The frameworks have their own standards, rules and guidelines. This reduces reworks and increases the efficiency of review and testing processes.

Testing practice is extended with also a single activity in the highest maturity level. It is expected to automate the test cases as much as possible in this level. Automation of regression tests, unit tests and other test material increases the efficiency. Expected integration of the stable architectures allow the automation of test cases. This still does not mean there are no changes but with the modification of the production code, test code is also modified where necessary. Automated test scenarios fasten the regression tests. Periodically running test cases can control the risks more frequently. With the statistical management of these tests, the SPL becomes more stable and the results get more predictable.

3.2.3. Modifications of Organization Maturity Area

In the organization maturity area, there are two modifications in the second level of maturity. These are the organizational counterparts of the testing practice in the architecture maturity area. One activity in the organizational learning practice is modified to stress the SPL testing trainings and the one activity is added to the organizational structure practice. The activity modified was;

- "Necessary training for SPL engineering is provided to employees"[9].

This is rephrased as

- Necessary training for SPL engineering, including SPL testing, is provided to employees.
- The organization has established testing career paths with well-defined responsibilities.

In the organizational structure in the second level of maturity, alongside with the roles and responsibilities of domain and application engineering areas, software test engineering roles and responsibilities should also be defined. As this can be achieved in the form of software test specialists or software test architects, it is also profitable to have a test group in the organization. These specialists or the testing group should also involve in the management of improvements in the test activities. Without the career paths, there will be impediments in the management of SPL testing. Hence, the reason in the modified activity to stress the SPL testing is not to let it be disregarded. To have a vision in how to evolve SPL testing assets, it is important to have the employees aligned. It is important they share the same vision to achieve the business goals. Testing activities are expected to evolve together with the evolution of the SPL itself. Lagging test practices and assets mean that SPL is not benefited fully. It is true that SPL test activities can also be considered as generic activities. However, in single product software development, we may not expect a software test way of working as structured as it is in the SPLs. Therefore. there is a need to pay special attention and evaluate the testing framework for maturity in SPLs. And when it comes to trainings, there should not be exceptions for this practice as well.

CHAPTER IV

APPLICATION OF THE BAPO AND THE MODIFIED BAPO

In this chapter, empirical testing of the assessment methodology is presented. Initially, the question that is asked is how we can present a model to assess the maturity of a SPL. After the initial investigations of related literature, it is decided to extend the existing models rather than presenting one from scratch. Therefore, the research question of the study has evolved to "How we can extend the present models in the literature to have a better and overall SPL evaluation framework?" During the search for an answer, 4 case studies are conducted. First three studies are conducted in a company that operates in the field of defense industry in Turkey. We will call this company as A. The last case study is conducted in a company that operates in the semiconductor industry in the Netherlands. We will call this company as B.

4.1. Application of BAPO

4.1.1. Pilot Study

Before beginning the data collection phase, a pilot study is conducted in company A to see the applicability of the questionnaires first. Before beginning the data collection phase, it is necessary to see if the questionnaires are clear enough to understand and answer. What a participant expects from a questionnaire and whether these are fulfilled or not is the main question pursued in this phase of the thesis work. The content of the questionnaires and thus the thesis work is not the main objective of the pilot study. The advantage of doing a pilot study is that it gives the opportunity to realize typos, questions that may be misunderstood, questions that may be explained in a simpler and clearer way and so on. It also helps to realize the shortcomings of the questionnaire in general. Some details which seem obvious to the presenter may totally be a new concept for the participants. To realize these and cover these shortcomings before it is too late, pre studies are done. If not, collected data may not give as accurate results as they should.

The pilot study of this thesis work is conducted with seven participants all of which work for the same military defense company that all the thesis work is conducted at. The ages of the participants are ranging from 25 to 40. Two of the participants are from the software testing department and the other two are from the embedded software development department and the remaining three of them are from the software development department. Two of the participants are females and five of them are males. All participants have at least 4, at most 18 years of software experience, so all participants have a general understanding of product lines. Five of the participants have their BSc degrees from Electronical Engineering and two of them from Computer Sciences. Six of the participants have their MS degrees and one of them is pursuing his MS degree.

In the pilot study, the base BAPO framework is applied as is. Each maturity area questionnaire takes about 30 minutes to complete. Each questionnaire is divided into 5 sections within themselves, each section corresponding a maturity level defined in the base model. The case study is conducted online. Participants are also asked for feedbacks for possible improvements both in activities and the general outline of the study. No evaluation is applied for the final results since not all the participants are working on the same SPL.

As responses to the pilot study, general feedbacks about the survey are collected as expected. The introductory information and some typos are corrected as pointed out by the responses. As stated by the participants, questions were clear, understandable and gradable enough. There was one exception, though. There were some feedbacks about the first maturity level of the base study. There were questions with negative sentences and this made the grading confusing at some points. Moreover, the questions were mostly pointing out the absence of any kind maturity. Another drawback of the study was time. It takes twenty to thirty minutes to answer the questions per questionnaire. In total, it takes about an hour to ninety minutes for all questionnaires. However, the time spent is necessary and required for meaningful results. To overcome these drawbacks, the questionnaires are purposefully divided into three maturity areas in the beginning. This way, instead of a single ninety minutes survey, three shorter, independent and twenty to thirty minute surveys are used. Half an hour a day for three days do not seem to be as scary and tiring as ninety minutes at once. Collecting data without boring and distracting the participant increases the accuracy of the results.

Moreover, some feedbacks about the concepts are collected too. Software testing is one of them. It is an important part of the software product development and the product architecture. However, this concept seems to be missing in the questionnaires. Besides, some questions about financial issues seem to be not applicable to this study's concept. Questions about debt maintenance and profitability cannot directly indicate the success or failure of a SPL in this study. The reason for this is that the company does not rely all his finances on these SPLs. Nevertheless, these feedbacks are not directed to the first exploratory study. The reason why is to collect data for the raw surveys in the exploratory study and check whether similar feedbacks will also be collected during the exploratory study. This way, it is aimed to strengthen the feedbacks about the subjects without manipulating the participants.

4.1.2. Exploratory Study- BAPO

After the pilot study, first exploratory study is conducted in company A. This study is the first phase of the actual data collection phase. The purpose of this phase is to see the result of the assessment and again get feedbacks about the content. Although some feedbacks were discussed after pilot study about the questionnaires contents, no changes are made for this phase. However, these feedbacks are presented to the participants and asked for their feedback on these. To be able to get unbiased feedbacks, first, participants' suggestions are asked and after that comments for the suggestions of the author are requested. It was important not to bias and limit participants' thoughts on the questionnaires. Requesting feedback for the author's suggestions before asking their own might limit them to these topics. Moreover, after the questionnaires phase, informal interviews are made with the participants. Feedbacks from other participants and the previous phase are discussed with each participant. Then these feedbacks and the related literature is investigated in detail. The output of that investigation is presented in the upcoming case study.

This exploratory study is conducted with 8 participants who are also working for the company A. All the participants are working on the same SPL, either as a developer, a manager or a system design engineer. The ages of the participants vary between 27 and 45. Two of the participants are managers in the company. Two of the participants are system engineers with software engineering background and the remaining four are still software engineers. All of the participants have their MS degrees and one of them is pursuing his PhD degree. All of the participants are males in this part of the study. All of the participants has at least five years of experience with the product line being developed. Three of the participants are working on the related SPL since the first steps taken for the evolution of the SPL. The team is one of the frontiers in the company for managing a SPL. The SPL being discussed in this exploratory study is a command control software application.

Similar to the pilot study, each maturity area questionnaire takes about 30 minutes to complete. Each one is divided into five maturity levels, first questions representing the lowest level and the latest ones representing the highest level of maturity. This case study is also conducted online.

The feedbacks about the content of the model was quite similar to the ones in the pilot study. Some comments are made about the negative sentences in the first level of maturity. The absence of testing framework in the architecture maturity area is also mentioned by two of the participants. During the informal face to face discussions with the participants, they are asked about the possible modifications on the study which are based on literature and the previously collected feedbacks. Their comments are also collected and analyzed. Based on the collected data, possible modifications are agreed by all of the participants to be applied for the next study.

The maturity level of the SPL under investigation is assessed individually for each maturity area. The results of individual questionnaires are presented in Table 6. The business and organization maturity areas of the investigated SPL is found to be at level 5 by most of the participants. These two areas are leading the architecture area for improvement for the corresponding SPL. The architectural maturity is found out to be at least level 3, but it is close to achieve level 4 with some improvements. Although it is not in the scope of this study and it is not evaluated by this study, the department has a CMMI Level 3 certificate. Therefore, it seems fair to say that the SPL under investigation has all the BAPO aspects covered. The results of the exploratory study are in table 6.

Table 6 Exploratory Study Assessment Results

	Level 1	Level 2	Level 3	Level 4	Level 5
Business	0	0	0	2	6
Architecture	0	0	7	1	0
Process			CMMI certified		
Organization	0	0	0	1	7

4.2. Application of the Initial Model of Modified BAPO

After the first exploratory study, first explanatory study is also conducted in company A. This study is the first one that we actually collected data for the extended model. The purpose of this phase is to see the result of the assessment with the extended model and see if any further adjustments are required. Although all the feedbacks were discussed after the first two studies, actual data collection can always lead to new information and improvements. The same process is followed for the data collection as in the previous applications.

This study is conducted with 9 participants who are also working for the company A. All the participants are working as developers on the same SPL. The ages of the participants vary between 25 and 37. Two of the participants are pursuing their master degrees, other two participants are pursuing their PhD degrees and the rest of the participants have their MS degrees. 4 of the participants are females and 5 of the participants are males in this part of the study. All of the participants has at least three years of experience with the product line being developed. Three of the participants are working on the related SPL since the first steps taken for the evolution of the SPL. The SPL under investigation is an embedded machine control SPL.

In terms of extensions, 4 level maturity is applied in this study. First maturity levels of each maturity area are omitted. Moreover, all the omissions mentioned in chapter 3 are realized. For the business maturity area, extensions in the financial management practice stated in section 3.3.1 are applied. In the architecture maturity area, there are two extensions. The commonality management activity added to the second maturity level and the architecture analysis and evaluation activity added to the fourth maturity level mentioned in section 3.3.2 are presented to the participants in this case study. In the organizational maturity area, all the modifications mentioned in the section 3.3.3 are applied.

The maturity level of this SPL is also evaluated individually for each maturity area. The details of the participants' individual outcomes are presented in table 7. The organization maturity area of the investigated SPL is found to be at level 5 by most of the participants. The business maturity is at mostly found as level 4 and architecture maturity is evaluated to be at level 3 by all the participants except one. After the evaluation, the team also analyzed themselves for improvement. They have some room for improvement in the analysis and evaluation practice and have set some goals to improve. This department also has CMMI Level 3 certificate. The results of the second and third study are consistent with each other. It is good to have consistency between these two case studies since both SPLs are from the same company and they have many common projects. Especially the business and organization areas are quite overlapping, therefore this enables the validation of the study. The product line in the second study has a bit of longer history than this one. This early start can also explain why the previous SPL is slightly more mature than the one in this case study.

Table 7 First Explanatory Study Assessment Results

	Level 1	Level 2	Level 3	Level 4	Level 5
Business	0	0	1	7	1
Architecture	0	1	8	0	0
Process			CMMI certified		
Organization	0	0	0	2	7

With the application of modified BAPO model and assessment reviews, it is seen that this model is not sufficient to provide a successful capability maturity framework. The model is incomplete and hard to use. Surveys take a lot of time and some questions are hard to answer due to negative sentences. It also fails to provide a roadmap and guideline for improvement. Its capability levels do not match with the standards like CMMI and SPICE, either. For capability and maturity assessments, it is very important to eliminate the human factor. The model is based on survey responses and do not trace hard evidence. Respondents' perception of the processes, practices and questions may affect the results. For a successful implementation, it is crucial to look for the actual evidence that will prove the capability and provide the guideline for further improvements. Due to all these reasons, we concluded that the BAPO model is insufficient to fulfill the needs of the concept. Therefore, we realized the need for a whole new approach and came up with SPL-CMM.

4.3. Validity Threats

For a successful case study implementation, 4 threats to validity are tried to be eliminated.

4.3.1. Construct Validity

For the threats against construct validity, there are 3 measures taken for 3 specific threats. The first threat to the study is the mono method bias. To overcome the mono-method bias, multiple surveys are done for the study including a pilot, an exploratory and a validation study. Before each case study, participants of the previous studies are asked about the modifications presented. The respondents of those studies have a familiarity with the questions and concepts being studied. Moreover, based on their feedbacks there are applied and unapplied modifications on the survey. With this extra step, participants are able to check the results of their feedbacks, if any, and comment on the extensions before they are deployed. Beyond these studies, the results are discussed with the experts of the field and the advisor of this thesis work.

Against the mono operation threat, the study itself is actually a guard. The antecedent surveys are applied to two different companies in North America. With this study, we have the chance to apply the surveys in 2 three different SPLs in a different company from the previous studies. This helps to prove the results of both the base study and the modified study presented currently.

To overcome the evaluation apprehension, complete anonymity of the participants is guaranteed. None of the participants know the answers of each other to the questionnaires. Their names are not matched with their feedbacks so that they remain anonymous. Within this ensured mutual trust, unbiased results are aimed to be collected throughout the case studies.

4.3.2. Internal Validity

Surveys are divided into three, namely, architectural evaluation, organizational evaluation and business evaluation. Each of the surveys take about 30 minutes. If done sequentially, maturation might be an issue in such long durations. The reason behind this division is to alleviate the maturation risk, so that each can be done separately. The surveys are also separated into five parts, each of which can be completed separately. This gives an opportunity to give breaks between parts. The participants are also given a week to complete the surveys due to their tight schedules and availabilities. To let them complete the tasks without a rush, which may result in inaccurate responses, this longer period is chosen. Even though these precautions, a few participants needed some extra days to complete the surveys.

The same methodology is applied in all phases of the study to alleviate the risks of instrumentation. All these surveys are also submitted online instead of face to face meetings to avoid biasing the participants. Altering the methodology may result in different effects on the participants which may eventually effect the outcome of the study.

4.3.3. External Validity

As stated by Yin, external validity is about how generalizable a study's findings are [72]. The surveys are, in a way, repeated as many times as possible with different work groups that work on different product lines to conclude that the results can be generalized at a certain level. A pilot study, an exploratory studies and an explanatory study are performed. Through the studies, some improvements to the models are tried to be introduced. Therefore, to get the opinions of the previous participants, they are also asked about their notions regarding this changes. That way, along with the antecedent models used, the modifications are also presented to all participants of different phases.

The antecedent models are also applied in a different geography, work environment and culture. This study, in a way, alleviates the threats to external validity of the previous study while also proving its own generalizability. This study is designed based on previous researches findings and standards. By depending on the surveys in the literature, all subjective elements are carefully eliminated. The main effort of this study is actually highly related with this validity test. The focus is on the applicability, in other words, generalizability of the antecedent studies and required modification introductions on them.

4.3.4. Reliability

The reliability is the fourth validity test that needs to be fulfilled. It is highly crucial to prove, as it is in its name, that the results of the study are reliable. Without reliable conclusions, a study cannot be told to be worthy and meaningful. If the case study cannot be repeated, its results may not be trusted. The same also goes for case studies which does not give the same results when repeated.

Due to confidentiality policies of the company that the case study is conducted, third parties may not be able to repeat the case study. This is an unsolvable issue since there are strict confidentiality regulations. The company is not always willing to share such information since they sometimes consider such information as company secret. Nevertheless, the processes, results of questionnaires and steps of the case study are explained in detail to maximize the reliability.

CHAPTER V

SOFTWARE PRODUCT LINE CAPABILITY MATURITY MODEL

In this chapter, the developed SPL-CMM is presented. The aim is to achieve a model that will suit the needs of the businesses. Available literature has been reviewed to understand the current status of the concept. The purpose is to present a generic assessment methodology that can also be used for benchmarking. Another important issue that was a concern in this study is to make the model easy to apply and easy to understand. The first step is to identify the structure of the model. The structure defines the maturity areas, their practices to be fulfilled and maturity scales.

To achieve a good framework, both the sources in the literature and the field is applied. All the information and feedback collected through them are analyzed. The feedback from the field experts are supported with the literature. The findings of the literature are discussed with the field experts. Through this double check mechanisms and investigations, the assessed BAPO model is apparently insufficient. Therefore, a new SPL process capability model is worked on and tested. The initial model of modified BAPO is an improved version of the previous studies based on the BAPO framework[5], [8], [9]. After the further investigations of the literature and the case studies, this initial framework is also proven to be insufficient. As a major drawback, the model does not follow causality and the link between consecutive capability levels. For instance, to achieve level 3, the organization should probably fail in level 2. This is because of the negative expectancies of the lower level questions. These negative questions expect the absence of an aspect to claim lower levels, which actually implies incapability and should not be used to assess capability.

Although there are limited number of studies related to measuring SPL process capability/maturity are in the literature, none of existing studies is developed based on a well-known PCMM. The purpose of this study is to develop the SPL-CMM based on SPICE by developing SPL Process Reference Model (SPL-PRM) including process definitions for SPL engineering domain. The aim of SPL-CMM is to provide the base for improving the SPL engineering processes. It pursues a structured and standardized approach by assessing relevant processes in order to perform improvement initiatives in a consistent, repeatable manner, assessed by adequate metrics with guidance on what to do for improvement in SPL.

The significance of the study is that the SPL-CMM including SPL specific process definitions and a measurement framework enables objective rating. It provides to determine the current process capability level and to achieve a road-map for process improvement.

SPL-CMM consists of two dimensions, process and capability. Process dimension includes SPL-PRM covering four main process areas of Business, Architecture, Technical and Organization, and 16 critical SPL processes defined in these four process areas. The process definitions include measurable objectives of a process: Process Outcomes, Base Practices (BPs), and Work Products which are constructed based on SPICE. The capability dimension includes Capability levels and Process Attributes (PAs) adapted from SPICE. The PAs representing measurable characteristics necessary to improve the process are applicable to all processes.

5.1. Process Dimension of SPL-CMM

SPL specific processes are identified and defined. For the level 1 capability assessments of the processes, base practices are described. ISO/IEC TR 15504-2 structure and aspects are utilized for the definitions of the processes. There are 4 main aspects of process definition. These aspects are purpose, outcomes, base/generic practices and work products.

- The purpose defines the desired goal of the process;
- The outcomes are the achieved results when the process is performed;
- The base practices are the expected actions and enablers that will result in the outcomes;
- The work products are the inputs and outputs of the processes such as plans, documents and reports.

The process definitions are developed for 16 SPL specific processes defined under 4 main process areas of business, organization, technical and architecture as shown above. The developed process definitions are given in Appendix.

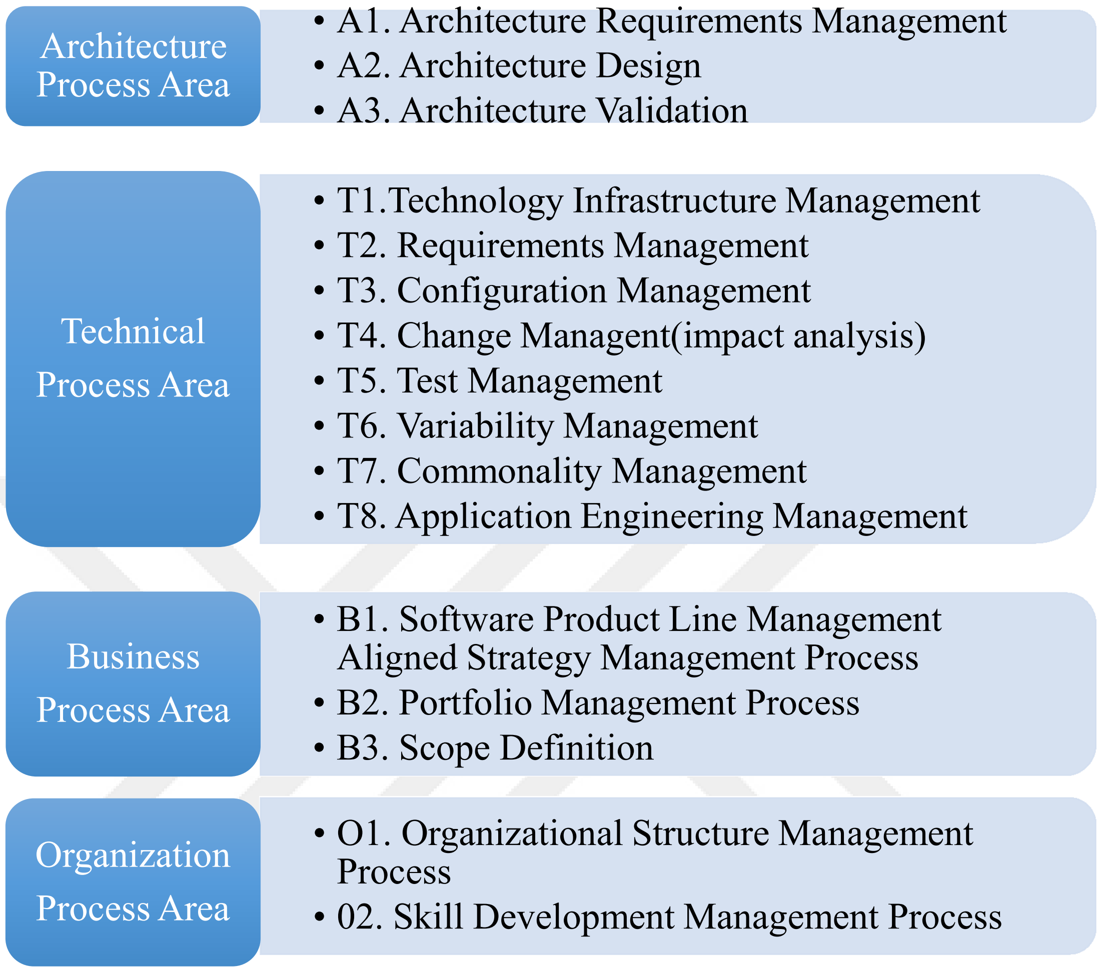

Figure 11 Software Product Line-Process Reference Model (SPL-PRM)

5.2. The Capability Dimension of SPL-CMM

There were 5 capability levels in the [5], [8], [9] models. First of these levels actually maps to level 0 of SPICE. Even the second level of those models imply absence of some processes in some aspects. Moreover, insufficiency in lower levels such as 1 or 2 do not imply any capability or incapability about higher levels. Therefore, after the initial case studies, these assessment models are abandoned and the new holistic approach evolved.

The capability scale of the presented framework consists of six levels as in the ISO/IEC 15504-2[73]. These levels are from level 0 to 5 in the following order; Incomplete, Performed, Managed, Established, Predictable and Optimizing. The capability levels indicate:

- Level 0 – Incomplete: The process is not performed to make it to its purpose or is not even defined.
- Level 1 – Performed: The process is implemented but these implementations are not always consistent. Though it achieves its purpose.
- Level 2 – Managed: The performed process is planned, monitored and adjusted to achieve its purpose.
- Level 3 – Established: The organization is adjusted and defined the way they perform the process.
- Level 4 – Predictable: The process is managed and performed in a way that the variation of its outcome is reduced and limited.
- Level 5 – Optimizing: The quantitative data is used and the process is relentlessly improved to meet its goals.

Each capability level is said to be achieved when its PAs are fulfilled. Level 0 do not have a PA since it actually defines the lack of it. Level 1 has one PA to be achieved while the rest of the levels have 2 PAs each. These process attributes and capability levels are illustrated in Figure 12.

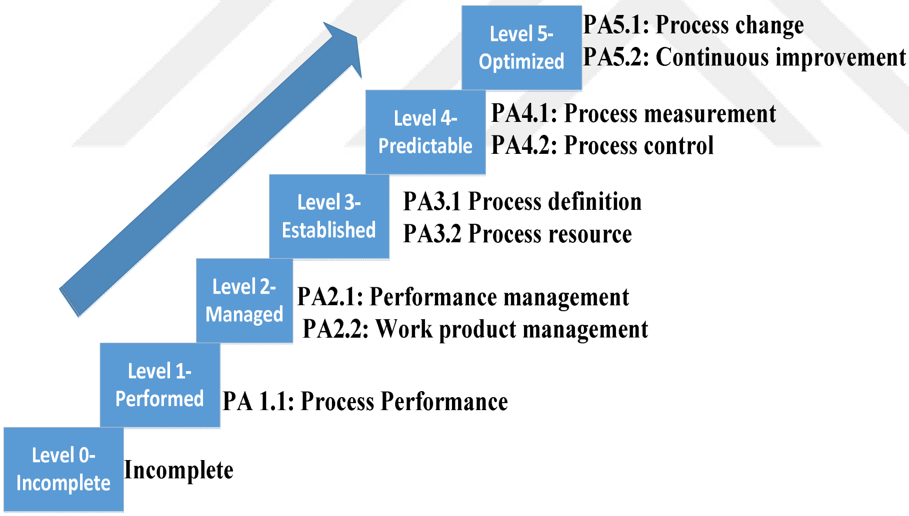

Figure 12 Capability Levels as Adapted from ISO/IEC 15504-2[73]

The measure of capability is based upon a set of process attributes (PA). To claim a certain capability level, the PAs of that level need to be fully achieved (F.A.) or largely achieved (L.A.) and all lower level PAs should be F.A.. However, for level 1, there are no lower level PAs, therefore, process performance attribute is sufficient by itself. If that is not also even L.A., then the capability level is determined as level 0 – incomplete. As shown, the ordinal ratings of achievements in Figure 13 Achievement

Rating Scale, Each PA is measured by an ordinal rating of F.A. (Fully Achieved) (86% to 100%), L.A. (Largely Achieved) (51% to 85%), P.A. (Partially Achieved) (16% to 50%), or N.A. (Not Achieved)) (1% to 15%). These ratings represent the extent of achievement of the process attribute. A process is measured to be at a certain capability level if it fully or largely achieves the process attribute(s) of that level and fully achieves all the process attribute(s) of the lower levels.

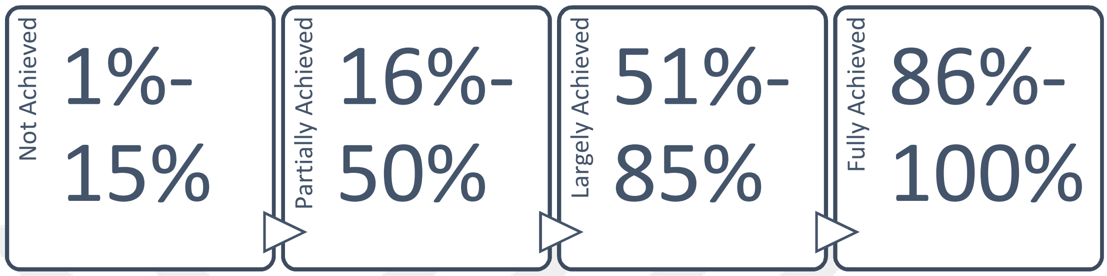

Figure 13 Achievement Rating Scale

According to these ratings and capability levels, Table 8 Capability Levels and Process Attribute Achievement illustrates the process capability ratings against PAs. The achievements of PAs are measured by the success of business practices related to them.

Table 8 Capability Levels and Process Attribute Achievements(Adapted from [74])

Process Attributes	Level 1 (Performed)	Level 2 (Managed)	Level 3 (Established)	Level 4 (Predictable)	Level 5 (Optimizing)
PA 1.1 Process Performance	L.A. or F.A.	F.A.	F.A.	F.A.	F.A.
PA 2.1 Performance Man.	-	L.A. or F.A.	F.A.	F.A.	F.A.
PA 2.2 Work Product Man.	-	L.A. or F.A.	F.A.	F.A.	F.A.
PA 3.1 Process Definition	-	-	L.A. or F.A.	F.A.	F.A.
PA 3.2 Process Resource	-	-	L.A. or F.A.	F.A.	F.A.
PA 4.1 Process Measurement	-	-	-	L.A. or F.A.	F.A.
PA 4.2 Process Control	-	-	-	L.A. or F.A.	F.A.
PA 5.1 Process Change	-	-	-	-	L.A. or F.A.
PA 5.2 Continuous Improvement	-	-	-	-	L.A. or F.A.

First capability level has 1 process attribute which is the process performance attribute. Its achievement means that process purpose is reached. For its assessment, introduced practices and work products are used. For the remaining levels, the generic practices and the generic work products of the SPICE model are utilized.

5.3. SPL-CMM Process Assessment

The assessments are to be conducted according to ISO/IEC TR 15504 part 3. Assessment plan and assessment report documents describe the details of assessment activities. These activities include the planning of the assessment, collection of data, validation of data, rating of the PAs and briefing the participants.

- Assessment Plan Documentation: It defines which activities shall be performed for conducting the assessment. It also involves the schedule and the required resources of the assessment.
- Data Collection: Systematic collection of evidence and data for the process evaluations. The assessment should rely on objective data and evidence for each process attribute of each process.
- Data Validation: The validation of the collected evidence and data in terms of consistency, comprehensiveness, objectivity and sufficiency.
- PA Rating: Assigning a rating based on validated data for each PA.
- Reporting Assessment: Reporting the assessment results including inputs, evidences.

CHAPTER VI

VALIDATION OF THE SPL-CMM

Case study approach has been used as the validation method. Case study approach is *"the most common qualitative method used in information systems"*[75]. It is a highly suitable method to seek answers for the research questions and come up with a solution.

The followed research methodology satisfies the qualitative research requirements. In this study, data is collected in the organizations own site. Data collector is the assessor. Inductive data analysis is conducted and collected data has multiple variations in type.

The case study is conducted according to the template presented by Yin[72]. The collected data has 5 different forms, "*documents, interviews, direct observation, participant-observation and physical artifacts".* This is an important parameter to deal with the construct validity and reliability.

The design type is in the form of a single case study. It is conducted after developing the SPL-CMM.

The objective of the study is to examine how applicable and useful is the presented SPL-CMM model. The model is based on the SPICE model and it aims to determine the capability level of identified processes. It also aims to provide a guideline and a roadmap for process improvement.

The measure used in the research is the capability level of the SPL specific processes.

Case Selection Strategy is to select the organization for the assessment. It is an upside that we know the processes of the organization. In this manner, it is easier to identify the weaknesses and strengths.

Case Study Research Questions: The research questions of the case study are;

RQ1: How suitable it is to use the SPL-CMM with the purpose of identifying the current SPL specific process capability level and how well it provides roadmap for the process improvement?

RQ2: What are the strengths and weaknesses of the SPL-CMM?

Field Procedure, Data Collection, and Limitations: ISO/IEC TR 15504-2 is used as a guide for defining the SPL specific processes and ISO/IEC TR 15504- 3 is used while conducting the assessment. The extent of process capability is determined by an identified set of process attributes.

After the assessment is conducted and capability levels are determined, results are discussed with the process owners. During these discussions determined capability levels and improvement suggestions are presented. A small questionnaire is also utilized to get some feedback. The questionnaire consists of the following questions;

- Are measuring process capability and obtaining guideline for improvement useful?
- Do you think that applying these suggestions will improve the process performance?
- Is there any information you want to add in the process definitions?
- Are there any missing items in the guideline for improvement list?

6.1. Case Study Implementation

The company is operating in the semiconductor industry and the main product is lithography machines. There are more than 20000 employees worldwide from more than 100 different nationalities. The headquarters are in the Netherlands and around 12000 employees work in the Netherlands. More than 2000 software developers work for the company. The company is on the NASDAQ-100 index and holds more than 80% of the market share with more than 10 billion € yearly revenue.

The assessments are conducted by visiting the organization in 3 consecutive days. During these visits, semi-structured interviews are held with the stakeholders of the SPL-CMM processes. The motive of these interviews are to gather evidence about the process capabilities.

For Level 1 assessment, the developed SPL specific process definitions are used. We checked if the outcomes are achieved and the base practices are performed. For the remaining levels, generic work products and generic practice indicators are checked.

To address our RQs, the SPL-process capability assessment is conducted and a road map for process improvement is identified. After determining the capability levels and guidelines for improvement, results are discussed with the respondents. During these discussions, we questioned if the SPL-CMM model is useful and adequate.

For the **construct validity**, the case study constructs should be well structured and objective. To ensure this, multiple stakeholders in different roles are interviewed.

Beyond the respondents, various sources like the process framework of the organization, role definitions and other documents are investigated.

In order to ensure **internal validity**, the findings are discussed with the respondents again to eliminate any kind of bias.

6.2. Validity Verifications

For a case study design to be successful, validity threats should be eliminated. There are four tests that should be satisfied in this concept. These tests are construct validity, internal validity, external validity and reliability. The tests are defined by Yin as[72],

- Construct Validity is about the detection and realization of the accurate operational measures for the concepts under investigation.
- Internal Validity is about the verification of causality between the consecutive conditions and statements that are presented.
- External Validity is about the definition of the domain, to which extent the findings of the study can be generalized.
- Reliability is about the repeatability of the study. The study should be able to be repeated and the same results should be achieved to prove the study is reliable.

All these constraints were analyzed during the design phase of the study. Each threat to validity is tried to be eliminated with some measures. Each action to prove the validity of the study is explained in detail in the following subsections.

6.3. SPL-CMM Assessment

For the capability assessment, process owners are interviewed and the process frameworks are investigated. For the assessments, one cluster architect, two sub function architects and one product owner is interviewed. Each of the interviews lasted around an hour. The interviews are conducted in three different days. The notes of the interviews and organization process frameworks and documents are investigated offline in detail. Final assessment scores are identified with this offline investigation.

As an example, T4 is interviewed with all three architects and data is gathered. During the assessments, evidence is asked for the implementation of the base practices. Outputs of the process are searched for evidence. Inputs and outputs are analyzed and verified. In the organization software process framework, the change management process is defined and the process owner is identified. The strategy, way of working, steps and documentation requirements are all defined. Change management framework and infrastructure is investigated. It is governed by a change management portal available to all stakeholders. The portal holds all the related information and stakeholders. The lifecycle of the request is monitored such as the requester, status, owner of the change, investigator, developer, dates and details of the change request. For all the change requests, software impact analysis is conducted and documented. There exists template documents for such analyses, therefore the process is standardized. In this document, all the effected stakeholders, components, interfaces, regression and progression effects, validation and verification objectives are addressed. This impact analysis is reviewed by the stakeholders. These stakeholders may include other developers, product owners, related architects like test architects and functional architects and industrialization engineers. After the analyses, the change is approved or rejected. If approved, the change is implemented, tested and deployed. The change management process is also supported by review, waiver and issue analysis processes of the software process framework. Process audience, inputs, outputs, process owner, controlling processes, process indicators and process risks are all identified if any. Process flow diagrams are also provided for the stakeholders. Change management infrastructure provides quantitative measures such as number of requests and change durations however objectives for these measures are not set. The analysis of these data is also performed on an ad hoc basis. Therefore, standard analysis procedures and techniques are not yet established or communicated through the organization. As a result of this assessment, the process capability is found to be at level 3 since it fails to achieve level 4 attributes.

All the processes are assessed as described for T4, change management process. As a result of the assessments, all of the processes except T5, test management process, are found to be at level 3. Only T5, test management process is conducted at level 4. Test management process largely achieves level 4 while fully achieving all the previous levels of capability. Commonality and variability management processes, T6 and T7, largely achieve level 3 and fully achieves all the previous capability levels. The rest of the processes all fully achieve the first three levels, but only partially achieves level 4. Assessment results of T4 is illustrated below on Table 9. All the detailed assessment results are presented in Technical Report METU/II-TR-2019-100[76].

Table 9 T4 Change Management Process Capability Assessment

GPs	**Evidence**	**Achievement**
PA 1.1. Process Performance Attribute	**The process attribute is fully achieved.**	**F.A.**
T4.BP1: Develop a change management strategy.	Change management process is defined and the process owner is identified. The strategy, way of working, steps and documentation requirements are all defined.	F.A.
T4.BP2: Establish a change management framework to record and track change request.	Change management framework and infrastructure is established. It is governed by a change management portal available to all stakeholders. The lifecycle of the request is monitored such as the requester, status, owner	F.A.

GPs	Evidence	Achievement
	of the change, investigator, developer, dates and details of the change request.	
T4.BP3: Analyze and document the impact of the change.	For all the change requests, software impact analysis is conducted and documented. There exists template documents for such analyses, therefore the process is standardized.	F.A.
T4.BP4: Identify validation and verification needs and regression effects.	In the impact analysis document, effected stakeholders, components, interfaces, regression and progression effects, validation and verification objectives are addressed.	F.A.
T4.BP5: Implement changes when approved.	After the analyses, the change is approved or rejected. If approved, the change is implemented, tested and deployed.	F.A.
PA 2.1 Performance management attribute	**The process attribute is fully achieved.**	**F.A.**
GP 2.1.1 Identify the objectives	Process performance objectives are defined in reference and guide books and software process framework.	F.A.
GP 2.1.2 Plan and monitor the performance	Process performance is monitored by the architects, process owners and other stakeholders.	F.A.
GP 2.1.3 Adjust	Process is monitored and adjusted by the owners and all stakeholders whenever required.	F.A.
GP 2.1.4 Define responsibilities and authorities	Responsibilities and authorities are defined and structured.	F.A.
GP 2.1.5 Identify and make available resources	Resources are identified in yearly and long term plans documents. Staff to manage the process identified. Actions are realized by the staff. Resource management process is defined in project management processes.	F.A.
GP 2.1.6 Manage the interfaces	Reference and guide books are available and processes are defined.	F.A.
PA 2.2 Work product management attribute	**The process attribute is fully achieved.**	**F.A.**
GP 2.2.1 Define the requirements for the work products	Process is established and released.	F.A.
GP 2.2.2 Define the requirements for documentation and control	Documentation process and templates are released, reviewed and waivered.	F.A.

GPs	Evidence	Achievement
GP 2.2.3 Identify, document and control the work products	Processes are implemented by the related personnel.	F.A.
GP 2.2.4 Review and adjust work products	Change management, review and waiver processes are defined in the project management processes.	F.A.
PA 3.1 Process definition attribute	**The process attribute is fully achieved.**	**F.A.**
GP 3.1.1 Define the standard process	Software process framework processes are defined, documented and distributed to stakeholders.	F.A.
GP 3.1.2 Determine the sequence and interaction	Software process framework is structured, their relations, layers, sequence and interactions are defined.	F.A.
GP 3.1.3 Identify the roles and competencies	Roles, competencies and owners are identified and structured.	F.A.
GP 3.1.4 Identify the required infrastructure and work environment	Infrastructure needs and the structure is defined and provided.	F.A.
GP 3.1.5 Determine suitable methods	Processes are defined, reviewed and followed.	F.A.
PA 3.2 Process deployment attribute	**The process attribute is fully achieved.**	**F.A.**
GP 3.2.1 Deploy a defined process	Software process framework processes are defined, documented and distributed to stakeholders.	F.A.
GP 3.2.2 Assign and communicate roles, responsibilities and authorities	Roles and responsibilities of process owners, architects and other stakeholders are defined and assigned.	F.A.
GP 3.2.3 Ensure necessary competencies for performing the defined process.	Necessary competencies and competency owners are identified for processes.	F.A.
GP 3.2.4 Provide resources and information	Required resources and information is available and communication process is established.	F.A.
GP 3.2.5 Provide adequate process infrastructure to support the performance of the defined process	Infrastructure is provided maintained and has a process owner of its own.	F.A.
GP 3.2.6 Collect and analyse data about	Process performance is monitored and reviewed regularly.	F.A.

GPs	Evidence	Achievement
performance of the process		
PA 4.1 Process measurement attribute	**The process attribute is partially achieved.**	**P.A.**
GP 4.1.1 Identify process information needs	Process needs are defined in the process documents.	L.A.
GP 4.1.2 Derive process measurement objectives	Qualitative process objectives are defined in process documents.	P.A.
GP 4.1.3 Establish quantitative objectives	Quantitative objectives are mainly ad hoc or missing	N.A.
GP 4.1.4 Identify product and process measures	Some measures are established but the boundaries are missing	P.A.
GP 4.1.5 Collect product and process measurement results	Results are monitored.	L.A.
GP 4.1.6 Use the results of the defined measurement	Analysis of the measurements are merely ad hoc.	P.A.
PA 4.2 Process control attribute	**The process attribute is partially achieved.**	**P.A.**
GP 4.2.1 Determine analysis and control techniques	Some standard quantitative analysis and control mechanism are available but the process is mostly ad hoc.	P.A.
GP 4.2.2 Define parameters	Qualitative parameters are mostly available but quantitative parameters are ad hoc or missing.	P.A.
GP 4.2.3 Analyze process and product measurement results	Processes are analyzed by process owners and stakeholders but quantitative measurements are ad hoc or missing.	P.A.
GP 4.2.4 Identify and implement corrective actions	Lessons learned are used for corrective actions.	P.A.
GP 4.2.5 Re-establish control limits	Quantitative control limits are missing.	N.A.
PA 5.1 Process innovation attribute	**The process attribute is not analyzed since level 4 attributes are not fully achieved.**	
GP 5.1.1 Define the process improvement objectives	-	
GP 5.1.2 Analyse measurement data	-	

GPs	Evidence	Achievement
GP 5.1.3 Identify improvement opportunities	-	
GP 5.1.4 Derive improvement opportunities	-	
GP 5.1.5 Define an implementation strategy	-	
PA 5.2 Process optimization attribute	**The process attribute is not analyzed since level 4 attributes are not fully achieved.**	
GP 5.2.1 Assess the impact of each proposed change	-	
GP 5.2.2. Manage the implementation of agreed changes	-	
GP 5.2.3 Evaluate the effectiveness of process change	-	

Table 10 presents the process capability assessment results for T4 change management process. Process attributes of level 5 are not assessed since level 4 is not fully achieved.

Table 10 T4 Change Management Process Capability Assessment Summary Result

Capability Level	Process Attribute	Rating Result	Capability level of the Process
Level 1	**PA 1.1. Process Performance Attribute**	F.A.	Level 3
Level 2	**PA 2.1 Performance management attribute**	F.A.	
	PA 2.2 Work product management attribute	F.A.	
Level 3	**PA 3.1 Process definition attribute**	F.A.	
	PA 3.2 Process deployment attribute	F.A.	
Level 4	**PA 4.1 Process measurement attribute**	P.A.	
	PA 4.2 Process control attribute	P.A.	

The current capability level of the process is determined as level 3 as seen in the table above. Capability level improvement of the process means transition to capability level 4 from level 3 by improving PA 4.1 and PA 4.2. from P.A. to L.A. or F.A. It covers improving rating of GPs of Level 4. The road-map for process improvement of T4, Change Management Process including transition to fully satisfying the requirements of Level 4 is as follow:

- Quantitative process measurement objectives should be identified and established.
- Quantitative product measurement objectives should be identified and established.
- Related information and data for the objectives should be identified and collected.
- For the analysis, investigate and choose appropriate analysis methods.
- Define analysis method and communicate it with stakeholders.
- Analyze collected data. Identify corrective actions and apply.
- Adjust objectives and boundaries according to analyses results.

Table 11 illustrates the achievement status of process attributes for each process while Table 12 shows the highest achieved capability levels for each of the processes. The process attributes which are not assesses in table 11 are marked with a dash.

Table 11 Overall Assessment Results

	PA 1.1 Process performance attribute	PA 2.1 Performance management attribute	PA 2.2 Work product management attribute	PA 3.1 Process definition attribute	PA 3.2 Process deployment attribute	PA 4.1 Process measurement attribute	PA 4.2 Process control attribute	PA 5.1 Process innovation attribute	PA 5.2 Process optimization attribute
B1. Software Product Line Management Aligned Strategy Development Process	FA	FA	FA	FA	FA	PA	PA	-	-
B2. Portfolio Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
B3. Scope Definition	FA	FA	FA	FA	FA	PA	PA	-	-
O1. Organizational Structure Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
O2. Skill Development Process	FA	FA	FA	FA	FA	PA	PA	-	-
A1. Architecture Requirements Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
A2. Architecture Design Process	FA	FA	FA	FA	FA	PA	PA	-	-
A3. Architecture Validation Verification Process	FA	FA	FA	FA	FA	PA	PA	-	-
T1. Infrastructure Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
T2. Configuration Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
T3. Requirements Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
T4. Change Management Process	FA	FA	FA	FA	FA	PA	PA	-	-
T5. Test Management Process	FA	FA	FA	FA	FA	LA	LA	-	-
T6. Commonality Management Process	FA	FA	FA	LA	LA	-	-	-	-
T7. Variability Management Process	FA	FA	FA	LA	LA	-	-	-	-
T8. Application Engineering Management Process	FA	FA	FA	FA	FA	PA	PA	-	-

Table 12 Capability Level Ratings of the SPL Processes Performed in the Organization

Processes	Capability Level Rating Result
B1. Software Product Line Management Aligned Strategy Development Process	Level 3
B2. Portfolio Management Process	Level 3
B3. Scope Definition	Level 3
O1. Organizational Structure Management Process	Level 3
O2. Skill Development Process	Level 3
A1. Architecture Requirements Management Process	Level 3
A2. Architecture Design Process	Level 3
A3. Architecture Validation Verification Process	Level 3
T1. Infrastructure Management Process	Level 3
T2. Configuration Management Process	Level 3
T3. Requirements Management Process	Level 3
T4. Change Management Process	Level 3
T5. Test Management Process	Level 4
T6. Commonality Management Process	Level 3
T7. Variability Management Process	Level 3
T8. Application Engineering Management Process	Level 3

The assessment results are discussed with the respondents in a meeting after the capability and guidelines for improvement are identified. During this discussions, a small questionnaire is conducted and the results are presented in Table 13. First two questions of the questionnaire are rated on a scale of 1 to 5 and the other two questions are open ended. The 1 to 5 ratings mean, completely disagree, disagree, neutral, agree and completely agree, correspondingly. The resulting median of the responses is calculated to be 4 for both questions.

Table 13 Interview Results

Question	Survey Type	Response
Q1) Are measuring the process capability and obtaining guideline for improvement useful?	5 points Likert scale	Median: 4

Q2) Do you think that applying these suggestions will improve the process performance?	5 points Likert scale	Median: 4
Q3) is there any information you want to add in process definition of the process? Please write, if any.	Open-end	
Q4) is there any missing item(s) in guideline for improvement list? Please write, if any.	Open-end	

For the usefulness, the assessment is found useful by the respondents and it is stated that it is definitely useful to have an educated way to drive the improvements. It is also agreed that applying these suggestions should help. As additions, two suggestions are received;

- The key product/service characteristics created by the key processes can be improved.
- The process variables that exert important influence can be controlled.

For the missing items, no feedback is received.

Based on these answers, it can be concluded that it is agreed the assessment and the resulting guideline for process capability improvement is useful. It is also agreed that applying these suggestions will help the organization to improve in these processes. Respondents also agree that the process definitions and the guideline are adequate. No additional remarks were done in that manner.

Based on the results of the case study, the answers for our research questions are found out to be;

RQ1: How suitable it is to use the SPL-CMM with the purpose of identifying the current SPL specific process capability level and how well it provides roadmap for the process improvement?

Considering the case study results and the opinions of the respondents on the results, we conclude that the SPL-CMM can be used to identify the process capability level and to provide roadmap for SPL process improvement.

RQ2: What are the strengths and weaknesses of the SPL-CMM?

We interpreted the strengths and weaknesses of SPL-CMM based on the results of the case study in the conclusion section.

CHAPTER VII

CONCLUSION

The SPL is receiving an increasing amount of attention from software industry due to its promising results in cost reduction, quality improvements, and reduced delivery time. Providing a SPL assessment to measure the capability/maturity of a SPL processes is defined as a necessity and there are a limited number of studies in the literature to satisfy this necessity. Most of these studies are developed based on the BAPO. In this thesis, we also conducted case studies to check the applicability, usefulness and completeness of the BAPO and determined that the BAPO is insufficient, incomplete and not useful to measure the capability/maturity of SPL processes. In order to provide a useful, applicable and complete capability/maturity assessment, we have developed SPL-CMM based on SPICE which is a well-known software process improvement model. We customized the SPICE-ISO/IEC TR 15504 by developing SPL-PRM including 16 SPL specific process definitions under 4 main process areas of business, organization, technical and architecture. In order to validate SPL-CMM, we conducted case study in one software organization to assess the SPL capability/maturity of the organization. During the case study, we conducted the assessments via semi-structured interviews with process owners and stakeholders. We evaluated the direct evidences. Then, we discussed the results with respondents to check if they agree with the results. The findings are presented in chapter 5 and according to these findings it is seen that the SPL-CMM approach has been found useful and adequate.

We identified the contributions regarding this study as follows:

- The main contribution of this study is the SPL-CMM, including SPL-PRM which consists of SPL specific process definitions and a measurement framework which provides objective ratings. It is intended to be a holistic approach for SPL process capability/maturity assessment. It presents a basis for SPL processes improvement. It aims to provide a structured and standardized approach for the assessment of SPL processes. It is designed in such a way that it provides the improvement initiatives in a repeatable and consistent manner. In the existing literature, no such approach has been presented yet.
- Defining SPL-CMM based on SPICE-ISO/IEC TR 15504 will improve the applicability of the model. It is also supported by the SPICE community itself to apply the standards to different domains.

- In this study, the applicability of the SPL-CMM is checked with a case study. The case study results indicated that the SPL-CMM succeeds in identifying process capabilities and defects of processes at various levels. SPL-CMM also achieves to provide a road-map for process improvement. The answers of the process owners and stakeholders verify that they are in agreement with the results and the road-map. They also agree that having a road-map to guide the organization on what they need to do for process capability improvement is valuable and helpful. It is approved that these suggestions identified in the guideline will help the organization in improving the process performance.

We identified the limitations regarding this study as follows:

- We could observe the capability levels until level 4 however level 5 could not be observed. Since level 4 is not fully achieved by any of the processes, we could not perform any level 5 assessment. Evaluation of level 5 will definitely be useful to further prove and improve the completeness of the model.

- The case study is performed in one company, however, SPL process assessments in different organizations are needed to improve the reliability and generalizability of the results.

We identified the improvement opportunities regarding SPL-CMM as follows:

- Developing a SPL process capability/maturity self-assessment method covering the complete set of questions that are in alignment with SPL-CMM. Publishing the model and the collecting new assessment data from different organizations and benchmarking the data;

- Developing a tool to support the SPL-CMM assessment activities;

- Performing more case studies in different companies;

- Extending the number of defined processes.

REFERENCES

[1] K. Kim and D. Chhajed, “Commonality in product design : Cost saving , valuation change and cannibalization,” *Eur. J. Oper. Res.*, vol. 125, no. 3, pp. 602–621, 2000.

[2] M. H. Meyer and A. P. Lehnerd, *The power of product platforms*. 1997.

[3] P. Clements, “On the Importance of Product Line Scope,” in *International Workshop on Software Product-Family Engineering{Software Product-Family Engineering }*, 2001, pp. 70–78.

[4] F. Bellotti, R. Berta, A. De Gloria, and V. Zappi, “Exploring Gaming Mechanisms to Enhance Knowledge Acquisition in Virtual Worlds,” in *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, 2008, pp. 77–84.

[5] F. Ahmed and L. F. Capretz, “An Architecture Process Maturity Model of Software Product Line Engineering,” *Innov. Syst. Softw. Eng.*, vol. 7, no. 3, p. 191, 2011.

[6] M. Röglinger, J. Pöppelbuß, and J. Becker, “Maturity models in business process management,” *Bus. Process Manag. J.*, vol. 18, no. 2, pp. 328–346, 2012.

[7] D. Goldenson and D. L. Gibson, “Demonstrating the impact and benefits of CMMI: an update and preliminary results.” 2003.

[8] F. Ahmed and L. F. Capretz, “A business maturity model of software product line engineering,” *Inf. Syst. Front.*, vol. 13, no. 4, pp. 543–560, 2011.

[9] F. Ahmed and L. F. Capretz, “An organizational maturity model of software product line engineering,” *Softw. Qual. J.*, vol. 18, no. 2, pp. 195–225, 2010.

[10] C. Y. Baldwin and K. B. Clark, *Design Rules: Volume 1, The Power of Modularity*, vol. 1. MIT press, 2000.

[11] J. Roger and J. Timothy, “Product family design and platform-based product development : a state-of-the-art review,” *J. Intell. Manuf.*, vol. 18, no. 1, pp. 5–29, 2007.

[12] H. A. ElMaraghy, “Flexible and reconfigurable manufacturing systems paradigms,” *Int. J. Flex. Manuf. Syst.*, vol. 17, no. 4, pp. 261–276, 2005.

[13] P. Clements and L. Northrop, *Software product lines: practices and patterns*. Addison-Wesley, 2002.

[14] J. Bosch, G. Florijn, D. Greefhorst, J. Kuusela, H. Obbink, and K. Pohl, “Variability Issues in Software Product Lines,” in *International Workshop on Software Product-Family Engineering*, 2001, pp. 13–21.

[15] F. Van Der Linden, “Software Product families in Europe, The Esaps and Café Projects,” *IEEE Softw.*, vol. 19, no. 4, pp. 41–49, 2002.

[16] C. P. T. Carnegie Mellon University, *Capability Maturity Model ® Integration (CMMI CMMI SM for Software Engineering (CMMI-SW, V1.1) Improving processes for better products*. 2002.

[17] SEI, “A Framework for Software Product Line Practice version 5.0. technical report,” 2007. [Online]. Available: http://www.sei.cmu.edu/productlines/frame_report/index.html.

[18] J. Bayer *et al.*, “PuLSE: a methodology to develop software product lines,” in *SSR ’99 Proceedings of the 1999 symposium on Software reusability*, 1999, vol. 99, no. May, pp. 122–131.

[19] J. Bosch, “Maturity and evolution in software product lines: Approaches, artefacts and organization,” *Softw. Prod. Lines*, pp. 257–271, 2002.

[20] S. Saarlo, “Software Product Line Engineering Maturity Model for Small and

Medium Sized Organisations," p. 47, 2009.

[21] F. Van Der Linden, "Family Evaluation Framework overview & introduction," *Philips Med. Syst. version 1*, no. October, 2005.

[22] J. C. Narver and S. F. Slater, "The Effect of a Market Orientation on Business Profitability," *J. Mark.*, vol. 54, no. 4, p. 20, 1990.

[23] A. Birk, G. Heller, I. John, K. Schmid, T. Von Der Massen, and K. Muller, "Product line engineering, the state of the practice," *IEEE Softw.*, vol. 20, no. 6, pp. 52–60, 2003.

[24] K. B. Clark and T. Fujimoto, "The power of product integrity," *Harvard business review*, pp. 107–118, 1990.

[25] C. Markides and L. Sosa, "Pioneering and First Mover Advantages : The Importance of Business Models," *Long Range Plann.*, vol. 46, no. 4–5, pp. 325–334, 2013.

[26] E. F. Brigham and J. F. Houston, *Fundamentals of financial management*. Cengage Learning, 2012.

[27] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt, "New Product Portfolio Management for New Product Development : Results of an Industry Practices Study," *r&D Manag.*, vol. 31, no. 4, pp. 361–380, 2001.

[28] F. Delmar and S. Shane, "DOES BUSINESS PLANNING FACILITATE THE DEVELOPMENT OF NEW VENTURES ?," *Strateg. Manag. J.*, vol. 24, no. 12, pp. 1165–1185, 2003.

[29] J. M. Bryson, *Strategic planning for public and nonprofit organizations: A guide to strengthening and sustaining organizational achievement*. John Wiley & Sons, 2018.

[30] J. G. Wijnstra, "Critical Factors for a Successful Platform-Based Product Family Approach," *Proc. Int. Softw. Prod. Line Conf.*, pp. 68–89, 2002.

[31] Z. Lindgardt, M. Reeves, G. Stalk, and M. S. Deimler, "Business model innovation," *When the Game Gets Tough, Change the Game, The Boston Consulting Group, Boston*, no. December, 2009.

[32] F. Van Der Linden, J. Bosch, E. Kamsties, K. Känsälä, and H. Obbink, "Software Product Family Evaluation," *Softw. Prod. Lines*, pp. 110–129, 2004.

[33] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*. Addison-Wesley Professional, 2003.

[34] D. E. Perry and A. L. Wolf, "Foundations for the study of software architecture," *ACM SIGSOFT Softw. Eng. notes 17.4*, pp. 40–52, 1992.

[35] C. Hofmeister, P. Kruchten, R. L. Nord, H. Obbink, A. Ran, and P. America, "A general model of software architecture design derived from five industrial approaches," *J. Syst. Softw.*, vol. 80, no. 1, pp. 106–126, 2007.

[36] A. Metzger and K. Pohl, "Software Product Line Engineering and Variability Management : Achievements and Challenges," in *Proceedings of the on Future of Software Engineering*, 2014, pp. 70–84.

[37] S. Bühne, K. Lauenroth, and K. Pohl, "Modelling Requirements Variability across Product Lines," in *13th IEEE International Conference on Requirements Engineering*, 2005, pp. 41–50.

[38] L. Dobrica and E. Niemelá, "A survey on software architecture analysis methods," *IEEE Transactions on Software Engineering*, vol. 28, no. 7. pp. 638–653, 2002.

[39] e A. S. P. K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, *Feature-Oriented Domain Analysis(FODA) Feasibility Study - A Technical Report*, vol. 98. 1990.

[40] S. Ferber, "Feature Interaction and Dependencies : Modeling Features for Reengineering a Legacy Product Line," *2nd Int. Softw. Prod. Line Conf.*, pp. 235–256, 2002.

[41] H. Ye and H. Liu, “Approach to modelling feature variability and dependencies in software product lines,” *IEEE Proceedings-Software*, vol. 152, no. 3, pp. 101–109, 2005.

[42] A. Van Der Hoek, “Design-time product line architectures for any-time variability,” *Sci. Comput. Program.*, vol. 53, no. 3, pp. 285–304, 2004.

[43] S. Thiel and A. Hein, “Systematic Integration of Variability into Product Line Architecture Design,” in *International Conference on Software Product Lines*, 2002, pp. 130–153.

[44] C. W. Krueger, “Variation Management for Software Production Lines,” in *International Conference on Software Product Lines*, 2002, pp. 37–48.

[45] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch, “Covamof: A framework for modeling variability in software product families,” *Int. Conf. Softw. Prod. Lines*, pp. 197–213, 2004.

[46] T. Asikainen, T. Männistö, and T. Soininen, “Kumbang: A domain ontology for modelling variability in software product families,” *Adv. Eng. Informatics*, vol. 21, no. 1, pp. 23–40, 2007.

[47] D. L. Webber and H. Gomaa, “Modeling variability in software product lines with the variation point model,” *Sci. Comput. Program.*, vol. 53, no. 3, pp. 305–331, 2004.

[48] G. Halmans and K. Pohl, “Communicating the variability of a software-product family to customers,” *Softw. Syst. Model.*, vol. 2, no. 1, pp. 15–36, 2003.

[49] Mikyeong Moon, Keunhyuk Yeom, and Heung Seok Chae, “An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line,” *IEEE Trans. Softw. Eng.*, vol. 31, no. 7, pp. 551–569, 2005.

[50] F. Loesch and E. Ploedereder, “Optimization of variability in software product lines,” *Proc. - 11th Int. Softw. Prod. Line Conf. SPLC 2007*, pp. 151–160, 2007.

[51] H. Samih and R. Bogusch, “MPLM - MaTeLo product line manager,” *Proc. 18th Int. Softw. Prod. Line Conf. Companion Vol. Work. Demonstr. Tools - SPLC ’14*, pp. 138–142, 2014.

[52] H. Eichelberger, S. El-Sharkawy, C. Kröher, and K. Schmid, “EASy-producer: product line development for variant-rich ecosystems,” in *Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools - Volume 2*, 2014, pp. 133–137.

[53] L. Chen, M. Ali Babar, N. Ali, M. A. Babar, and N. Ali, “Variability management in software product lines: a systematic review,” *Proc. 13th Int. Softw. Prod. Line Conf.*, pp. 81–90, 2009.

[54] K. Pohl, G. Böckle, and F. Van Der Linden, *Software Product Line Engineering. Foundations, Principles, and Techniques*, vol. 49, no. 12. 2005.

[55] A. Metzger, K. Pohl, P. Heymans, P. Y. Schobbens, and G. Saval, “Disambiguating the documentation of variability in software product lines: A separation of concerns, formalization and automated analysis,” in *Requirements Engineering Conference, RE 2007 15th IEEE International. IEEE, 2007*, 2007, pp. 365–366.

[56] P. Mäder and A. Egyed, “Do developers benefit from requirements traceability when evolving and maintaining a software system?,” *Empir. Softw. Eng.*, vol. 20, no. 2, pp. 413–441, 2015.

[57] J. Bosch, “Software Product Lines: Organizational Alternatives,” *Icse*, pp. 91–100, 2001.

[58] J. Bosch, “Organizing for software product lines,” *Softw. Archit. Prod. Fam. Int. Work. IW-SAPF-3*, pp. 117–134, 2000.

[59] H. Koziolek *et al.*, “Assessing software product line potential: an exploratory industrial case study,” *Empir. Softw. Eng.*, vol. 21, no. 2, pp. 411–448, 2016.

[60] B. Levitt and J. G. March, “Organizational learning,” *Annu. Rev. Sociol.*, vol.

14, no. 1, pp. 319–338, 1988.

[61] D. H. Kim, "The link between individual and organizational learning," in *The strategic management of intellectual capital*, 1998, pp. 41–62.

[62] J. Child, "ORGANIZATIONAL STRUCTURE, ENVIRONMENT AND PERFORMANCE: THE ROLE OF STRATEGIC CHOICE," *Sociology*, vol. 6, no. 1, pp. 1–22, 1972.

[63] L. Bass, P. Clements, J. Withey, S. Cohen, and L. Northrop, "Product Line Practice Workshop Report," 1997.

[64] J. Klein, B. Price, and D. Weiss, "Industrial-strength software product-line engineering," in *Proceedings of the 25th International Conference on Software Engineering*, 2003, pp. 751–752.

[65] J. Bosch and P. Bosch-Sijtsema, "From integration to composition: On the impact of software product lines, global development and ecosystems," *J. Syst. Softw.*, vol. 83, no. 1, pp. 67–76, 2010.

[66] L. Northrop, S. Thiel, S. Bühne, T. Käkölä, P. Knauber, and G. Chastek, "Exploring the Context of Product Line Adoption," in *International Workshop on Software Product-Family Engineering*, 2003, pp. 19–31.

[67] S. Alper, E. N. Mun, H. Kong, and K. E. M. T. H. S. Law, "Conflict Management, Efficacy, and Performance in Organizational Teams," *Pers. Psychol.*, vol. 53, pp. 625–642, 2000.

[68] A. Rahim and T. V. Bonoma, "Managing Organizational Conflict: A Model for Diagnosis and Intervention," *Psychol. Rep.*, vol. 44, no. 3_suppl, pp. 1323–1344, 1979.

[69] M. E. Pacanowsky and N. O. D. Trujillo, "Organizational communication as cultural performance," *Commun. Monogr.*, vol. 50, no. 2, pp. 126–147, 1983.

[70] C. J. Cobb-Walgren, C. A. Ruble, and N. Donthu, "Brand equity, brand

preference, and purchase intent," *J. Advert.*, vol. 24, no. 3, pp. 25–40, 1995.

[71] TMMi Foundation, "Test Maturity Model integration (TMMi ®)." 2018.

[72] R. K. Yin, "Case Study Research: Design and Methods," *Appl. Soc. Res. Methods*, vol. 5, pp. 33–39, 2003.

[73] ISO/IEC, "Software engineering — Process assessment — Part 2 : Performing an assessment," vol. 2. 2003.

[74] E. GÖKALP, "Gov-PCDM: GOVERNMENT PROCESS CAPABILITY DETERMINATION MODEL," Middle East Technical University, 2016.

[75] J. W. Creswell, "Research design: Qualitative, quantitative, and mixed methods approaches," *Sage Publications*, 2009.

[76] R. B. Çalışkanbaş, "TECHNICAL REPORT METU/II-TR-2019-100," 2019.

APPENDIX

SPL-CMM PROCESS REFERENCE MODEL

Table 14 Software Product Line Management Aligned Strategy Development Process Definition

Process ID	B1
Process Name	Software Product Line Management Aligned Strategy Development Process
Process Purpose	The purpose of strategy management is strategic and organizational management and planning.
Process Outcomes	a. Business vision including software product line management and corresponding organizational strategy and goals are defined.
	b. The business vision and organizational strategy is communicated to all members of the organizations and their commitments to achieve organizational goals are received.
	c. Resources are allocated for software product line development in the organizational strategy document.
	d. The software product line aligned organizational strategy document is established and documented.
	e. The strategy document is shared with all related parties.
	f. The strategy document is regularly reviewed, updated as needed and communicated to all in the organization.
Base Practices	B1.BP1: Monitor the external environment: Monitor the environment economic trends, social and cultural changes, and new Software Product Line management issues. [Outcome: a]
	B1.BP2: Define organizational strategy: Identify the business vision, organizational strategy and goals aligned with Software Product Line management. [Outcome: a,b]
	B1.BP2.1: Develop a business vision including product line aims at retaining current customers and attracting future ones.
	B1.BP2.2: Develop a strategy document covering following:
	*placed the software product lines an important strategic consideration and even a strategic asset.
	*identified key market segments for the software product line.
	*highlighted an evolution in the software product line under changing business conditions.

Table 14 Software Product Line Management Aligned Strategy Development Process Definition *(Continued)*

Base Practices	B1.BP2.3: Develop organizational goals. [Outcome: a,b]
	B1.BP3: Allocate resources for software product line management. [Outcome: c]
	B1.BP4: Document the strategic plan: Develop organizational and business strategic and operational plans.[Outcome: d]
	B1.BP5: Consult to all stakeholders: Contact with stakeholder which can be organizations and institutions outside of the agency, or inside of the agency. [Outcome: b, e]
	B1:BP6: Publish the strategy document and share it all related parties. [Outcome: d, e]
	B1.BP7: Regularly review the strategy document and update it when it is necessary. [Outcome: f]

Input(s)	**Output(s)**
	Strategy Document [Outcome a, d, e, f]
	Communication record [Outcome: b, e]
Organizational structure [Outcome: b, e]	
	Review records [Outcome: f]
	Commitment / agreement [Outcome: b]

Table 15 Portfolio Management Process Definition

Process ID	B2
Process Name	**Portfolio Management Process**
Process Purpose	The purpose of the Portfolio Management Process is to initiate and sustain necessary, sufficient and suitable programs and projects in order to meet the strategic objectives of the organization. This process commits the investment of adequate organization funding and resources, and sanctions the authorities needed to establish selected programs and projects. It performs continued qualification of programs and projects to confirm they justify, or can be redirected to justify, continued investment.
Process Outcomes	a) business opportunities, investments or necessities are qualified, prioritized and selected;
	b) resources and budgets for each project are identified and allocated;
	c) governance mechanism for managing programs and projects is defined;
	d) The programs and projects are controlled;
	e) The decision of sustaining or terminating programs and projects is given.
Base Practices	B2.BP1: Collect business opportunities, investments and necessities as items for current and future programs and projects. [Outcome: a]
	B2.BP2: Conduct feasibility study for alternative programs and projects [Outcome: a]
	B2.BP3: Analyze and prioritize the alternative programs and projects based on organization level criteria. [Outcome: a]
	B2.BP4: Select the best alternative(s) and initiate the related programs and projects [Outcome: a]
	B2.BP5: Estimate resources and budget for the portfolio. Calculate and balance resources and budget for the selected development items. [Outcome: b]
	B2.BP6: Define governance mechanism, accountability and authority for programs and projects. [Outcome: c]
	B2.BP7: control programs and project at pre-defined milestones based on time, budget, quality and requirements satisfaction level. [Outcome: d]
	B2.BP8: Redirect or terminate project if agreement, stakeholder requirements or business benefits are not expected to be achieved. [Outcome: e]

Table 15 Portfolio Management Process Definition *(Continued)*

Input(s)	**Output(s)**
Business goals [Outcome: a, b]	Business goals [Outcome: a, b]
Market analysis report [Outcome: a]	Market analysis report [Outcome: a]
	Feasibility analysis report [Outcome: a]
	Resource and budget estimation report [Outcome: b]
Organizational level criteria [Outcome: a]	
	Commitment / agreement [Outcome: a]
Organizational structure [Outcome: c]	Program and Project Organizational Structure [Outcome: c]
	Communication record [Outcome: c]
Project status report [Outcome: d]	Project status report [Outcome: e]
Project control limits [Outcome: d, e]	
Stakeholder requirements [Outcome: a, d, e]	

Table 16 Scope Definition Process Definition

Process ID	B3
Process Name	Scope Definition
Process Purpose	The purpose of the scope definition process is to draw the boundary between in and out in such a way that the product line is profitable. A scope defines an organization's product line area of expertise—the set of systems that it can build efficiently. Thus, scoping can let an organization take the initiative, by providing a basis for discovering products that may have an untapped market. These new products might be squarely within the defined scope, or they might be outside but "nearby".
Process Outcomes	a) Existing "as-is" situation of the organization is analyzed.
	b) Existing products are examined;
	c) Domain scope is determined;
	d) Product scope is determined;
	e) Assets scope is determined.
Base Practices	B3.BP1: Perform pre-scoping activities. [Outcome: a]
	B3.BP1.1: Analyze market by obtaining information of the market segments in which the domains are inserted for identifying issues.
	B3.BP1.2: Identify organizational context, i.e: structure, maturity
	B3.BP1.3: Identify operational context i.e: business constraints (time-to-market and resources), processes
	B3.BP1.4:Analyze stakeholders
	B3.BP1.5: Analyze business goals (i.e: reduce costs, improve the productivity, improve the time-to-market, improve the quality, increase the company portfolio and gain new markets).
	B3.BP1.5: Analyze the profile of the project team
	B3.BP3: Examine existing products. [Outcome: b]
	B3.BP3.1:Conduct a thorough study of existing products helps identify commonality across a potential product line and identifies the types of differences that are likely to occur
	B3.BP3.2:Identify existing products similar to those that will be part of the product line
	B3.BP3.3:Determine which elements of the products should be considered part of the product line
	B3.BP3.4: Develop an attribute/product matrix which sorts, in order of priority, the important attributes by which products in the product line differ

Table 16 Scope Definition Process Definition *(Continued)*

Base Practices	B3.BP4:Analyze the domain scope based on the criteria of maturity, volatility, market, potential, reuse potential, risks, experience, coupling and existing code potential [Outcome: c]
	B3.BP4.1: Review domains (the stakeholders express their understanding on the domains, enabling them to identify new domains, exclude existent ones or maintain the initial set of domains)
	B3.BP4.2: Analyze sub-domains based on criteria of experience, maturity, risks, volatility, code market, potential reuse potential, coupling
	B3.BP4.3: Identify sub-domains by conducting brainstorm sessions with stakeholders (The stakeholders gather information about their needs and expectations regarding to key sub-domains that are of particular concern to them)
	B3.BP4.4: Prioritize domains and sub-domains
	B3.BP5: Determine product scope based on the determined domain scope [Outcome: c, d]
	B3.BP5.1: Construct user stories
	B3.BP5.2: Identify features
	B3.BP5.3: Identify products
	B3.BP5.4: Construct product map
	B3.BP5.5: Validate product map.
	B3.BP6: Determine assets scope to Establish the reusability of features relevant for the development of the reference architecture [Outcome: c, d, e]
	B3.BP6.1: Create metrics (based on the template of <Purpose><Issue><Object> <Context>, such as Minimize the effort needed for the development of new applications from the viewpoint of software architects in the company.)
	B3.BP6.2: Apply metrics
	B3.BP6.3: Prioritize Product Map to select the features with more potential for the product line
	B3.BP6.4: List and validate the determined appropriate features that be built for reuse.

Input(s)	**Output(s)**
Strategy Document [Outcome: a]	
Organizational structure [Outcome: a]	
Domain Scope Report [Outcome: d]	Domain Scope Report [Outcome: c]
Product Scope Report [Outcome: e]	Product Scope Report [Outcome :d]
	Asset Scope Report [Outcome: e]

Table 17 Organizational Structure Management Process Definition

Process ID	O1	
Process Name	Organizational Structure Management Process	
Process Purpose	The purpose of the Organizational Structure Management Process is to establish pattern of relationships between the parts of an organization, outlining communication as well as control and authority in order to successfully adopt the software product line approach. Organizational Structure Management Process provides incorporate clearly identified roles for individuals and have strong coordination and communication links, are more likely to institutionalize a software product line than organizations that lack the supporting coordination and communication.	
Process Outcomes	a) The organization establishes domain and application engineering units for the software product line engineering process and a joint team from the domain and application engineering units oversees the synchronization of activities in both departments.	
	b) A clear set of guidelines is developed for the management of software product line.	
	c) The roles and responsibilities of individuals and groups are well defined and documented within the organization.	
	d) Committed work is matched to unit resources and qualified individuals are recruited, selected, and transitioned into assignments.	
	e) Software product line core assets are maintained at departmental levels, and the information is shared on a need-to-know basis.	
	f)All conflicts related to software product line are resolved in a professional environment.	
Base Practices	O1.BP1: Establish domain and application engineering units. [Outcome: a]	
	O1.BP1.1: Design the unit structure	
	O1.BP1.2: Define the objectives of the units	
	O1.BP1.3: Determine the roles, effort and skills required for the units,	
	O1.BP1.4: Define appropriate selection criteria are defined for each open position (s).	
	O1.BP1.5:Recruite the candidates for open positions	
	O1.BP1.6: Offer the positions to the candidate whose skills and other qualifications best fit the open position.	
	O1.BP1.7: Provide resources and information to support the performance of the software product line engineering process.	

Table 17 Organizational Structure Management Process Definition *(Continued)*

<table>
<tr><td rowspan="7">Base Practices</td><td colspan="2">O1.BP2: Constitute a joint team from the domain and application engineering units oversees the synchronization of activities in both departments. [Outcome: a]</td></tr>
<tr><td colspan="2">O1.BP3: Define the processes performed in the domain engineering and application engineering units. [Outcome: b]</td></tr>
<tr><td colspan="2">O1.BP3.1: Develop a clear set of guidelines to handle the management of core assets with respect to the software product line.</td></tr>
<tr><td colspan="2">O1.BP3.2: Share the processes and guidelines on a need-to know basis.</td></tr>
<tr><td colspan="2">O1.BP4: Establish and maintain a policy for conducting the Communication and Coordination activities for the software product line engineering processes. [Outcome: c]</td></tr>
<tr><td colspan="2">O1.BP5: Establish the communication mechanism for that the various units within the organization providing feedback to each other on the software product line engineering processes. [Outcome: d]</td></tr>
<tr><td colspan="2">O1.BP6: Track resolved conflicts for the software product line engineering processes. [Outcome: e]</td></tr>
<tr><th colspan="2">Input(s)</th><th>Output(s)</th></tr>
<tr><td colspan="2">Organizational structure [Outcome: a]</td><td></td></tr>
<tr><td colspan="2"></td><td>HR needs analysis [Outcome: a]</td></tr>
<tr><td colspan="2"></td><td>Software Product Line Engineering Processes and Guidelines [Outcome: b]</td></tr>
<tr><td colspan="2"></td><td>Roles, responsibilities and authorities for Software Product Line Engineering Processes [Outcome: c]</td></tr>
</table>

Table 18 Skill Development Process Definition

Process ID	O2
Process Name	Skill Development Process
Process Purpose	The purpose of the Skill Development Management Process is to ensure that all individuals have the skills and knowledge required to perform their assignments related to software product line engineering and to enhance constantly the capability of the workforce to perform its assigned tasks and responsibilities.
Process Outcomes	a) The current and strategic workforce needs and corresponding skill needs for software product line engineering processes in the organization is determined.
	b) A software product line engineering related skill development plan is established and maintained.
	c) Training, mentoring, or other services for developing workforce skills are provided.
	d) Progress in meeting the objectives of the skill development plan is tracked.
	e) It is ensured that individuals develop their knowledge, skills, and process abilities related to software product line engineering.
Base Practices	O2.BP1: Define the organization's current and strategic workforce needs related to software product line engineering. (i.e: A product line architect must be skilled in current and promising technologies, the nuances of the application domains at hand, modern design techniques and tool support, and professional practices such as the use of architectural patterns. The architect must know all of the sources of requirements and constraints on the architecture, including those (such as organizational goals) not traditionally specified in a requirements specification) [Outcome: a]
	O2.BP2: Determine necessities for the software product line engineering related skills. [Outcome: a]
	O2.BP3: Prioritize skill development activities to align with cost, schedule, and other business considerations. [Outcome: a]
	02:BP4: Establish a software product line engineering related skill development plan. [Outcome: b]

Table 18 Skill Development Process Definition *(Continued)*

Base Practices	02:BP5: Review and revise the software product line engineering related skill development plan on a periodic and event-driven basis. [Outcome: b]
	02:BP6: Provide training, mentoring, or other services for skill development. [Outcome: c]
	02.BP6.1: Plan training, mentoring, or other services for developing workforce skills.
	02.BP6.2: Ensure that the plan is implemented.
	02.BP6.3: Individuals who participate in skill development activities receive appropriate orientation in skill Development practices.
	02.BP6.4: Evaluate performance of the training, mentoring, or other services.
	02.BP6.5: Maintain training and development records at the organizational level.
	02.BP6.6: Establish and maintain the graduated training and development activities for the skill development.
	02.BP7: Track progress in meeting the objectives of the skill development plan. [Outcome: c, d]
	02.BP8: Skill-based experience and information is captured and made available [Outcome: c]
	02:BP9: Ensure that the product line training is consistent with and supportive of the overall product line adoption process or any process-improvement efforts [Outcome: e]
Input(s)	**Output(s)**
	Skill Development Plan [Outcome: a, b]
	Training records [Outcome: c]

Table 19 Architecture Requirements Management Process Definition

Process ID	A1
Process Name	Architecture Requirements Management Process
Process Purpose	The purpose of the Architecture Management Process is to identify the domain requirements and the needs of the SPLA
Process Outcomes	a) the needs that the SPLA need to support are identified
	b) interface requirements of SPLA are defined
	c) consistency and traceability are established between domain requirements and the SPLA requirements
	d) SPLA requirements are analyzed for correctness and testability
	e) the requirements are updated as needed
Base Practices	BP1: Describe SPLA requirements to be supported [Outcome: a, b]
	BP2: Analyze SPLA requirements for correctness, completeness, consistency, feasibility and testability. [Outcome: d]
	BP3: Analyze the changes and updates in the system and domain requirements and update SPLA requirements when needed. [Outcome: e]
	BP4: Ensure consistency of domain requirements to SPLA requirements. Establish and maintain traceability. [Outcome: c]
Input(s)	**Output(s)**
Domain Requirements [Outcome:a, b, c]	
	SPLA Domain Requirements [Outcome: a, c, d, e]
	SPLA Interface Requirements [Outcome: b, c, d, e]
	Analysis and review records [Outcome: b, c, d, e]
Change Request [Outcome: e]	
	Change control record [Outcome: e]

Table 20 Architecture Design Process Definition

Process ID	A2
Process Name	Architecture Design Process
Process Purpose	The purpose of the Architecture Management Process is to provide a design for the SPLA
Process Outcomes	a) a software architectural design is designed and developed
	b) internal and external interfaces of each SPLA are defined
	c) consistency and traceability between the SPLA requirements and SPLA design is maintained
	d) the functional and non-functional requirements of the SPL are addressed
Base Practices	BP1: Evaluate alternative architectures according to the defined criteria and define the rationale for choosing the current SPLA. [Outcome: a]
	BP2: Describe SPLA. Transform the domain and application requirements into an architecture that describes the structure and major software items. [Outcome: a, d]
	BP3: Define interfaces for software blocks and components. Identify and document the external and internal interfaces between the software blocks and components. [Outcome: b]
	BP4: Ensure consistency of software product line requirements to software design. Establish and maintain traceability between the requirements and the design[Outcome: c]
Input(s)	**Output(s)**
SPLA Domain Requirements [Outcome: a, c, d]	
SPLA Interface Requirements [Outcome: b, c, d]	
	SPLA Design Records [Outcome: a, b, c, d]
	High level SPLA design [Outcome: a, b]
	Traceability record [Outcome: c, d]

Table 21 Architecture Validation Verification Process Definition

Process ID	A3
Process Name	Architecture Validation Verification Process
Process Purpose	The purpose of the Architecture Management Process is to provide a design for the SPLA
Process Outcomes	a) verification between the requirements and SPLA is performed
	b) risks are identified
Base Practices	BP1: Verify SPLA meets all SPLA requirements. [Outcome: a]
	BP2: Qualitative metrics for the SPLA evaluation are defined. [Outcome: a]
	BP3: Analyze the software product line architecture and document the analysis results. [Outcome: a, b]
Input(s)	**Output(s)**
SPLA Design Records [Outcome: a, b]	
High level SPLA design [Outcome: a, b]	
Traceability record [Outcome: a]	
	SPLA Analysis Records [Outcome: a, b]
	SPLA Quality Metrics Record [Outcome: b]

Table 22 Infrastructure Management Process Definition

<table>
<tr><td>Process ID</td><td colspan="2">T1</td></tr>
<tr><td>Process Name</td><td colspan="2">Infrastructure Management Process</td></tr>
<tr><td>Process Purpose</td><td colspan="2">The purpose of the infrastructure management process is to provide and maintain necessary tools and facilities.</td></tr>
<tr><td rowspan="4">Process Outcomes</td><td colspan="2">a) SPL infrastructure requirements are identified</td></tr>
<tr><td colspan="2">b) SPL infrastructure elements are identified.</td></tr>
<tr><td colspan="2">c) SPL infrastructure elements are developed or acquired.</td></tr>
<tr><td colspan="2">d) SPL infrastructure is maintained and improved for changed requirements.</td></tr>
<tr><td rowspan="3">Base Practices</td><td colspan="2">BP1: Define infrastructure requirements including hardware, software, environment and standards. Identify if the SPL needs to support multiple hardware, OS or else. [Outcome: a, b]</td></tr>
<tr><td colspan="2">BP2: Develop and/or acquire required infrastructure elements. [Outcome: c]</td></tr>
<tr><td colspan="2">BP3: Maintain and improve the infrastructure. [Outcome: d]</td></tr>
<tr><th colspan="2">Input(s)</th><th>Output(s)</th></tr>
<tr><td colspan="2">Infrastructure requirements [Outcome: a, b]</td><td>Infrastructure requirements [Outcome: a]</td></tr>
<tr><td colspan="2">Development environment plan [Outcome: b, c, d]</td><td>Development environment plan [Outcome: c, d]</td></tr>
<tr><td colspan="2">Supplier selection policy [Outcome: c]</td><td></td></tr>
<tr><td colspan="2">Process description [Outcome: c, d]</td><td></td></tr>
<tr><td colspan="2">Delivery record [Outcome: c]</td><td>Delivery record [Outcome: c, d]</td></tr>
<tr><td colspan="2">Tracking system [Outcome: c]</td><td>Tracking system [Outcome: c, d]</td></tr>
<tr><td colspan="2">Acquisition plan [Outcome: c]</td><td></td></tr>
<tr><td colspan="2">Software assets register [Outcome: b, c]</td><td>Software assets register [Outcome: b, c]</td></tr>
</table>

Table 23 Configuration Management Process Definition

Process ID	T2
Process Name	Configuration Management Process
Process Purpose	The purpose of the configuration management process is to ensure the integrity of the products of the SPL.
Process Outcomes	a) a configuration management strategy is established
	b) Components that needs configurable properties are identified.
	c) Configurations of the products derived from the product line are baselined and maintained.
	d) Changes in the configurable items are controlled and recorded.
	e) consistency and completeness is ensured
Base Practices	BP1: Develop a configuration management strategy. [Outcome: a]
	BP2: Identify software components that will be configured. [Outcome: b]
	BP3: Establish product configuration baselines. [Outcome: c]
	BP4: Report, store and maintain baseline histories and enable recoveries of these baselines. [Outcome: c, d,e]
	BP5: Verify configurable components and their structures. [Outcome: e]
	BP6: Control modifications and releases. [Outcome: d]

Input(s)	**Output(s)**
Configuration items [Outcome: b, c, d, e]	Configuration items [Outcome: b, d]
Product configuration [Outcome: c, d, e]	Product configuration [Outcome: c]
Configuration management plan [Outcome: a, c, d]	Configuration management plan [Outcome: a, b]
Release plan [Outcome: a, d]	
	Delivery record [Outcome: c, d, e]
	Configuration management record [Outcome: b, c, d, e]
	Product release approval record [Outcome: c]
	Change history [Outcome: d]
Tracking system [Outcome: b, c, d]	
Configuration management library [Outcome: c]	

Table 24 SPL Requirements Management Process Definition

Process ID	T3
Process Name	SPL Requirements Management Process
Process Purpose	The purpose of the requirements management process is identify and maintain the requirements of the SPL.
Process Outcomes	a) SPL requirements are identified.
	b) SPL requirements are validated.
	c) Consistency and traceability is established between domain level and application level requirements.
	d) SPL requirements are maintained and updated when needed.
	e) SPL requirements are approved, baselined and communicated to all affected parties.
Base Practices	BP1: Discover, review, document, and understand the needs and constraints for the system. [Outcome: a]
	BP2: Define functional, nonfunctional and interface requirements of the SPL. [Outcome: a]
	BP3: Define acceptance criteria for the requirements. [Outcome: b]
	BP4: Analyze the requirements for completeness, correctness and testability. [Outcome: b]
	BP5: Maintain Bidirectional Traceability of Requirements [Outcome: c]
	BP6: Ensure consistency of the requirements. Ensure Alignment Between Project Work and SPL Requirements [Outcome: c]
	BP7: Document the requirements. [Outcome: d]
	BP8: Manage Requirements Changes. [Outcome: d]
	BP9: Share the requirements on a need-to know basis and communicate the requirements. [Outcome: e]
	BP10: Obtain Commitment to Requirements [Outcome: e]

Table 24 SPL Requirements Management Process Definition *(Continued)*

Input(s)	Output(s)
Software Domain Requirements [Outcome: a, b, c, d]	
Software Product Requirements [Outcome: a, b, c, d]	
Interface Requirements [Outcome: a, b, c, d]	
Change Request [Outcome: d]	
	Change control record [Outcome: d]
	SPL Requirements [Outcome: a, b, c, d, e]
	Analysis and review records [Outcome: a, b, c, d]
	Traceability record [Outcome: c]
	Communication record [Outcome: e]

Table 25 Change Management Process Definition

Process ID	T4
Process Name	Change Management Process
Process Purpose	The purpose of the change management process is to identify the effects of changes and act accordingly.
Process Outcomes	a) a change management strategy is established
	b) change requests are recorded and tracked
	c) change impacts are identified
	d) change acceptance criteria are identified
	e) resources for change implementations are planned and allocated
	f) possible regression effects of the changes on different products and components of the SPL are identified
Base Practices	BP1: Develop a change management strategy. [Outcome: a]
	BP2: Establish a change management framework to record and track change request. [Outcome: b]
	BP3: Analyze and document the impact of the change. [Outcome: c, f]
	BP4: Identify validation and verification needs and regression effects [Outcome: c, d, f]
	BP5: Implement changes when approved. [Outcome: e]

Input(s)	**Output(s)**
Configuration item [Outcome: b]	Configuration item [Outcome: b]
Product configuration [Outcome: b]	
	Change management plan [Outcome: a]
Change request [Outcome: b]	Change request [Outcome: b, c, d, e, f]
Tracking system [Outcome: b]	
	Impact analysis report [Outcome: c, d, f]
Configuration management library [Outcome: b]	

Table 26 Test Management Process Definition

Process ID	T5
Process Name	Test Management Process
Process Purpose	The purpose of the Test Management Process is to validate and verify the SPL
Process Outcomes	a) Validation and verification strategies are developed and implemented
	b) Validation and Verification criteria are defined
	c) Regression testing strategy is developed and implemented.
	d) Test activities are performed. This activities may include the testing of various components and products of the SPL due to regression effects.
	e) Defects and problems are identified and reported
	f) Test automation options are identified and implemented
	g) Criteria, activities and results are made available to the stakeholders
Base Practices	BP1: Develop SPL validation and verification strategies. [Outcome: a]
	BP2: Establish regression test strategy. [Outcome: c]
	BP3: Develop SPL validation and verification criteria. [Outcome: b]
	BP4: Conduct and report test activities [Outcome: d]
	BP5: Analyze test automation options. [Outcome: f]
	BP6: Analyze qualification results [Outcome: e]
	BP7: Communicate the activities with the stakeholders [Outcome: g]

Table 26 Test Management Process Definition *(Continued)*

Input(s)	Output(s)
Quality measure [Outcome: b]	
	Validation verification test strategy [Outcome: a, c, f]
	Validation verification test Plan [Outcome: a, c, f]
Quality policy [Outcome: a, c]	
	Communication record [Outcome: g]
	Problem record [Outcome: d, e, g]
	Change request [Outcome: d, e]
Quality record [Outcome: d, e]	Quality record [Outcome: e, g]
Traceability record [Outcome: d]	
Test Requirements specification [Outcome: b, g]	
	Validation verification test analysis results [Outcome: b, d, e, g]
Corrective action register [Outcome: e]	Corrective action register [Outcome: e]
Tracking system [Outcome: e]	
Quality criteria [Outcome: b]	Quality criteria [Outcome: b]

Table 27 Commonality Management Process Definition

Process ID	T6
Process Name	Commonality Management Process
Process Purpose	The purpose of the commonality management process is to manage, control and maintain the commonalities in the SPL.
Process Outcomes	a) Commonality management strategy is established
	b) Reuse opportunities are identified as commonalities among the SPL products
	c) Commonalities are documented, tracked and managed
	d) Commonalities are analyzed and updated.
	e)Consistency is established between common requirements of the products and common interfaces and components
	f) Communication mechanisms between affected stakeholders are established.
	g) Commonalities are implemented and tested.
Base Practices	P1: Define organizational commonality management strategy. [Outcome: a]
	P2: Identify commonalities of the components and products of the SPL. [Outcome: b]
	P3: Define, analyze and maintain the commonalities and their dependencies. [Outcome: c, d]
	P4: Monitor and control changes in the common parts. [Outcome: c, d]
	P5: Communicate the changes and the dependencies of the commonalities with the stakeholders. [Outcome: c, f]
	P6: Ensure consistency between the common requirements and common components and interfaces. [Outcome: e]
	P7: Develop and test the common assets. [Outcome: g]

Table 27 Commonality Management Process Definition *(Continued)*

Input(s)	Output(s)
Common object [Outcome: g]	Common object [Outcome: g]
Domain model [Outcome: b, d]	Domain model [Outcome: d, e]
Project plan [Outcome: g]	
Change request [Outcome: c, d, g]	
	Change control record [Outcome: c, d, f, g]
Software assets register [Outcome: g]	Software assets register [Outcome: g]
SPL architecture [Outcome: a, e]	SPL architecture [Outcome: a, e, f, g]
Business goals [Outcome: a]	
	Communication record [Outcome: f]
Change control record [Outcome: d]	Change control record [Outcome: d, f]
Configuration status report [Outcome: c]	Configuration status report [Outcome: c]
Commonality library [Outcome: b]	Commonality library [Outcome: b, c , g]
Commonality strategy [Outcome: b, f]	Commonality strategy [Outcome: a]

Table 28 Variability Management Process Definition

Process ID	T7
Process Name	Variability Management Process
Process Purpose	The purpose of the variability management process is to manage, control and maintain the variabilities in the SPL
Process Outcomes	a) Variability management strategy is established
	b) Variable requirements of the SPL are identified
	c) Variation points of the SPL are identified
	d) Variability of components, interfaces, classes are managed and tracked.
	e) Variabilities are analyzed and updated.
	f) Consistency is established between variable requirements of the products and variation points
	g) Communication mechanisms between affected stakeholders are established.
	h) Variation points are implemented and tested.
Base Practices	P1: Define organizational variability management strategy. [Outcome: a]
	P2: Identify variabilities of the SPL. [Outcome: b, c]
	P3: Define, analyze and maintain the variabilities. [Outcome: d, e]
	P4: Monitor and control changes in the variable parts. [Outcome: d, e]
	P5: Communicate the changes and updates in the variabilities with the stakeholders. [Outcome: g]
	P6: Ensure consistency between the variable requirements and variable components and interfaces. [Outcome: f]
	P7: Develop and test the variable assets. [Outcome: h]

Table 28 Variability Management Process *Definition (Continued)*

Input(s)	**Output(s)**
Variable object [Outcome: h]	Variable object [Outcome: h]
Domain model [Outcome: b]	Domain model [Outcome: e, f]
Project plan [Outcome: h]	
Change request [Outcome: c, d, e, h]	
	Change control record [Outcome: c, d, e, g]
Software assets register [Outcome: h]	Software assets register [Outcome: h]
SPL architecture [Outcome: a, f]	SPL architecture [Outcome: a, f, g, h]
Business goals [Outcome: a, c]	
	Communication record [Outcome: g]
Change control record [Outcome: e]	Change control record [Outcome: e, g]
Configuration status report [Outcome: d]	Configuration status report [Outcome: d]
Variability library [Outcome: c]	Variability library [Outcome: c, d, g, h]
Variability strategy [Outcome: b, g]	Variability strategy [Outcome: a]

Table 29 Application Engineering Management Process Definition

Process ID	T8
Process Name	Application Engineering Management Process
Process Purpose	The purpose of the application engineering management process is to analyze the system and software requirements and identify the application requirements then to develop and maintain the features and the end product.
Process Outcomes	a) the requirements and the interfaces of the application engineering elements are defined
	b) application engineering and product requirements are analyzed for correctness and testability
	c) consistency and traceability are established between the software product line requirements and application engineering requirements
	d) application engineering blocks that captures the product specific variations in features, capabilities, concepts, and functions are developed and tested
	e) changes to the software and product requirements are evaluated and reflected to application engineering requirements when needed
Base Practices	P1: Specify and define the functional and nonfunctional application engineering requirements and interfaces and document them. [Outcome: a]
	P2: Analyze the application engineering requirements for correctness, completeness, consistency, feasibility and testability. [Outcome: b]
	P3: Develop criteria for product and application engineering assets testing. Use the system and software requirements to define acceptance criteria. [Outcome: b]
	P4: Ensure consistency of software product line requirements to application engineering requirements. Establish and maintain traceability. [Outcome: c]
	P5: Develop and test the application engineering assets and the product. [Outcome: d]
	P6: Identify and Implement the required application engineering assets for product development activities. [Outcome: d]
	P7: Analyze the changes in the software requirements to identify the related application engineering requirements and perform necessary activities. [Outcome: e]

Table 29 Application Engineering Management Process *Definition (Continued)*

Input(s)	Output(s)
Commitment / agreement [Outcome: b, c, e]	
Test data [Outcome: c]	
	Low level software design [Outcome: b, d]
	Customer manual [Outcome: d]
Project plan [Outcome: a]	Project plan [Outcome: a]
Life cycle model [Outcome: a]	Life cycle model [Outcome: a]
Software element [Outcome: a]	Software element [Outcome: a, b, d]
	Traceability record [Outcome: c, e]
Application requirements [Outcome: a, c]	Application requirements [Outcome: a, c, e]
Product requirements [Outcome: b, d, e]	