VVMT:
Value Based Variability Management in Product Lines Using Traceability Information

Submitted by:

Rahila Ejaz

Supervised by:

Dr Naveed Ikram

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad
(2009)

Variability management
Traceability information

Department of Computer Science

International Islamic University Islamabad

Date: 08 - 10 - 2009

Final Approval

This is to certify that we have read the thesis submitted by **Rahila Ejaz 134-FBAS/MSSE/F06.** It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of **Ms in software Engineering**

Committee:

External Examiner:
Dr Saad Zafar
Dean,
Ripah international
university, Islamabad

Internal Examiner:
Ad[redacted]Ashraf
Assistant Professor
IIUI,Islamabad

Internal Examiner :
Shehbaz Ahmed
Assistant Professor
IIUI,Islamabad

Supervisor:
Dr Naveed Ikram
Associate Professor
IIUI,Islamabad

Dedicated to my parents who kindled a spark in my heart which has continued to illuminate my way to a life of struggle for seeking the ultimate truth

A dissertation Submitted To

Department of Computer Science,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the

Degree of *MS Software Engineering*

Declaration

We hereby declare that this Thesis ***"Value Based Variability Management in Product lines using traceability information"*** neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor *Dr.Naveed Ikram*. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, we shall stand by the consequences.

Rahila Ejaz

134-FBAS/MSSE/F06

Acknowledgement

All praise to ALLAH almighty who enabled me to complete this thesis successfully and my utmost respect to His last Prophet (PBUH). I am deeply grateful to my supervisor ***Dr.Naveed Ikram*** for his persistence guidance, support and encouragement. Throughout the process he remained a great source of encouragement and motivation. I am also grateful to my Co supervisor miss ***Salma Imtiaz*** for her valuable feedback and support. I am also thankful to my parents and family members for their continued prayers and support.

No to forget the unrestricted guidance and help of ***Mr. Abid Hussain Awan*** during process evaluation; I would like to express my sincere gratitude to him and his organization.

I am also thankful to my department for providing me with this opportunity to practically test myself after a year of learning and helping me throughout my efforts.

Last but not least, I am grateful to my friends, the technical staff of the research lab and all others for their valuable support.

Rahila Ejaz

134-FBAS/MSSE/F06

Abstract

Variability management is central to the success of software product line engineering. It aims at managing the unavoidable changes requested either from internal or external to the organization, mostly due to changing business values of an organization. Variability management is a multifaceted attribute and involves certain issues. Various solutions are employed to accomplish variability management. These solutions face challenges when employed for variability management, especially business values considerations, and value based traceability support. This thesis firstly identifies core issues of variability management for evaluating variability management solutions after an in-depth study of literature pertaining to variability management issues in product line, and factors that enable and influence variability management in a product line development environment. Based upon these issues, an evaluation of the existing solutions is performed to identify their strengths and weaknesses in a variability management context. A process for best dealing with the variability management problems in software product line development is presented. Finally, the proposed process is being validated by means of a case study.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Variability Management and its Role in Product Lines

Variability management is an essential element of product line change management which plays important role in successful software product line development (Van gurp et al, 2001), (de Oliveira et al, 2005) and (Buhne et al, 2005). It is defined as an ability of a system or set of artifacts to be changed in a specific context (Van gurp et al, 2001).

The need to manage variability increases as size of the product family increases. Findings indicate that inclusion, of new family members, introduces new variations and complex dependency between these variations is essential to manage (Buhne et al, 2005) and (Martin, 2003). Similarly, variability management is integral for managing complex dependency between different artifacts of product lines. Moreover, unanticipated variants may occur due to changing nature of product lines, for which variability management is essential (Mohan & Ramesh, 2002) and (Riebisch & Plilippow, 2001).However, extensive issues of variability management pose serious challenge for effective variability management. In order to address variability management in an effective and efficient way, it is important to understand these issues.

1.2 Problem Domain

Managing the differences between family members of the product line is known as variability management. In product lines, knowledge about variability is dispersed among different heterogeneous stakeholders (Dhungana et al, 2006).Therefore for extracting the realistic variability information, stakeholders' involvement is essential (Rabiser et al, 2008). Literature highlights that changing nature of product line development generates extensive amount of technically possible variants and associated variation points (Batchman & Bass,2001), (Buhne et al, 2005) and (Theil & Heindel,2002). However, all of this variability is not relevant to the customer (Rabiser et al, 2008) and (Markus & vancza, 1998).This requires value based variability management. Value based variability management means variability management is a collaborative process that relies on involving stakeholders because variability knowledge is dispersed among heterogeneous stakeholders (Rabiser et al, 2008).

Literature survey shows that changing nature of requirements, technology or business/market strategy are major sources of variations (Ajila et al, 2004) and (Ahmed & Capretz, 2007).Changes in business and market strategy lead to changes in organization process (Taulavuori et al, 2004) and (Schmid & John, 2001). Technology innovation (change in technology/variability) has a direct influence on product quality (McGregor, 2003).Besides technology and market trends, requirements (features) variability is directly related to customers' satisfaction because it represents the users' intents (Lee & Muthig, 2006). This indicates that variability management means achieving customers' satisfaction, improving process and product quality. However achieving customer satisfaction and improving process and product quality are business values of an organization (Samad et al, 2008) and (Roy & MSS, 2009). Recent researches on variability management (Rabiser et al, 2008) also emphasizes on variability identification based on business objectives and marketing decisions. This indicates that for effective variability identification, we need business values consideration.

The findings of literature survey (Buhne et al, 2005), (Berg et al, 2005) and (Mohan & Ramesh, 2002; 2003; 2007) indicate that traceability of variability is essential for efficient variability management as it assists in effective change management by linking variation to its variation points. Traceability of variability helps in understandability, maintenance and evolution of product lines, however traceability of variability means establishing and maintaining traceability links between various artifacts of the product line. This is because variability is first introduced in requirements (Lee & Muthig, 2006) and (Mohan & Ramesh, 2007), realized at design time (Van gurp, et al, 2001) and (Sinnema et al, 2004) and then implemented using various implementation mechanisms (Svahnberg & Bosch, 2000) and (Bosch et al, 2004).This indicates that for variability management, traceability should be established between several artifacts of product line. However, establishing and maintaining traceability links places big burden since documentation generated during the product line development lifecycle is of significant size and complexity, and specified at different levels of abstraction and granularity

(Kim et al, 2005). This implies that though traceability based variability management is viable for efficient variability management but traceability in a value neutral setting is not only difficult but also time consuming. It is also evident from literature that traceability for massive documentation without considering the value dimension is not productive for organizations (Heindl & Biffl, 2005) and (Ahn and Chong, 2006) and value-based tracing can help to find a subset of traces that saves proportionally more time and cost.

This concludes that for effective variability management we need value based variability management that also incorporates traceability of variability and business values consideration. However, no such process /method /approach exist that satisfy these requirements. The proposed research revolves around these needs.

1.3 Research Aim

This research aims at representing the 'value based' trends in product lines with a focus on traceability based variability management during product line software development. It is hoped that this task will highlight the significance of value based variability management for product line industry. Research work also intends to identify the issues /factors that are important for efficient variability management. Besides these, research work also presents the advantages and disadvantages of making 'value' an integral part of software product line variability management. Finally, all together, this research aims at proposing a process for managing variability in a value based manner as a final output.

1.4 Research Questions

The aim of this research is to understand the issues of existing variability management solutions and suggest a process after analysis. For this purpose, our research work is based on the following questions.

- What are the core issues of variability management?
- How effective are the current variability management processes, with respect to the core issues of variability management?
- What possible measures can be taken to improve the effectiveness of these processes?

1.5 Research Method

In order to answer the research questions, the adopted research method will consist of four steps.

1. Literature review
2. Analysis of findings
3. Presenting a solution
4. Validation of solution

Following is the list of tasks that will be performed:

- A study of literature to understand the concepts of variability management and various issues pertaining to it.
- A study of prominent work related to variability management to gauge the effectiveness of the existing approaches for supporting the core issues of variability management.
- An analysis and evaluation of variability management approaches (particularly processes) to discover the level to which they are impacted by variability management issues.
- A possible process to improve the use of variability management process in product line development.
- Validation of the proposed process through a case study.

1.6 Thesis Outline

The remainder of this thesis is structured as follows.

Chapter 2: This chapter gives an introduction of the problem domain which is variability management. It discusses the motivations behind the practice of variability management in software product line development. It reports the various issues that hinder the effectiveness of variability management as found in the literature. Special focus is on traceability of variability, which is deemed as the major issue of handling variability management. This chapter further gives a brief introduction to widely employed variability management solutions and also evaluates them to determine their effectiveness for variability management.

Chapter 3: The third chapter discusses the process that has been proposed after survey of literature and analysis of evaluation. A process is proposed that manages variability in a value based manner using traceability information.

Chapter 4: The fourth chapter performs the validation of the process proposed in the previous chapter by means of a case study. The chapter provides an introduction to the selected case, its relevance and the methods for data collection. It reports the problems encountered in the life of the software product line development and discussed how these problems could have been avoided by application of the proposed process.

Chapter 5: The final chapter of the thesis provides a conclusion by discussing the contributions of this thesis and how these contributions answer the research questions posed in the first chapter. It also discusses the possible directions for future research that can assist in further improving the variability management process in software product line development.

Chapter 2

Literature Survey and Analysis

2.1 Introduction

Change is unavoidable in software product line development. Success of product line depends on efficient change management for which variability management is prerequisite. Variability is a key difference between traditional software engineering and product line engineering. Variability is defined as the *ability of a system or an artifact to be changed or customized in a specific context* (Mohan & Ramesh, 2002).Variability and its management is a difference between success and failure of the product lines (Berg et al, 2005).In product lines, variety of products are produced in shorter time and low cost, due to systematic reuse of platform(also called core architecture, product line architecture) across multiple product variants but managing variations across these product variants demand efficient and effective variability management (Mohan & Ramesh, 2007).

Variability management is an issue of product line change management which is practiced to manage the inevitable differences among product family members (Mohan & Ramesh, 2003).The importance of incorporating variability management during the software product line development is proven to be essential and critical for the product line success. (Vangurp et al,2001),(De Oliveira et al, 2005), (Berg et al, 2005), (Sinnema et al, 2004) and (Mohan & Ramesh,2002).The need to manage variability increases as size of the product family increases; inclusion of new family members introduces new variations and complex dependency between these variations is essential to manage(Buhne et al, 2005) and (Martin,2003).Similarly researchers emphasize on incorporating variability management during product line software development to deal with issue of dependency management between different artifacts due to complex dependency between variations, conflicting customer requirements (Mohan &

Ramesh,2002) and (Bayer & Widden, 2001), unanticipated variants as a result of new requirements (Mohan & Ramesh, 2002) and missing variation points in architecture (Loesch & Ploedereder, 2007).

Variability management is a multifaceted attribute and incorporates certain issues. Without understanding such issues efficient variability management is difficult to achieve. For this purpose attempts have been made in literature. Variability identification, representation and dependency management both implicit and explicit are general issues of variability management reported in literature (Jaring & Bosch, 2002). Findings (Bosch et al, 2001) and (Van gurp et al, 2002) also reveal different levels of variability and certain issues with every phase of life cycle. This indicates that variability management includes certain issues and for effective variability management, we must have in depth knowledge of different issues associated to variability management. For this purpose in the next section, we have presented core issues of variability management. They are called core issues because they are frequently reported in literature related to variability management.

2.2 Core Issues of Variability Management

This section describes core issues pertaining to variability management.

2.2.1 Variability Identification

Variability identification is first step of variability management and is done in domain engineering phase (Vangurp et al, 2001). It implies identify point of differences between products (De Oliveira et al, 2005) and (Mohan &Ramesh, 2002).

Variability identification in an early stage of life cycle leads to better customization (Theil & Heindel, 2002) and improves company economy by increasing products variety. It is worth noting that identification of variability includes identification of all causes of variability. Frequently reported causes of variations are change in market strategy, advances in technology and change in customer requirements (Sybrenet al, 2008), (McGregor, 2003), (Ajila et al, 2004), (Ahmed & Capretz, 2007). Findings also indicate that stakeholders of the product line often have conflicting requirements which is a main source of variations (Bosch, 1999), (Mohan and Ramesh, 2002) and (Ajilla et al, 2004), so these sources along with their requirements should also be identified. Moreover, variability information is distributed among heterogeneous stakeholders therefore stakeholders' participation is essential for effective variability identification. In addition to this, the step wise extension of new requirements may introduce new variants (Riebisch & Plilippow, 2001).These variants should be identified as early as possible to avoid the architectural degeneration of product lines. Jirapanthong & Zisman, (2005) reports that product line members are evolvable and their evolution aspects should be identified to decide when new member should be incorporated into product line.

For effective variability identification, it is important to identify the business values of the organization. We infer it on the basis of literature survey which indicates that variations in product lines are because of changing business needs/values of the organization (Taulavuori, et al, 2004) and (Inoki & Fukazawa, 2007). Findings indicate that variations are usually in the form of requirements, technology or business/market strategy (Ajila et al, 2004) and (Ahmed & Capretz, 2007).Changes in business and market strategy lead to changes in organization process. This type of variability facilitates in process restructuring according to the market trends. Furthermore technology innovation (change in technology/variability) has a direct influence on product quality (McGregor, 2003). Besides technology and market trends, requirements (features) variability is directly related to customers' satisfaction because it represents the users' intents. This indicates that variability identification assists in achieving customers' satisfaction, improving process and product quality. However achieving customer satisfaction and improving process and product quality is related to the business values of the organization (Carvens et al, 1997), (Ross & MSS, 2009), (Samad et al, 2008), (Khalifa, 2004).Recent researches on variability management (Rabiser et al, 2008) also emphasize on variability identification based on business objectives and marketing decisions. This indicates that for effective variability identification, we need to identify the business values of the organization. It will help to identify all causes of variability. Ahmed & Capretz, (2007) demonstrated the relationship between business factors and software product line development. They concluded that identification of business values are critical as the rapid and continual changes common to the present business environment has a profound impact on successful software product line development and these changes are due to the changing nature of business values. Variability identification based on business objectives also highlights the intents of variability management (Whether Variability management is practiced for customer satisfaction or for the process improvement etc).

Summarizing it all, for effective variability identification, three factors are important which are

- Identification of requirements variability because change in requirements affect the whole system.
- Stakeholders' Participation
- Business values consideration

2.2.2 Variability Representation

The explicit representation of variability is proven to be essential in managing product lines variability(Sinnema et al, 2005) and (Jaring and Bosch, 2002).There are many ways to represent variability ranging from feature models, meta models, ontology of variability to UML notations (Vangurp et.al, 2000), (Andreas & Klaus, 2006), (Batchman et al, 2004), (Pohl, & Metzger, 2006), (Steffen & Andreas, 2002), (Mohan & Ramesh, 2003) and (De Oliveira et al, 2005). Literature survey reports that variability plays its role in every phase (Sinnema et al, 2004).Therefore different modeling approaches have been proposed by authors to represent variability in different phases. E.g., many feature modeling approaches are used to represent variability in problem space (Kang et al, 1998), (Becker, 2003), solution space (Weiler 2003) and source code level (Czarnecki & Eisenecker, 2000).Variability representation in design and architectural level is discussed by (Batchman & Bass, 2001). In addition to these, different UML notations are also used to represent variability (De Oliveira et al, 2005), (Atkinson c, et al., 2002), (Gomaa, & Webber, 2004).Though variability representation is approached in different directions but they have a single purpose i.e. to facilitate the assessment of impact of selection during product derivation and changes during evolution (Sinnema et al, 2005).It is important to understand that variability representation at different phases have different purpose. E.g. at requirement level variability representation helps in understanding the causes of variability, however, variability representation in design and architecture defines the points of variations that support those variations which are introduced during requirements. It is also worth noting that with the help of variability representation, technical staff of product line organization is provided with all the alternatives and their solutions during product derivation (Jaring & Bosch, 2002).Although, variability representation is central to variability management, no standard notations are available and working bodies are lacking with a common frame of variability representation (Jaring & Bosch, 2002) and (Vangurp et al, 2001).

2.2.3 Variability Realization

Among others, Variability specification and its realization are important factors of variability management (Becker, 2003). Both aspects are important but difficult to attain as variability is a multifaceted attribute that is a part of different phases of product line lifecycle. Specification of variability includes externally visible characteristics of variability and is achieved usually through feature models. Variability is realized usually at the implementation level. During variability realization, impact of variations on different software assets are understood and such variability is supported through appropriate implementation mechanisms (Becker, 2003).variability realization includes variation points, their associated variants, effects of variants on variation points and tasks attached to an individual variant point (Kim etal, 2005). In other words, variability realization involves details of variability so complex dependency between different variations can be understood easily. Literature suggests various implementation mechanisms to realize variability (Jaring & Bosch, 2002) and (Estublier & Vega, 2002). Becker, (2003) emphasizes that variability realization is predominant factor of variability management as it ensures variability implementation into product lines. Bosch et al, (2004) illustrates the importance of variability realization and concludes that without variability realization it is difficult to see the impacts of changes.

2.2.4 Product Instantiation

The current trend of globalization is pressuring IT industries to improve production rate while shortening time to the market. For this purpose, product line based software development is introduced in which variety of products are produced by managing variability and exploiting commonality. But managing the differences during product instantiation is not an easy task especially when products are large in number and complex in nature (Berg et al, 2005). Products instantiation means creating a product by reusing core assets of product line (Bosch, 2000). However, product instantiation is not always fully supported by reuse of core assets. Findings (Bayer &widen, 2001) and (Kim et al, 2005) report that some products have special features which are not supported by core assets. As a result, they have application specific variability which is important to manage in order to deliver customizable product. Findings (Kim et al, 2005) show that product derivation is incomplete until application specific variability is not coped. To mange such variability, we require to analyze the application specific features which are not supported by core assets.

Systematic and planned reuse is a distinguishing characteristic of product line based development. Product instantiation by reusing core assets is proven to be efficient approach (Mohan & Ramesh, 2002) and (Taulavuori, et al, 2004). But this approach has two pitfalls. First is identification of reusable components and second problem is selection of appropriate configuration among the components (Mohan & Ramesh, 2002), (Taulavuori, et al, 2004) and (Estublier, & Vega, 2005).

Product line is supposed to be successful if components present in core assets are used maximally to develop the new product variants but it happens rarely. Complex dependency between variation points of components makes components configuration a tedious task (Theil & Heindel, 2002) and (Mohan & Ramesh, 2002). Moreover, third party components are increasingly being used in product line based software engineering that also complicates the variability management process (Taulavuori, et al, 2004).

2.2.5 Dependency Management

Complexity is inherited attribute of product lines. Variety of artifacts is produced during domain engineering and application engineering. These artifacts are not isolated and are dependent on one another. Result is complex dependency between different artifacts. Since variation is always a part of some artifact therefore in order to manage variability, dependency management becomes critical. In addition to this, dependency management is important for variability management as variability cannot be localized and it has widespread impacts on product lines artifacts (Becker, 2003).

Literature highlights that dependency management during components configuration is essential due to complex dependency between different components because one variation point may be associated to more than one component- probably to be used in different context (Theil & Heindel, 2002)

In addition to components dependency, feature tangling and scattering is another issue of dependency management. It involves dependency between feature and an architectural component (Theil & heindel, 2002), (Sinnema, 2005) and (Loesch & Ploedereder, 2007). One feature may be implemented by more than one component result is feature scattering. Similarly one component may be responsible for more than one feature. Result is features tangling. Both creates problem during maintenance due to high rate of dependency (Loesch & Ploedereder, 2007). Literature (Berg et al, 2005) reports that variability specified in the form of features is realized in components. To achieve efficient variability management dependency between the two needs to be managed.

Findings (Ajilla et al, 2004) and (Staples, 2004) reveal that variability management requires consistent change integration throughout the software development lifecycle which is difficult due to *feature interaction* (Vangurp et.al, 2000) and (Lee & Muthig, 2006). Because features dependency is complex (Mohan &Ramesh, 2003) and impact analysis is difficult to perform in case of changed features.

2.2.6 Traceability of Variability

As suggested repeatedly in literature, for efficient variability management, information about variability should be documented explicitly (Behune et al, 2005), (Berg et al, 2005) and (Mohan & Ramesh 2002), (Mohan & Ramesh, 2003) and (Mohan & Ramesh, 2007). This is because simply identifying and modeling variability among the products in a product line does not define what features are associated with what products, as well as what dependencies and interrelationships exist among variability (Berg et al, 2005). However, for effective variability management, it is essential to understand and identify the interrelationship between different fragments of knowledge about variability (Mohan & Ramesh, 2007). In order to do so, traceability of variations becomes a compulsion.

The need for variability documentation (traceability) is also highlighted by the fact that product line software development generates wide range of artifacts. Traceability (documentation) identifies the relationships between these artifacts. They contribute to the better understanding of the system as they link its distributed knowledge. Furthermore, traceability supports understanding, maintenance and evolution of variability by establishing links between variability at different levels of abstraction and across development phases(Berg et al, 2005), (Ajilla,et al, 2004) and (Bayer & Widen, 2001).

Literature highlights that explicit documentation of variability aims at keeping track of variability information from its identification to its implementation (Sinnema et al, 2005), (Berg et al, 2005). In case of variability identification, traceability information provides knowledge about sources of variations and their requirements, evolution aspects within a product member, sources of variations demanding conflicting quality attributes and variable aspects between different product members(Bosch,1999), & (Ajilla et al, 2004) (Jirapanthong & Zisman, 2005), (De Oliveira et al, 2005) and (Mohan &Ramesh,2002). Furthermore traceability information informs about the extent to which reusability can be acquired. This facilitates in product instantiation.

In case of variability representation and realization, traceability information provides a comprehensive overview of variability at requirements, design and code level (Berg et al, 2005). Variability representation defines variability, whereas variability realization is practiced to understand the impact of variations on different software assets (Becker, 2003). Findings also indicate that as far as impact analysis is concerned, traceability is a viable approach (Imtiaz, 2008). It provides comprehensive overview about system dependency and thus facilitating in efficient impact analysis. This implies that traceability information facilitates in variability identification, variability representation and variability realization.

Besides these issues, traceability information also facilitates in product instantiation and dependency management. For product instantiation, systematic reuse is essential (Mohan & Ramesh, 2002). Reusability requires necessary information about reusable components, framework and core requirements. Furthermore it needs information about constraints on architecture, design and implementation. Traceability links between different fragments of knowledge provides this information to understand different issues of reusability. Such as when and how framework can be adapted and integrated, how reusable components can be configured to support the variability and so on.

Besides other issues, traceability also plays a vital role in dependency management. Findings indicate that comprehensive overview, of interdependency between various artifacts, is essential for dependency management (Riebisch, & Plilippow, 2001). Traceability due to its support of establishing links between different artifacts provides comprehensive overview of, interdependency between different artifacts. Literature reports that complex dependency structure is a big hinder to effective evolution of product line and traceability resolves this issue by providing information about the dependency between different artifacts (Becker, 2003). This information helps in understanding the dependency structure and its associated issues (Riebisch & Plilippow, 2001).

Findings indicate that traceability based variability management has been highly advocated in literature (Mohan & Ramesh, 2002; 2003; 2007), (Van Gurp et al, 2000), (Metzger and Pohl, 2006), (de Oliveira et al, 2005). Traceability based variability management supports understanding, maintenance and evolution of variability by establishing links between variability at different levels of abstraction and across development phases (Berg et al, 2005) and (Mohan and Ramesh, 2007). However variability management approaches in use tend to ignore the explicit traceability links for all the core issues of variability management. Among them research work of Kim, et al, (2005) is the only one who explicitly states the traceability information for variability management however such set includes traces only for variability identification and realization and ignores other important issues of variability management that are reported in literature.(Becker, 2003), (Kim, et al, 2005) and (Estublier & Vega, 2005). For this reason we have reviewed the literature to identify traceability information for the major c issues of variability management namely variability identification, variability representation, variability realization, dependency management and product instantiation as shown in table 2.1.

2.2.6.1 Traceability Information for Variability Management

This section presents traceability links for each core issue and its sub issues .These traceability links assist in providing a comprehensive overview about the type of artifacts and system elements required to tackle the different issues of variability management.

A. Traceability information for variability identification

It involves three issues as discussed below.

- **Requirements variability**

  (Vangurp etal, 2001) identifies that variability is generated in the form of requirements. It is then refined by feature diagram (Lee & Muthig, 2006) in the form of alternative, mandatory or optional feature (Bachmn & Bass, 2001) and (Vangurp etal., 2001).This indicates that *from requirement to feature* is an appropriate trace for variability (functional) identification.

  Product family has numerous members, each having its own set of requirements. Differences between them are inevitable which are utterly essential to identify in order to manage. One way to capture such differences is maintain traceability between artefacts (horizontal/vertical) of product members as discussed by (Jipanthrohg & Zisman, 2005).This trace helps to identify the differences between different product members by comparing the artifacts. In other words establish trace *between documents of product members (e.g. req to req, design to design, req to design etc.)* to identify the differences between product members.

- **Conflicting Quality Attributes Requested By Stakeholders**

  Mohan and Ramesh, (2002) shows that it is important to trace the sources that demand conflicting quality attributes as it facilitates to justify the implementation of same components for different functionality. We can capture such information with the help of maintaining *from origin to conflicting quality attribute* trace. This trace is also helpful to identify all the sources of variations either internal or external.

- **Evolution Aspects within a Product**

  Jipanthrohg & Zisman, (2005) discusses that importance of identifying evolution aspects in a single product member. Such evolution is identifiable by maintaining the relations *between documents of the same type for the same product member.* This trace shows change incorporated in requirements and its effect on other artifacts. As a result, it provides a complete picture of change within a product.

B. Traceability information for Variability Representation

It is clear that variability representation is not bounded to a single phase but it is an attribute of all phases of software development life cycle. It implies that this issue encompass artifacts of both problem and solution space. Berg et al, (2005) defines requirements related artifacts as part of problem space and architecture and implementation related artifacts in solution space. This indicates that variability representation involves following trace; *from requirements to architecture to design to implementation.* This trace defines variability of requirements in the form of features. Also it describes variability in design as variation points and how this variability is then represented in design documentations and finally variability representation at code level; thus covering the whole domain i.e. problem and solution space.

C. Traceability information for Variability Realization

In the context of variability realization, Kim, et al., (2005) has mentioned the artifacts and traces between artifacts. Commonality and variability specifications and Core asset model are stated artifacts for variability realization. *From Commonality and variability specification* to *core asset model* is reported by Kim, et al (2005). Variability is specified in an abstract form by commonality and variability specifications which is then realized and refined by core asset model. In addition to this Estublier & Vega, (2005) defines that *from feature to product line architecture* trace helps to realize variability at design time.

D. Traceability information for Product Instantiation

It involves three issues as illustrated below

- **Application Specific Variability**

  Product derivation is incomplete until application specific variability is not handled. To mange such variability we require to analyze the application specific features which are not supported by core assets. For this purpose Kim et al, (2005) suggests application analysis model which includes details of application specific requirements. As a second step we need to identify the options and alternatives available to satisfy these requirements. For this purpose, researchers suggest decision model (Berg et al, 2005), (Kim, et al, 2005), (Metzger & Pohl, 2006). Decision model contains details of alternatives and solutions in the form of variations and variation points. It implies that maintaining the trace *from application analysis model to decision resolution model* helps to map the application specific features to application specific variability and resolves the issue of application specific variability.

- **Identification of Reusable components**

  Systematic reuse is proven to be an effective approach during product instantiation for which identification of reusable assets is critical. Jirapanthong & Zisman (2005) identified that maintaining the link *from product line architecture to product member* assists in identification of reusable components. Estublier & Vega, (2005) support this idea and defines that maintaining the *link between abstract product line architecture and reusable components* facilitates in extracting the functional components of product line which are then used for product instantiation.

- **Components Configuration**

  Reusable components should be configured appropriately to reap the full benefits of reuse. At the time of configuration, various choices are available and selection of right choice is essential to instantiate a right product. To do this, Bosch (1999) emphasizes to *maintain the alternatives and constraints that lead to configurability of various components*. Mohan & Ramesh, (2002) argued that maintaining such information also provides rationale for various architectural design decisions. We call this trace *from origin to architectural decision*. By origin we mean alternatives and constraints of configuration. We relate it to architectural decisions because reusable components are designed during architecture.

E. Traceability information for Dependency Management

During dependency management three issues are reported (Theil & Heindel, 2002), (Loesch & Ploedereder, 2007), (Berg et al, 2005) (Becker, 2003).

- Component dependency (between components)
- Features tangling and scattering (between features and components
- Feature interaction (between features)

This implies that maintaining the trace *from component to component* addresses the issue of component dependency and *from feature to component* is required for the issue of feature scattering and tangling. For feature interaction we need *from feature to feature* trace.

Table2.1: Traceability information for core issues of Variability Management

Traceability link	Issue	Contributor
	Variability identification	
From Requirement to feature.	Variability in Functionality	(Vangurp etal, 2001), (Lee and Muthig, 2006), (Bachmn & Bass,2001), (Jipanthrohg & Zisman, 2005)
From Origin to Conflicting quality attributes	Conflicting quality attributes by customers	(Mohan & Ramesh, 2002)
From Feature to architectural decision to design documentation to	Evolution aspects within a product	(Jipanthrohg & Zisman, 2005)

implementation	member	
Relation between documents of same/different type for different product members	Variations among different product members	(Jipanthrohg & Zisman, 2005)
From requirements to design to implementation	**Variability Representation**	Berg et al, (2005)
From Commonality and variability model to core assets From Feature to product line architecture.	**Variability Realization**	Kim, et al., (2005), (Estublier & Vega, 2005)
	Product Instantiation	
From application analysis model to decision resolution model	Application Specific Variability	(Berg et al, 2005), (Kim, et al, 2005), (Metzger & Pohl, 2006)
From product line architecture to product member From abstract product line architecture to reusable component	Identification of reusable components	(Jirapanthong & Zisman 2005), (Estublier & Vega, 2005)
From origin to architectural decision	Components Configuration	(Bosch 1999), (Mohan & Ramesh, 2002)
	Dependency Management	
From component to component	Component dependency	(Theil & Heindel, 2002), (Loesch & Ploedereder, 2007), (Becker, 2003)

From feature to component	Feature Tangling and Scattering	(Loesch & Ploedereder, 2007), (Kathrin et al, 2005)
From feature to feature	Feature Interaction	(Kathrin et al, 2005), (Becker, 2003)

2.2.7 Support for Value Based Traceability

Traceability of variability means establishing and maintaining traceability links between various artifacts of the product line. This is because variability is first introduced in requirements, realized at design time and implemented later. This indicates that for variability management, traceability should be established between several artifacts of product line. However, establishing and maintaining traceability links between different artifacts in product lines places big burden since documentation generated during the development lifecycle is of significant size and complexity, and are specified at different levels of abstraction and granularity (Jirapanthong & Zisman, 2005). This implies that though traceability based variability management is viable for efficient variability management but traceability in a value neutral setting is not only difficult but also time consuming.

It is also evident from literature that traceability for massive documentation without considering the value dimension is not productive for organizations (Ahn, & Chong, 2006). Findings (Heindl & Biffl, 2005) also show that value based tracing took around 35% effort of full tracing. This is because value-based tracing can help to find a subset of traces that saves proportionally more cost and time than it loses benefit. This implies that value based traceability is essential for efficient and efficient variability management.

From the above discussion, it is concluded that variability management is a multifaceted attribute and it involves following issues.

- Variability Identification
- Variability Representation
- Variability Realization

- Dependency Management
- Product Instantiation
- Traceability support
- Value Based traceability support.

These are called core issues because these are frequently reported in literature of variability management (Bosch et al, 2001), (de Oliveira et al, 2005), (Mohan & Ramesh 2002; 2003; 2007), (Theil & Heindel, 2002), (Metzger and Pohl, 2006), (Lee & Muthig, 2006), (Estublier & Vega, 2005).Literature review also reveals that a good deal of work has been done in the area of variability management with many people still working on it. Among them the most significant contribution is done by (Vangurp 2001). Then (Sinnema, 2005) and (Rabiser et al, 2008) are also among other contributors making variability management an understandable and un-ignorable concept for the product line engineering. De Oliveira et al (2005) also share his contribution in describing and improving the concept of variability management by proposing a UML based variability management process. Besides, there are several others who worked in the same lines (Mohan & Ramesh, 2003), (Metzger and Pohl, 2006), (Estublier& Vega, 2005), (Buhne et al, 2005) and (Deelstra et al, 2009).

In the next session, we will present the evaluation of variability management processes based on the core issues to gauge their effectiveness with respect to variability management.

2.3 Evaluation of Variability Management Processes

We have evaluated the variability management processes on the basis of following parameters.

1. Variability identification
2. Variability representation
3. Variability realization
4. Product instantiation
5. Dependency management
6. Traceability Support

7. Value Based Traceability support

First we will present a brief description of the process and then we will describe our evaluation based on the parameters discussed in section 2.2

2.3.1 Evaluation of UML Based Variability Management Process

De Oliveira et al, (2005) discuss the variability management process. The proposed process consists of the following activities: 1) *Variability tracing definition, 2) Variability identification, 3) Variability delimitation* and *4) Identification of mechanisms for variability implementation.* These four activities use different uml artifacts for different purposes. E.g. *variability tracing definition* takes the use case and the feature models as input and generates the variability tracing model as output; whereas Variability *identification* takes the use case, the static type, and the feature models, plus the component model as input and generates the same artifacts with the variability identified as output. The effectiveness of the proposed process was evaluated with a case study. The purpose of the case study was to re-define the existing product line with the introduction of the proposed variability management process with the intention of observing the impact on the number of variability identified. Results indicate that variability management can be carried out effectively if *variability identification and traceability of variations* is carried out properly. The evaluation of the process is as below.

- Process identifies variability in the form of features and uses feature diagram and use case model for *variability identification.* However the process only deals with requirements variability. Support for business values consideration and stakeholders 'participation is absent.
- *Variability representation* is well established in the process. The process uses different stereotypes to represent variability on different uml artefacts. Moreover, the process uses different uml relations to represent the type of variability between the variation points and variants. Besides these, process uses uml notes to support the graphical representation of variability.

- For *variability realization*, process defines binding time in *variability delimitation* activity. The definition of binding time facilitates in the selection of implementation mechanism. Also the process defines *identification of mechanisms for variability identification* activity which provides variability implementation mechanisms along with implementation strategy and binding time for variability implementation.
- *Dependency management* is supported by the process. It defines two relations *mutex* and *requires* indicating the dependency relationship between variants. Moreover, the relationship between variant artifacts of the product line is defined in *variability tracing and control* activity.
- *Product instantiation* is not discussed explicitly. However analysis reveals that traceability support facilitates the process of product derivation as it keeps track of all the information that can be reused later.
- *Traceability support* is well established in the process. The process presents variability tracing definition. It uses use case model and the feature model. The tracing is possible because the features are related to the use case model that is related to the static type model and the component model. This way traceability between different artefacts is established.
- Support for *value based traceability* is absent.

2.3.2 Evaluation of VBE Variability Management Process

Rabiser et al, (2008) presents a value-based process for eliciting product line variability which aims at integrating the technical and business perspectives in product line engineering. The proposed process emphasizes on identifying and understanding the variability of existing systems, capturing the tacit knowledge of different stakeholders and choosing the right level of granularity for modeling variability. It consists of nine steps and seven different outputs. First two steps are just to start up a process in which participants are divided into different group and subsystems are assigned to them for which they are responsible to define the functionality. In the third step (describe significant variability of the subsystem), each subgroup defines the significant variability of the subsystem which is further discussed in the fourth step. After variability identification, impact of the identified variability on engineering and/or business is discussed e.g. reason for importance of variability, possible consequences of not taking variability into account etc. After capturing the impacts of variability, variability is prioritized. In the last two steps, questions are discussed to decide about the elicited variability. The purpose of this whole process is to capture the most relevant variability so that it can be used as a starting point to model the variability. The evaluation of VBE on the core issues of variability management is as below.

- This is the only process that involves stakeholders in the variability identification process. It emphasize on involving people with an intimate knowledge about subsystems 'variability. Therefore this process not only provides requirements variability but also supports stakeholders 'participation for *variability identification*. However, support for business values consideration is missing.
- *Variability representation* is present in the form of variability cards. They describe the differences that occurred in the last few projects. Variability from other subsystems that influenced the local variability of the subsystem is also described through these cards. Variability in the form of standard notations like feature model are not discussed. Moreover variability both for problem and solution space is not discussed. For this reason, we conclude that support for variability representation is partial.
- The process *realizes variability* in the form of variation points. Variation points are elicited in the form of questions representing decisions to be taken during product derivation/instantiation. Discussion about implementation mechanisms and variability realization into lower levels of abstraction is not discussed. For this reason, we conclude that support for variability realization is partial.
- *Dependency management* is missing.
- *Product instantiation* support is present in the process. The process emphasizes on capturing the consequences of variability for product instantiation. Moreover, process aims at capturing the most important variability that addresses most important business impacts in application engineering. Documentation of

rationale and other information regarding variability also facilitates in systematic reuse of this information.

- Process provides means of documenting the rationale and other information related to identify variability. This way it provides *traceability of variability*. But the process made no mention about establishing links between different pieces of variability information however researchers emphasizes that without establishing the links between different pieces of information the purpose of traceability is not fulfilled(Berg et al2005),(Ajilla et al,2004).
- Concept of value based traceability is not present in the process.

Findings indicate that explicit processes for variability management are limited in number. UML based variability management and VBE are the only two processes of variability management that explicitly deals with variability management and incorporates various issues of variability management. Besides processes, variability management is also discussed in the form of models, approaches, method etc (Jirapanthong & Zisman, 2005), (Kim et al, 2005), (Vangurp et al, 2000), (Sinnema et al, 2005), but none of the dimension is mature enough and the body of knowledge in each dimension is thin. For this reason, instead of focusing only on one dimension, we have evaluated all the prominent work of variability management which includes an approach, models, method and traceability based variability management literature. The reason for their inclusion was twofold; to gain an in-depth understanding of prominent works on variability management and secondly to propose an efficient and effective process that eliminates the gaps present in these solutions.

2.4. Evaluation of Variability Management Approach

This section presents the evaluation of variability management approach.

2.4.1 Evaluation of COVAMOF: An approach to Variability Management

Sinnema et al, (2005) proposed a variability modeling approach for managing the complexity of variability. The approach is proposed to support the effective product derivation by explicit representation of variability. Researchers (Sinnema etal, 2005) identify the problems related to variability modeling and established eight requirements for variability modeling approaches. These requirements are 1) explicit representation of variability, 2) explicit representation of dependencies across all layers of abstractions, 3) hierarchical organization of variability, 4) tool support for effective variability modeling, 5) traceability of variability across all layers of abstractions, 6) tight coupling between model and the artefacts in case of change, 7) multiple views on variability and 7) interaction between dependencies. Purpose of all these requirements is to reduce the complexity of variability by providing explicit representation of variability at all layers of abstractions and simplifying the interaction between different levels of variability. It provides CVV (COVAMOF variability view) which resolves all the eight requirements. Evaluation of the approach is as below.

- *Variability identification* is not discussed
- *Variability representation* is well supported by the approach. Variability representation across all layers of abstraction is provided by covamof variability view (CVV). CVV gives variability representation across features, architecture and components implementation layers.
- *Variability realization* is supported in two ways. Variation points in a software product family can be realized by variation points on a lower level of abstraction, or by a realization mechanism in the product family artefacts. For the first case, realization relation is introduced. The realization relation in the CVV defines a set of rules that describe how a selection of variation points directly depends on the selection of the variation points in a higher level of abstraction in the product family.
- *Dependency management* is not mentioned explicitly. Realization relation supports traceability between different layers of abstraction and therefore dependency between different artefacts can be viewed and managed. Dependency between different artefacts supports feature tangling and scattering, however feature interaction and component dependency are missing.
- *Product instantiation*/product derivation is supported in two different ways.

- ➢ Variation point view provides comprehensive overview of different alternatives available thus facilitating the process of product derivation.
- ➢ For effective product derivation, overview of interaction among different artefacts is required. The approach supports this overview by artifact relation in which artefacts are organized hierarchically. This eliminates the cognitive complexity and therefore improves understanding. Both variation point view and artifact relation promotes systematic reuse of variability information.

- It presents *traceability* through realization relation in the CVV that provides mapping between choices on a higher level of abstraction and choices on a lower level of abstraction.
- *Value based traceability* is absent.

2.5 Evaluation of Model Based Variability Management

This section presents the evaluation of model based variability management solutions.

2.5.1 Evaluation of VMDE: A Model for Variability Management in Domain Engineering:

Kim et al, (2005) present variability management model to manage variability over the lifecycle of domain engineering with an emphasis of identify the artifacts and their relationship and maps variability into product family assets. Researchers highlight the four issues pertaining to variability management and proposed a model to encounter these issues. The highlighted issues are variability representation through meta- modeling, mapping between variability and product family assets, generic modeling extensions for variability and multiple view meta-modeling for product families. The Proposed model resolves all the four issues of variability management i.e. it provides variability mapping into product family by using enhanced product map and refined feature diagram. Extension from generic modeling is achieved through refined use case and multiple views of meta-modeling are represented with software architecture view. Discussed below is the evaluation of VMDE.

- *Variability identification* is not fully supported. Requirements variability in the form of features is present. But business values considerations and stakeholders' participation is absent.
- *Variability representation* is supported by different uml artefacts. Three activities are defined for domain engineering process. Scoping, domain analysis and product line architecture. First two deals with features/requirements variability identification and uses feature diagram and use case model and product map to represent this variability. For product line architecture, variability is represented by architecture view and architecture commonality and variability decision model. However, variability representation at component and source code level is absent.
- *Variability realization* support is half done. Requirements variability is realized at architecture level. Variability realization from architecture to component and source code level is missing.
- *Dependency management* support is fractional. Only support for feature interaction is present using the enhanced product map. However support for features tangling and scattering and components dependency is missing.
- Support for *product instantiation* is absent.
- *Traceability support* is fractional. Enhanced product map is the only used artifact that provides traceability between features and their corresponding products.
- *Value based traceability* is also absent.

2.5.2 Evaluation of CMVM: A Conceptual Model for Variability Management:

(Berg et al, 2005) identifies four requirements for a unified variability management approach which are consistency, scalability, traceability of variability across various generic development artifacts and variability visualization. The proposed process focuses specifically on the aspect of traceability and present a conceptual variability model, which captures variability information in a third dimension. Third dimension enables traceability of variations from problem to solution space in one to one fashion therefore facilitates in traceability of variability from problem to solution space. Researchers (Berg et al, 2005) have done a comparative analysis of proposed work with the feature model because feature model is commonly used for traceability of variability. The comparison concludes that the proposed model satisfies all the four requirements for a unified approach, whereas feature model is good for visualization and provides good views of variability in the form of a hierarchical tree like structure. But this adds complexity in case of traceability because feature model is another development artifacts and relationship between feature model and other elements also needs traceability. The conceptual model supports traceability as it captures variability information in a third dimension. Furthermore, it consistently captures variability information across all layers of abstraction for all generic artifacts. Researchers have highlighted that the proposed model does not fully support variability visualization. Detailed evaluation of the model on the basis of core issues is discussed below.

- Variability identification is not discussed. Although feature diagram is discussed but for the purpose of traceability of variability, not for variability identification. Similarly business values consideration and stakeholders' participation is also not present.
- Variability representation is not discussed explicitly. Variability in the form of variation points is captured and structured in a third dimension, which is the only information about variability representation.
- Variability is realized in the form of variation points, their dependencies and interrelationship. Such information is captured and structured in a third dimension.
- The model made no mention about product instantiation.
- Model provides support for dependency management. It provides one to one mapping between different artefacts of the product line both from problem space and solution space. This way it provides overview on relationship between these artefacts, thus supporting feature tangling and scattering, however feature interaction and components dependency is absent.
- Traceability of variability is the main focus of this model. It captures and structures the variation pints (VP) of each artifact in a third dimension and trace them to their appropriate dependent/ related variation point in the other artifact.
- Support for value based traceability is absent.

2.6 Evaluation of Traceability Based Variability management

This section presents the evaluation of Traceability based variability management solutions

2.6.1 Evaluation of TBKMS: A Traceability Based Knowledge Management System for Variability Management

Mohan & Ramesh, (2002) proposed a traceability based knowledge management system for e service families. They emphasize on traceability of design decisions rationale for product instantiation and configuration. Their model focuses on tracing commonality and variability in customer requirements to their corresponding design artifacts. They have discussed the importance of such traceability and concludes that tracing variability in customer requirements and then to the corresponding design artefacts facilitates in design process .In addition to this, capturing rationale for various design decisions facilitates at the time of making change. Similarly, traceability of architectural decisions to their sources and implementations helps in reducing customization costs. Evaluation of traceability based knowledge management system is discussed below.

- For variability Identification, requirements variability identification is supported by this system. It provides identification of conflicting customers' requirements along with their sources. Business values consideration and stakeholders' participation for variability identification are missing.
- Variability representation is not discussed.
- Support for variability realization is present. Forward traceability i.e. traceability from requirements to implementation facilitates in variability realization.
- Dependency management is also not fully supported by this system. Backward and forward traceability only provides an overview of interaction between different artifacts. Simple overview of interaction between requirements, and implementation artefacts supports only feature tangling and scattering but not the other two attributes of dependency management.

- It provides a mean of capturing platform oriented and architectural decisions which provides systematic reuse of this information at the time of product instantiation. This way it provides support for product instantiation.
- It focuses on traceability based knowledge management system; therefore traceability support is present in it. It provides traceability from requirements forward to implementation and from requirements back to the stakeholders. This way it provides forward and backward traceability.
- Value based traceability is missing.

2.6.2 Evaluation of TRM: A Traceability Reference Model

To support the product line development traceability reference model is discussed by (Jirapanthong & Zisman, 2005) which includes traceability between product line architecture and product member. Traceability reference model contains nine different types of relations for eight different types of documents. Traceability relations are classified in six different groups. These groups assist software development from different perspective. Paper contributes for variability management by simplifying the complex relationship between product line architecture and product members. Such simplicity provides easy access to identify the differences between product members and within individual products. In addition to it, traceability between product members and product line architecture facilitates in product instantiation. Detailed evaluation is discussed below.

- Variability identification is partially supported. Only Requirements variability is identified in the form of features, however, business values consideration and stakeholders' participation is missing. Requirements variability is identified with the help of "different" relation. This relation is used between documents of the same type for different product variants.
- Different UML artifacts are used for variability representation both at domain engineering level and application engineering level. Examples are feature model, use case, sequence diagram etc.
- Variability realization is not discussed.

- For dependency management, model provides relationship between different document types of product line level and product member level. Such relationship between these artefacts provides comprehensive overview thus making dependency management trouble -free. For feature tangling and scattering the model provides overlaps relation between feature and subsystem model. However other two attributes are absent.
- Product instantiation is well supported by the model. Relationship between product line architecture and product member facilitates in the identification of reusable assets and application specific variability.
- Traceability links between different artifacts provide well defined support of traceability. Model provides traceability reference model which defines ten traceability relations between eight different documents.
- Value based traceability is missing.

2.6.3 Evaluation of TM: A Traceability Map for Product Line Engineering

(Kim etal,2005) describe product line engineering traceability map and presents all the artifacts involved in product line engineering process and their relationships with each other. Research work identifies that variability identification and realization are important contributors in variability management and also presents traceability links for these two issues. Artifacts include core asset, decision resolution model, application analysis model, and instantiated core asset and commonality and variability specification. The relationship between these artifacts helps in identifying the application specific variability, and variability realization. Moreover, relationship between these artefacts facilitates in product instantiation. Evaluation is given below.

- Requirements variability in the form of features is identified in two different activities. First in domain analysis activity and secondly in core asset modeling. In case of domain analysis activity, variable features (requirements variability) are identified in Commonality and variability analysis (C&V). Whereas, Core asset modeling specifies features related variation points and variants. Technique made no mention about stakeholders' participation or business values consideration during variability identification.

- Variability representation is established with the help of traceability links between different artifacts.
- Variability realization is done in two stages. Firstly, Requirements variability is identified in C&V specifications in the form of variable features. These features are then realized in architectural and component specifications in the form of style, software components, hardware components and objects. Secondly, variability specified in C&V specification is realized in the form of variation points, variants, effects and attached tasks. The purpose of these two types of realization is to transform from more abstract elements in analysis phase to more concrete elements in design phase.
- Dependency management is absent.
- Product instantiation is supported by core asset model, application requirement analysis, gap analysis activity and resolve variability activity. Core asset model and application analysis activity assists in systematic reuse during product instantiation, whereas gap analysis activity and resolve variability activity identify and resolves issue of application specific variability. Traceability support is well established in the technique. Technique presents traceability map for all the activities of product line engineering and explicit traceability links for variability identification and variability realization.
- Value based traceability is not discussed in the proposed traceability map.

2.7 Evaluation of Variability Management Method

This section presents the evaluation of variability management approach.

2.7.1 Evaluation of VMM: A Method for Variability Management

Vangurp et al, (2000) presents a framework of concepts and terminology for variability and proposes a method based on this framework. The proposed method has four steps. 1) Identification of variability, 2) constraining variability, 3) implementing variability and 4) managing variability. Variability identification is the first and foremost step of variability management for which researchers suggest feature diagram. After variability identification it should be constrained to manage variability in a cost effective way. To do so, four steps are suggested in the method which are selection of binding time, decisions on adding new variants (when, how), picking a variability pattern and variability representation for the purpose of its realization. Similarly variability is implemented after selecting an appropriate variability realization mechanism and variation management is suggested as a final step to incorporate any variation static or dynamic. The proposed method aids in recognizing the need of variability early on and therefore it helps to design systems accordingly. Detailed evaluation is given below.

- Method provides incomplete support for variability identification. Variability identification is reported as first step of variability management but only requirements variability is supported by the method, other two factors i.e. business values consideration and stakeholders participation during variability identification are missing.
- For variability representation, feature diagram is used in the method.
- Variability realization is established in *constraining variability step* which enforce to select a representation for realization and provides various realization techniques.
- Support for dependency management is fractional because method only highlights feature interaction and made no mention about other issues of dependency management.

- Product instantiation, traceability support and value based traceability are missing in the method.

2.8 Summary of Evaluation of Variability Management Processes/Approach/Model/Knowledge Management System

The following section provides a summary of the evaluation of variability management processes based upon the core issues of variability management. Table 2.2 presents scale used for representing the support level for various factors of variability management on these processes. Figure 2.1 summarizes the results of evaluation.

Table 2.2: Scale for Support Level

Scale	Represented By
High Support	HS
Partial Support	PS
No Support	NS

Table2.3. Summary of Evaluation

VM literature / Parameters	Approach	Model Based Variability Management		Traceability Based Variability Management			Method	Process	
	Covamof	VMDE	CMVM	TRM	TBVM	TM	VMM	UML based VM	VBE
Variability Identification Business Values Consideration	NS	NS	NS	NS	NS	NS	NS	NS	NS
Requirement/features Variability	NS	HS	NS	HS	HS	HS	HS	HS	HS
Stakeholders 'participation	NS	NS	NS	NS	NS	NS	NS	NS	HS
Variability Representation	HS	PS	PS	HS	NS	PS	PS	HS	PS
Variability Realization	HS	PS	HS	NS	PS	HS	HS	HS	PS
Dependency Management Feature Tangling/scattering	PS	NS	PS	HS	HS	NS	NS	HS	NS
Feature Interaction	NS	HS	NS	NS	NS	NS	HS	NS	NS
Components Dependency	NS	NS	NS	NS	NS	NS	NS	NS	NS
Product Instantiation Application specific variability	NS	NS	NS	HS	NS	HS	NS	NS	PS
Systematic Reuse	HS	NS	NS	HS	HS	HS	NS	NS	HS
Traceability Support	HS	PS	HS	HS	HS	HS	NS	HS	PS
Value Based Traceability	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.9 Literature Analysis

The evaluation presented in table 2.3 evaluates prominent work of variability management based on the parameters discussed in section 2.2.As mentioned above, the body of knowledge, related to variability management processes is very thin. For this reason, we have not limited our evaluation only to processes but we have also evaluated the other important literature of variability management. Based on the evaluation it is evident that although, researchers have worked on different issues of variability management in their proposed solution but none of them has addressed all the core issues of variability management. E.g. in case of model based variability management, features identification, traceability support, variability realization and variability representation are addressed but dependency management, product instantiation , stakeholders participation and value based traceability are missing attributes. However, these are equivalently important for effective and efficient variability management (Theil & Heindel, 2002), (Sinnema, 2005), (Loesch & Ploedereder, 2007) and (Jirapanthong & Zisman, 2005).

Furthermore, the analysis also reveals that traceability support is the only attribute that has been addressed by almost all the researchers except Vangurp etal, (2000). This indicates the significant of traceability support for variability management.

Besides these, the area which needs attention is business values consideration, stakeholders' participation and value based traceability support. As discussed in section 2.2.1 product lines are large systems and different teams are in charge of different parts of the system and knowledge about variability is distributed among numerous heterogeneous stakeholders (Dhungana et al, 2006). By stakeholder we mean all people affecting or affected by the system. Different stakeholders have different perception about the variability information and therefore they must be involved in the variability identification process, for the purpose of extracting the right information (Rabiser etal, 2008).

Similarly business values consideration is important attribute of variability management. *Business values are the qualitative goals which the company strives to achieve in all its activities* (Roy Posner& MSS). The need of business values consideration during variability identification is already highlighted in (section 2.2).

Besides business values and stakeholders participation, value based traceability is also ignored by all of the proposed solutions of variability management. Value based traceability aims at prioritizing the traceability information. This implies that though traceability based variability management is viable for efficient variability management but traceability in a value neutral setting is not only difficult but also time consuming. Therefore it is important to consider this attribute while designing a solution for variability management.

From the evaluation of variability management literature, it is concluded that VBE proves to be the finest process for product line variability management. It involves stakeholders' participation in variability identification thus improving the quality of variability identification. Although for variability representation and realization, process does not use any formal notations but still discusses it in the form of variability and question cards. Documentation of variability information on different type of cards provides not only traceability of variability but also allows systematic reuse of this information at the time of product instantiation. However, the process has few shortcomings. In case of variability identification, business values consideration is missing. Furthermore dependency management is not discussed at all. Besides these, despite the importance of value based traceability, it is also absent. However all these are important attributes of variability management (for details see section 2.2).Therefore, it is of utmost importance to come up with an appropriate solution that addresses the issues and challenges that this processes faces so as to make it fruitful and effective.

After the evaluation and analysis on variability management literature, we conclude that for effective and efficient variability management we need to incorporate the following issues:

- Business values consideration for variability identification
- Stakeholders participation during variability identification
- Value based traceability.

In the next chapter, we will present our proposed process that will attempt to fill up the gaps which are identified in literature analysis.

Chapter 3

VVMT: Value Based Variability Management Using Traceability Information

3.1 Introduction

In previous chapter, we have evaluated and presented our literature analysis on prominent work of variability management. Based on the analysis it is evident that VBE proves to be the finest process but still it fails to address the important issues of variability management which includes business values consideration, traceability support (full), value based traceability support and dependency management (Rabiser et al, 2008),(Metzger& Pohl, 2006), (Batchman & Bass, 2001) and (Bosch et al, 2004). For this reason, instead of selecting VBE and modifying it to add these concepts, it was better to design a new process while incorporating all the core issues pertaining to variability management.

The proposed process aims at managing variability in a value based manner using traceability information. It uses the concepts of value based software engineering. It involves stakeholders in the process of variability identification and traceability prioritization. For variability identification the concept of stakeholders' participation is conceived from VBE (Rabiser et al, 2008). The process manages variability using traceability information- therefore it also provides traceability support. Besides these two major improvements, the process supports business values consideration for variability identification and also supports dependency management. This way, we have attempted to fill up the gaps identified during literature analysis. Besides these issues, the process also supports variability representation and realization.

3.2 VVMT Process Description

As stated in section (3.1), VVMT is designed to manage variability in a value based manner with the help of traceability information. Proposed process mainly consists of four main activities namely *variability identification*, *variability representation*, *variability realization*, and *monitor and control change*. In case of variability identification, stakeholders will identify the business values of the organization. The output of this activity will be core business values of product line organization. After variability identification, next activity is variability representation. In literature, various ways for it, have been discussed. For details see section (3.1.2). For our process, we will use tabular format. The table contains variability and different attributes/values associated to it. Next to variability representation, is its realization. For this purpose, we have proposed traceability approach. In this activity, detailed traceability information for variability identified previously is captured. Such information provides a comprehensive overview on variability and its various aspects. After variability realization, captured traceability information is prioritized. Finally monitor and control change activity monitors all of the above activities and manages change accordingly. Fig 3.1 shows the proposed process.

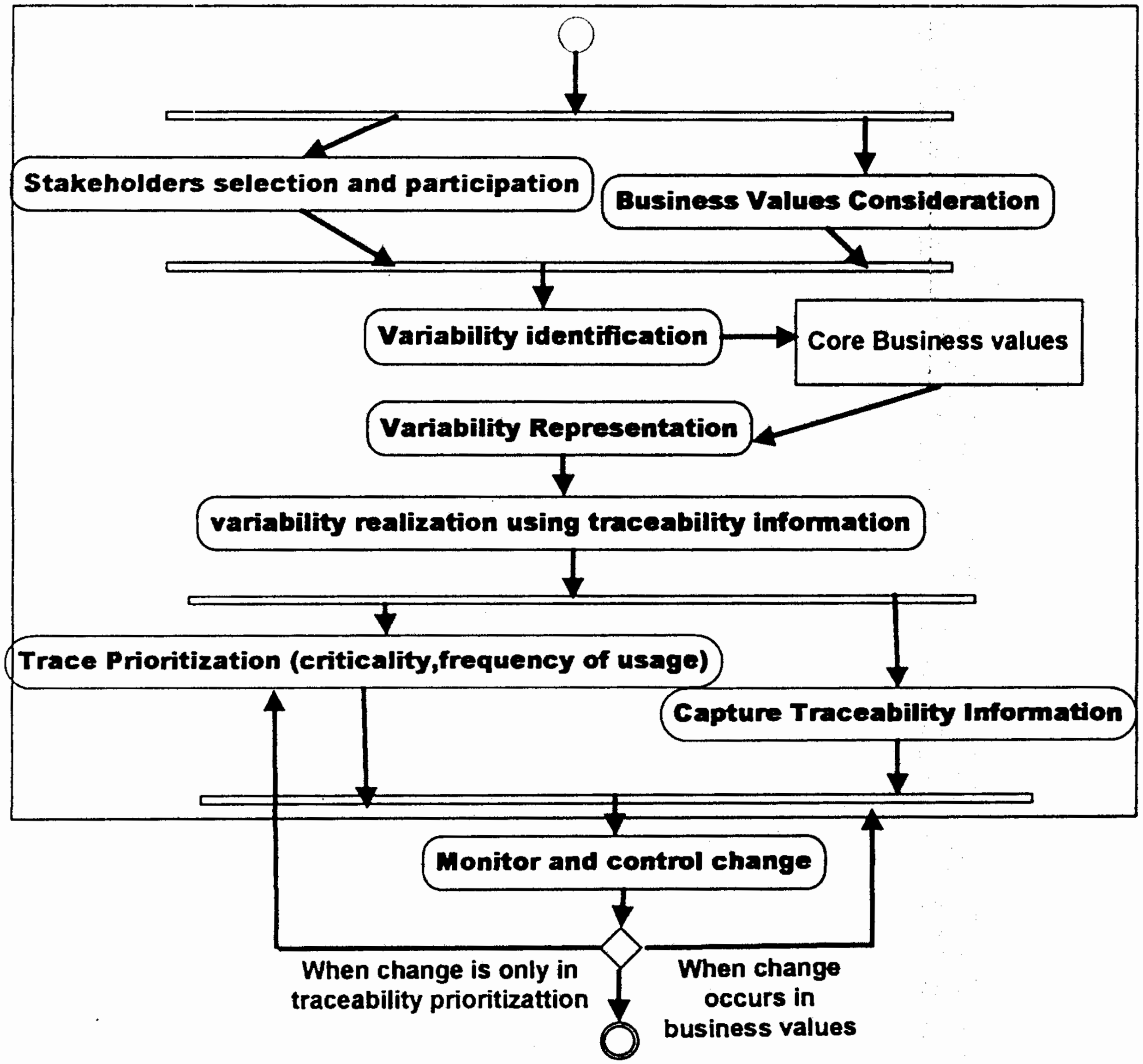

Figure 3.1 VVMT Process

3.2.1 Variability identification

This is the first activity of the proposed process. This activity includes stakeholders' participation for eliciting the core business values of the organization. We have adapted this activity from VBE (Rabiser et al, 2008). Stakeholders in case of variability identification are those who are well aware of the organization's business objectives.

They identify the core business values of organization. In case of product line development, business values usually refer to the aspects like cost reduction, time to the market reduction, or quality improvement (Schmid & John, 2001). However, every organization has its own set of business values (Ahmed & Capretz, 2007). Therefore the output of this activity will vary from organization to organization. After acquiring the core business values of the organization from different stakeholders, these values are then reconciled to achieve a mutually agreed set of business objectives.

3.2.2 Variability Representation

Next to variability identification is its representation. In this activity, important attributes/variants for variability (identified previous) are gathered. Such information is maintained in a tabular form. The table consists of two columns; one representing the name of the variability (business value in our case) and second column contains possible variants for variability. Variability representation in terms of variation points is already reported in literature (Berg et al, 2005). In our case, first column includes business values and second column contains variant attributes of each business value. E.g. improved product quality may includes fault free system, on time delivery, high performance etc. such information is helpful to understand what is meant by each business value for specific product variant.

3.2.3Variability Realization using traceability Information

Variability realization means understanding the impact of variations on different software assets so that variability can be implemented efficiently. For variability realization, we propose maintaining traceability information against each business value. We propose this on the basis of literature analysis which indicates that traceability information is an effective mean of impact analysis (Imtiaz, 2008).Traceability provides comprehensive overview of interaction between different parts of the system and therefore variability and its impact on different software assets can be understood easily (Ajjila etal, 2004).However we have proposed one more step for variability realization. I.e. value based traceability information which aims at separating the important set of traces from less important to manage variability in a value based manner. The need of value based traceability is discussed in (2.2.7). Consequently, value realization is accomplished in two steps. *a) Capture traceability information for each business values b) Traceability information prioritization.*

a) Capture traceability information for business values

The purpose of this step is to capture and map the business values and their related traceability information. Traceability information related to variability is captured. This information includes but not limited to activities, artifacts, actors (if any) and other information related to the business objective. In other words, this step helps to know how different business values are achieved by organizations, and what traceability information is required to achieve these values. In variability representation, only different attributes /variants are captured however in this activity, detailed information pertaining to each business value is captured.

b) Trace Prioritization

Based on the literature review (Jirapanthong & Zisman, 2005) and (Bayer & Widden, 2001), it is evident that product lines are complex in nature and traceability information associated to business values is large in number and complex in nature. Moreover, all traces are not equally important therefore value based tracing is required to reduce time and cost (Heindl & Biffl, 2005) and (Ahn & Chong, 2006). Therefore in this activity, stakeholders will prioritize traceability information on the basis of two parameters.1) *Criticality of traceability information, 2) frequency of information usage*. All traces are evaluated by using these two parameters on three point scale. High, medium and low as shown in table 3.1.

Table3.1: Scale for Trace Prioritization

		Criticality of information(c)		
		High	**Medium**	**Low**
Frequency of information usage (f)	**High**	Hfhc	Hfmc	Hflc
	Medium	Mfhc	Mfmc	Mflc
	Low	Lfhc	Lfmc	Lflc

This simple matrix prioritizes the traceability information for each business values. Traceability information will fall in one of nine possible categories as defined below

a. Information which is most critical and most frequent.(HcHf)

b. Information which is highly critical but not very frequently used.(HcMf)

c. Information which is most critical but its frequency of usage is low.(HcLf)

d. Information which is most frequently used but its significance is medium(McHf)

e. Information which is slightly critical and slightly used.(McMf)

f. Information which is least frequent but slightly critical.(McLf)

g. Information which is most frequently used but its significance is least.(LcHf)

h. Information which is slightly used but not critical.(LcMf)

i. Information which is neither critical nor used (LcLf).

The output of this activity separates the important traceability information from less important. This prioritization therefore helps to achieve core business values while saving time and cost.

As stated above, product lines encompass huge traceability information and to implement all of it is not feasible with respect to time and cost. Therefore traceability information is prioritized and only important traces are captured in detail. Less important traces are either discarded or implemented with lower level of details. Decision about threshold value depends on process user.

3.2.4 Monitor and Control Change

Business values *are the qualitative goals which the company strives to achieve in all its activities* (Roy Posner& MSS). It implies that they are not static and highly volatile with the time. Many factors both internal and external to the organization force them to change. Therefore, it is important to monitor this change so that it can be controlled in order to prevent any problem that may leads to drastic failure. As defined in figure (1), variability identification is related to all other activities; therefore any change in it will affect the other activities of the system. For this purpose, this activity is linked with other activities of the process. However, another arrow in the figure indicates the change in the priority of traceability information. In this case business value remains the same, but stakeholders' priority for the particular traceability information is changed. Therefore it is linked only with the trace prioritization activity.

3.3 Evaluation of the Proposed Process

This section evaluates the proposed process against the evaluation parameters discussed in section 2.2.

Starting from variability identification, we have divided it into three attributes; stakeholders' participation, business values consideration and requirements variability. The process has explicit support for stakeholders' participation and business values consideration. The process aims at identifying the core business values with the help of relevant stakeholders. For requirements identification, the process has implicit support for it. As stated in (Ajilla et al, 2004) and (McGregor, 2003), business objectives/values, define the requirements of an organization and their clients, therefore, identification of business values has implicit support for requirements identification. For variability representation, we have used tabular format. The table contains variability and its corresponding variants. Variability representation in terms of variants has already been discussed in literature by Berg et al, (2005). Variability realization is supported by traceability information. The process captures in depth traceability information for all the identified variabilities. This information facilitates in understanding the impact of variations on different software assets which is required for variability realization. For dependency management, we have not defined any explicit mechanisms but with the help of traceability information, relationship between variability and its associated variants can be understood which makes dependency management possible. Similarly for product instantiation, our process provides support for systematic reuse however, explicit mechanisms for application specific variability is not discussed. Systematic reuse requires necessary information about different parts of the system. Traceability links between different fragments of knowledge and provides this information to understand the different issues of reusability. Our process has traceability support. It captures and maintains traceability information which can be reused as desired. Moreover such information provides comprehensive overview of different fragments of knowledge thus improving understanding; essential for effective systematic reuse. Besides these issues, our process has explicit support for value based traceability. The traceability information captured during variability realization activity is prioritized by involving stakeholders. From the results of prioritization, important information is maintained and other information is either discarded or maintained at lower level of details.

The process has following limitations.

- The process does not support application specific variability.
- It does not provide explicit means of dependency management and requirements (functional) identification.
- Implementation mechanisms are not suggested in case of variability realization.

3.4 Further Discussion: Integration between VVMT and Product family engineering process

In this section, we will discuss the possible integration of VVMT with this product family engineering (Wikipedia, 2007). The purpose of this integration is to highlight how proposed process (VVMT) can play its part in the process of product family engineering. Product family engineering process (Wikipedia, 2007) has three phases as depicted in the figure given below; Product management, domain engineering and product engineering. Each of these phases has different set of activities for different purpose. Instead of selecting some traditional software development process, we have focused on product family engineering process due to its suitability to our context. Following are the possible ways in which VVMT and product family engineering process (Wikipedia, 2007) can communicate and support each other.

Evaluating business visioning (an activity of Product management phase) includes defining clear cut market strategy, collecting external market information such as consumer demands and context information relevant to product line scoping. In order to establish all this information, stakeholders' participation is required because they are the people who affect or get affected by the solution. *Stakeholders' participation* (sub activity of VVMT) can be helpful in this regard. Similarly, in order to capture external market requirements such as consumer demands and defining clear cut market strategy, organizations should have clear cut business values (objectives).This is because business values are the qualitative goals, which shapes the market strategy (Taulavuori et al,2004) &(Ahmed & Carpetz,2007). This shows that *business values consideration* and *stakeholders' participation* can assist in *evaluating business visioning*. Once business vision evaluation finishes, it may help in identifying variability.

The second possible integration between the two processes can be established in domain engineering phase, where *variability representation* (part of VVMT) can assist in defining the *domain design* (part of product family engineering process). The purpose of *domain design* is to generate an abstract structure for all products in a product line (reference architecture), for which comprehensive information about the system particularly variability and its associated variants should be available. Variability representation (part of VVMT) defines different variants for a single variability. It improves overall system understanding and gives a comprehensive overview of different choices for a single variability.

The third possible link between the two processes can be set up in product engineering phase. *Variability realization* (part of VVMT) captures and prioritizes detailed information regarding variability (business value). This information includes all aspects that are required to engineer a product effectively and efficiently. The purpose of variability realization is to have in depth information so that product having application specific features can be developed/ engineered effectively and efficiently while focusing on important and critical aspects. *Monitor and control change* (part of vvmt) may facilitate in monitoring and controlling any change in the product family engineering process.

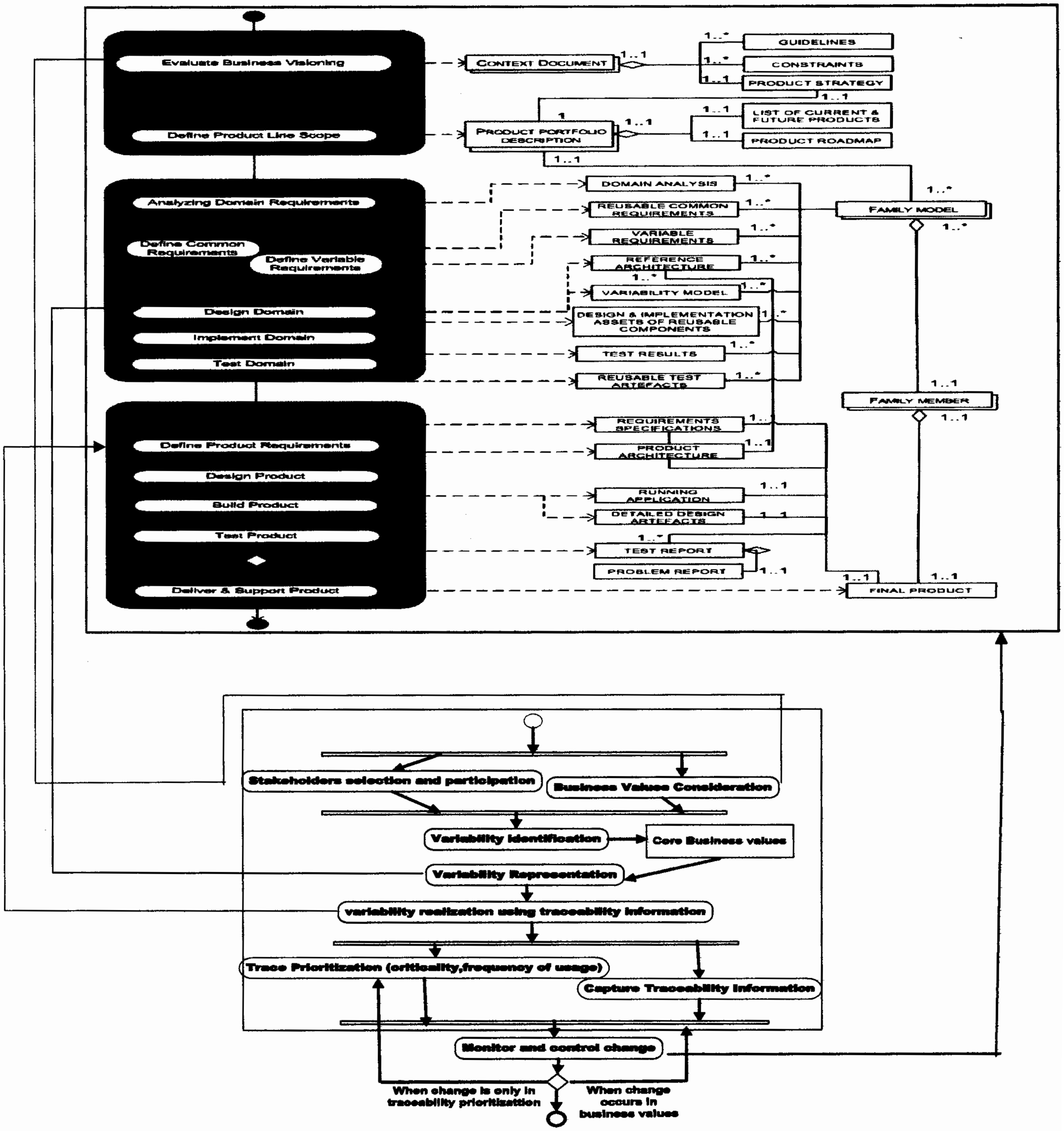

Figure 3.2 Integration between VVMT Process and Product family engineering process

Chapter 4

Validation of Proposed Process

4.1 Introduction

The proposed solution for product line variability management, while employing value based software engineering is validated through a case study. The following sections describes the selected case, methods of data collection, the variabilities (business values) encountered in the product line and discussion on how these variabilities could have been managed by application of the proposed solution.

4.2 Introduction of the case

The case study chosen is a product line for document management system, developed by one of the software development company that provides its services in the area of variable data publishing to print archival systems and also web based applications. The name of the company however is not mentioned here due to confidentiality.

The type and size of the product line was suitable for us to apply VVMT process in a software product line development. For our process validation, we required a product line (minimum of three products) in which variations occurs due to changing business values and traceability information related to business values and product line should be maintained. The selected case satisfied all these requirements. i.e. the product line under study had three products and it incorporated important concepts of product line engineering such as it used the core architecture of the product line. Product members of the product line had variations. The sources of these variations were changing business values of the organizations. Furthermore, traceability information related to the product line and its associated business values were maintained in different logs. The objective of case study is to determine the effectiveness of the proposed process for variability management by introducing value based variability management using traceability information.

4.3 Data Collection

Data for the case study was gathered from various sources to back up interpretations from multiple sources as opposed to considering a single source. This triangulation i.e. drawing the same conclusion from multiple sources of information, adds credibility to the results and the conclusions (Runeson & Host, 2008). The data sources employed for this case study included interviews, questionnaire and tool study.

4.3.1 Interviews

Interviews were used as a direct source for gathering the relevant information from stakeholders who were involved in this software product line development endeavor. The interviews were semi-structured and included both open and close ended questions. The questions were designed with the specific intent of extracting the core business values of organization, identification of attributes (variants) for these business values and for understanding the rationale of traceability information maintained in different logs. The major findings from the interviews were summarized to the relevant stakeholders in the end to gain a feedback and avoid any misunderstandings regarding interpretation of their point of views and comments.

4.3.2 Tool

A number of logs were also analyzed to study the patterns of information which is captured and maintained for accommodating different types of variations (business values). These logs are special type of databases and data in it is loaded from the main databases present at server side. These logs are part of company own designed tool. Although this tool performs many other functions but we had studied it only for the analysis of different logs. Three different logs namely call; problem and release were studied to capture the information (maintained) for accomplishing business values.

4.3.3 Questionnaire

Questionnaire was used another source for data collection. It was designed with a specific intent to study stakeholders' opinion, about importance of information, maintained in different logs. With the help of questionnaires, stakeholders were also asked to prioritize the traceability information that was maintained by the company for acquiring different business values. The questions were presented in the form of check list. Before presenting the questionnaire, the objective and meanings of the questionnaire was delivered in a short presentation. The results of the questionnaire were then analyzed to determine the difference between the information maintained and the information actually used by the company A for acquiring its business values.

4.4 Brief History of the Case

The product line under study had included three products referred as Product A, product B and Product C in the remaining document. Product A and Product B were document composition systems address the complexities of designing data-driven, high volume transaction output for print, the web, and email and the Product C was providing corporate users a web interface to generate documents on-the-fly. Initially this project was started in late 90's and then gradually functionality added time to time.

The company works on five main business objectives and all of its products (including the product line under study) are manifestation of these business values. The set includes technical support on products, enhancements in existing products, improved quality, on time delivery and competition to the market.

4.5 Implementation of the Proposed Process

This section describes the proposed process implementation on the selected case study.

4.5.1 Variability identification

A first activity for the proposed process is variability identification. The purpose/objective is to identify the core business values of the organizations. It is important to note that in our case variability is not treated as functional but variability means core business values of the organization. This activity is composed of two different sub-activities.

4.5.1.1 Stakeholders' Participation

There were total nine different types of stakeholders participating in identification of organization business values. Following is the list of them with brief introduction for each.

a. Customer: provides revenue and in return demands product/products satisfying objectives and needs.

b. Channel partners Channel partner are simple business/company partners usually for a co-branding relationship in distributors, vendors, systems/technology etc. Also they make the business strategies together for improving communication for shipping, schedules and other real-time information.

c. Sales: Group of marketing people who help to automate sales and plays very important role in the company's revenue generation. Also are called the front face of product line.

d. Engineering: Group of people responsible to develop new products, fixes bugs and enhances existing products.

e. Professional Services: Builds solutions on products on payment from customer generating revenue beyond software install.

f. Technical Support: Provides support to customers generating revenue and winning customer loyalty.

g. Product Management: Achieve market requirements/satisfies customers future needs. It includes product owner and team leader from the development.

h. Change control board: Change controls board is authority which allows making change in product after taking views of different stakeholders.

i. Configuration Management: configuration management is responsible to make/create installs/set ups of product.

4.5.1.2 Business Values Identification

The business values of the company A were elicited after the detailed interview with the top management. To reconfirm these business values a questionnaire is also given to the relevant stakeholders. From the results of both, it was observed that different stakeholders had different perception about the organization business values and sometimes their perception overlapped. It was also observed that stakeholder's participation for variability (business value/functional/non functional) identification is important and it facilitates in acquiring the correct knowledge about variability. Combining the opinion of all and after verifying it from the stakeholders, five business objectives were identified presented as below.

a. Technical Support on Existing Products

Besides developing new products, company also focuses on providing extreme technical support to its clients on its existing products. For this purpose, company has maintained a separate database which records and maintains different information regarding technical support. Technical support usually includes learning, or client may report some bug. Client request is analyzed and appropriate support is provided by the company. Company has maintained several support levels for each client request. During an interview with the company top management, we came across that technical support is an important business value as technical support helps in generating revenue and winning customer loyalty.

b. Enhancement in Existing Products

Enhancements in existing products include new features, patches update or any solution to the problem. Enhancement requests are maintained in an enhancement request log. Sales people communicate with customers and inform to the product managers about all the requirements. On the basis of customer's importance, requirements are fulfilled by the product manager and his team members. From the questionnaire given to the company development team, we came to know that client was not the only source for requesting enhancements. But sales and marketing people were also generating requirements after analyzing the market trends. Product manager further analyzes these requirements and after his approval development team works on these requirements.

c. Improved Product Quality

Product quality includes bug free systems, on time product's patch release, easy to use systems and provision of complete user guides and help. From the questionnaire and interview, it was evident that company strongly focuses on this business value as they do believe that customers' satisfaction is achievable only if they provide quality products to the company.

d. Time to market

Time to the market includes on time delivery of new products, on time delivery of existing products releases etc. In other words, Company A works on meeting the client's deadlines on new products and on existing products to generate the revenue. In case of existing products, client request can be bug fixing, enhancement in existing products in terms of functionality or quality or simple technical support. For new products company acquires client requirements, and then a detailed procedure is carried out to attain those requirements set. This includes requirements gathering, it's mapping on to design and later on its realization in terms of code. In order to achieve time to the market, company has maintained different records, details of which will be provided later in the case study.

e. Competition with the market/Competitors

In order to understand this business value, detailed interview along with questionnaire was conducted and the results reveal that company believes that competition with the competitors is crucial for the success. For this purpose company collects features lists of their competitors, and try to provide the same features with ease and at low cost. To do this, company provides best combination of performance parameters, usability parameters and maintenance parameters. Finally about cost, company and its business partners collectively decide after analyzing different aspects like effort in feature development, market value of that feature etc.

After value elicitation, next part of second step was to identify any conflict among stakeholders about these business values. It was interesting to note that there was no major conflict about business values. Some minor conflicts were about naming business values. The reason of this was the context in which these people are working. As an example, Company engineering and development staff call *market* however same value is named as *competition with the competitors* by the sales staff. However, inner details of both are same; therefore we have put these two as one value.

4.5.2. Variability Representation

Variability representation is a second activity. In this activity, we have represented variability (identified in 4.5.1) in the form of variants. Variants are possible set of values that can be used to fill in the identified variability. For this activity, we interviewed product managers of the product line. The questionnaires were designed with the intent of identifying the possible variants of each business value. By variant, we mean different possible variants that can be used to fill in the variability. E.g. in case of product B and product C required variant was learning for technical support business value whereas for product A required variant was correction. The results of interview were then presented to them in order to identify any discrepancy. Variability and its associated variants are placed in a tabular format as shown below.

Table 4.1: Variability Representation

Variability	Variants
Technical Support	Correction , learning on new item,
Enhancements in Existing Products	patch update, addition of new feature, bug fixing
Improved product Quality	Bug free systems, Reusability Ease of use portability
Time to the market	On time delivery of new feature On time bug fixing On time patch release On time delivery of new product(s).
Competition with the Competitors	Cost Performance Usability Maintenance

Variability representation in the form of variants improves understanding regarding variability. E.g. for product C *competition with the competitors'* business value included performance and usability as variants, however for product B required variant was cost. This implies that variability representation in terms of variants helps to identify and understand the client requests and it also provides different views on variability; consequently improving the understanding and facilitating in managing variability. In case of company A, explicit representation of variability in the form of variants was not employed and usually it is extracted from the customer requirements that was time consuming and sometimes ineffective according to the opinions of the top management.

4.5.3 Value Realization using traceability Information:

Once business values were identified and represented, next step was to realize these business values by using traceability information. The purpose of this step was to extract information which was used to acquire these business values. We call this information as traces/traceability information related to these business values. In other words, this step helped to know how different business values were achieved by organizations, and what traceability information was required to achieve these values. To accomplish this step, we had studied different logs of the company own designed tool. Logs were Call log, problem log and release plan log. These logs are kind of databases that were used to maintain traceability information. After capturing the business values related information, we interviewed from the team leaders of testing, professional services, and development departments. The reason for taking interview from them is that they keep and acquire different information from these logs and they had better information about the rationale for traceability information kept in these logs. To reconfirm the results a questionnaire was also given to the team members of these departments and the results of both were combined.

4.5.3.1 Capture Traceability information for business values

For this activity, we have studied different logs and collected the data related to the core business values of an organization. After this, we had conducted an interview with the team leaders of testing, professional services and development departments to extract the rationale for traceability information.

4.5.3.2 Trace Prioritization

In this step all the traces (collected in mapping step) were prioritized. The purpose of prioritization was to determine the set of traces which are important for acquiring business values so that we can separate important traces from less important; in order to achieve business values with minimum amount of information.

For this activity questionnaire was given to the company top management, in which they were asked the criticality/importance and frequency of usage against traceability information for each business value. For this activity, we had selected top management instead of team leaders of different departments (testing, professional services and development) because they decide about the traceability information to be kept in logs based on the importance of this information for the products under development. Three point scales (high (1-3), medium (4-6), low (7-9)) was used in questionnaire. The combined results of the activities 4.5.3.1 and 4.5.3.2 are given below.

a. Technical Support on Existing Products

Table 4.2 shows traceability information and its prioritization for *Technical Support existing products* business value.

Table 4.2: Technical Support realization using traceability information

S. No	Traceability Information	Description	Criticality of Information(C)			Frequency of Usage(F)			C*F	Required Log
			High 1	Medium 2	Low 3	High 1	Medium 2	Low 3		
1	Call		•			•			1	Call log
2	Customer Deadline		•			•			1	Call log
3	Status		•				•		2	Call log
4	Fixed in Release	release In which issue is actually fixed	•				•		2	Call log
5	Release due	release in which issue should be fixed	•			•			1	Call log
6	Closed during initial call	if issue is resolved early			•			•	9	Call log

7	Source	employee/client		•				•	6	Call log
8	Contact	name of the source	•					•	3	Call log
9	title	job designation		•				•	6	Call log
10	phone			•			•		4	Call log
11	fax		•				•		2	Call log
13	Company			•			•		4	Call log
14	address			•			•		4	Call log
15	City			•				•	6	Call log
16	Country			•			•		4	Call log
17	Customer Priority	requirements are fulfilled on the basis of this parameter	•			•			1	Call log
18	Product	name of the product	•			•			1	Call log
19	Product type	category		•			•		4	Call log
20	Severity	Criticality of an issue. Value 1 to 5	•			•			1	Call log
21	Location	client location		•				•	6	Call log
22	FAQ required	queries regarding technical support		•				•	6	Call log
23	Support level	three values		•			•		4	Call log

b. Enhancements in Existing Products

Table 4.3 shows traceability information and its prioritization for *enhancements in existing products* business value.

Table 4.3: Enhancements in existing products realization using traceability information

S. No	Traceability Information	Description	Criticality of Information(C)			Frequency of Usage(F)			C*F	Required Log
			High 1	Medium 2	Low 3	High 1	Medium 2	Low 3		
1	Product	name of the product	•			•			1	Problem Log
2	Customer Deadline	time given by the customer	•			•			1	Call Log
3	Component	enhancement related component	•			•			1	Problem Log
4	Fixed in Release		•			•			1	Call Log
5	Release due	release number in which enhancement is made	•			•			1	Call Log
6	Closed during initial call			•				•	6	Call Log
7	Source	client/employee		•			•		4	Call Log
8	Contact	name of the source	•			•			1	Call Log
9	title	designation(employee)		•			•		4	Call Log
10	phone		•				•		2	Call Log
11	fax			•				•	6	Call Log
12	Email		•			•			1	Call Log
13	Company			•			•		4	Call Log
14	address			•			•		4	Call Log
15	City			•			•		4	Call Log
16	Country			•			•		4	Call log
17	Customer Priority	as in a	•			•			1	Call Log

18	Package	enhancement related package		•				•	6	Call Log
19	Product type	Category of the product		•				•	6	Call Log
20	Severity	as in a	•			•			1	Call Log
21	Location			•			•		4	Call Log
22	FAQ required	as in a			•		•		9	Call Log
23	Support level	as in a			•			•	9	Call Log
24	Function	enhancement related function(s)		•			•		4	Problem Log
25	Affected Module		•			•			1	Problem Log
26	Problem discussion		•			•			1	Problem Log

c. On time Delivery

Table 4.4 shows traceability information and its prioritization for *On time delivery* business value.

Table 4.4: On time Delivery realization using traceability information

S. No	Traceability Information	Description	Criticality of Information(C)			Frequency of Usage(F)			C*F	Required Log
			High 1	Medium 2	Low 3	High 1	Medium 2	Low 3		
1	Product	name of the product	•			•			1	Release Plan
2	Package	time given by the customer		•			•		4	Release Plan
3	program manger		•			•			1	Release Plan
4	product manager		•			•			1	Release Plan
5	version	release number in which enhancement is made	•			•			1	Release Plan
6	release type(patch,		•			•			1	Release

	product)									Plan
7	build		•			•			1	Release Plan
8	code stream		•			•			1	Release Plan
9	Current release	designation(emplo yee)		•		•			2	Release Plan
10	Release stakeholders		•			•			1	Release Plan
11	release plan		•			•			1	Release Plan
12	product phase			•			•		4	Release Plan
13	due date		•			•			1	Release Plan
14	Current date			•			•		4	Release Plan
15	complete details		•			•			1	Release Plan
16	date started		•			•			1	Release Plan
17	Customer Priority		•			•			1	Release Plan
18	Customer		•			•			1	Release Plan
19	Source			•			•		4	Call Log
20	Contact		•					•	3	Call Log
21	title			•				•	6	Call Log
22	phone			•			•		4	Call Log
23	fax				•			•	9	Call Log
24	Email		•				•		2	Call Log
25	Company			•				•	6	Call Log
26	address			•			•		4	Call Log
27	City			•				•	6	Call Log

28	Country			•			•		4	Call Log

d. Improved Product Quality

Table 4.5 shows traceability information and its prioritization for *Improved Product Quality* business value.

Table 4.5: Improved Product Quality realization using traceability information

S no	Traceability Information	Description	Criticality of Information(C)			Frequency of Usage(F)			C*F	Required Log
			High 1	Medium 2	Low 3	High 1	Medium 2	Low 3		
1	Product	as above	•			•			1	Call Log
2	Product Type	as above		•			•		4	Call Log
3	Release number	as above		•			•		4	Call Log
4	Release due date	as above		•			•		4	Call Log
5	Quality type	what form of quality is required	•			•			1	Problem Log
6	source	(employee) responsible for improving product quality	•			•			1	Call Log
7	Source of request	(Client, company internal staff usually sales and marketing).	•			•			1	Call Log
8	Customer deadline		•			•			1	Call Log
9	Components	Components that affects or affected by achieving quality		•		•			2	Problem Log
10	Functions	Func quality.	•						1	Problem Log

e. Competition with Competitors/market

Table 4.6 shows traceability information and its prioritization for *Competition with competitors/market* business value.

Table 4.6: Competition with competitors/market realization using traceability information

S. No	Traceability Information	Description	Criticality of Information(C)			Frequency of Usage(F)			C*F	Required Log
			High 1	Medium 2	Low 3	High 1	Medium 2	Low 3		
1	Product	as above				•			1	Call Log
2	Product Type	as above					•		4	Call Log
3	feature list	feature sets of the same type of products in order to compare the functionality	•			•			1	Problem Log
4	usability	to compare the products on this factor	•			•			1	Problem Log
5	performance	to compare the products on this factor	•				•		2	Problem Log
6	cost		•			•			1	Problem Log
7	Competitors – if any			•			•		4	Problem Log
8	Any Documents / Specs		•			•			1	Problem Log
9	Fixed in Release		•				•		2	Problem Log
10	Any Bench Marks			•			•		4	Problem Log
12	Man efforts		•			•			1	Problem

										Log
13	Stakeholders		•			•			I	Problem Log

When top management was asked about the threshold value, they reported that they will consider "most important" category and traces falling in others two categories will be discarded, because for company A out of 99 traces 51 were actually important and others were not significant and therefore they agreed to discard. This implies that after the introduction of proposed process same business values were achievable by maintaining 51 traceability links however before the process introduction they were 99. Summary of the results are as below.

Table 4.7: Summary of Trace Prioritization

Summary of the traceability information	**Value**
Total number of traces	99
Traces fall in "most important" category	51
Traces fall in "not very important/ ignorable" category	48

As indicated in table 4.7, out of 99 traces, 51 were actually required; however company spent its resources on 99 different types of traces. This indicates that wastage of time, cost and effort in maintaining the 48 unimportant traces. In our case, we have focused only on the time dimension and showed that with the introduction of VVMT, same business values can be achieved effectively and efficiently. For this purpose, we have collected the following information from interviews and tool study.

- Total number of fields in each log(n)
- Time (in minutes) required to process (create, open, edit) individual field of each log (ti). Time required for each field differs minutely therefore, we assume that time required for processing each individual information is same.
- Total amount of time (in minutes) required to process (usually includes opening and editing of fields) a complete log(T).

- Correction factor. This is overall delay of the system which may occur due to network, data access time or multiuser processing etc. The related information for each log is given in table below.

Table 4.8: Time requirements against each log

Log name	Total number of fields n	Time(minutes) required for individual field ti	Time (minutes) required for a complete log. T= n*ti*c, c=1.1
Call log	23	0.6	15.2 = 15
Problem log	30	0.66	21.8 = 22
Release plan	18	0.66	13.06 = 13

Starting from business value"technical support on products", total number of traceability information were 23 and 12 of them were either not frequently used or not critical for achieving the business values. all the traceability information for technical values were related to the call log. Therefore they decided to discard them. Before introducing VVMT, total time for a call log was 15 minutes however 7 minutes (no of unimportant traceability information) were spent on the information which was not required. The formula used for calculating was

(*Time required for each field)* (no of un- important traceability information)*. (0.6*12) = 7 mins app

Similarly for "enhancements in existing products" 14 out of 26 traceability links were unimportant, 13 of them were related to call log and one of them was maintained in problem log. For accurate results, time for both were calculated separately and then added to get the combined result. (13*.6+1*.66)=7.8+.66=8.46 or 8.5mins app.

Similarly we calculated the results for all the business vales which are given in the table below.

Table 4.9 Time required/utilized for maintaining traceability information before and after VVMT

Business value	Before VVMT, time spent on each business value	After VVMT ,time spent each business value	Time improvements/saving
Technical support	15minutes	8 minutes	7 minutes
Enhancements in existing products	16 minutes	8 minutes	8 minutes
On time delivery	19minutes	10.5	8.5 minutes app
Improved products quality	6 minutes	4 minutes	2 minutes app
Competition with the market	8.5 minutes	6.2 minutes	2 minutes app

4.5.4 Monitor and Control Change

This activity is performed to monitor and control changes in business values, and trace prioritization. During this activity, business values were same; however slight difference in the traces prioritization was noticed.

4.6 Discussion and Analysis

Traceability information for various business values, pertaining to software product line development, was identified in this case study. Based on the questionnaire and reinforcement from the tool, it was observed that company had maintained repeatable information for its business values e.g. product name, customer information, product release number etc. it was also observed that some traceability information that were neither important nor frequently used was also maintained. Keeping redundant and unimportant information created performance degradation, wastage of time, cost and effort. E.g. with the presence of redundant and unimportant information, approximately 15 minutes were required for opening a call log to work on it, whereas by applying the value based variability management, opening a call log required 8 minutes. Besides this issue, database is placed at server side and numerous employees accessed it only daily basis. Therefore, in the presence of unimportant or redundant information load on network was increased which created performance degradation. Moreover, against individual field/traceability information, there were different options available. So the user may spend more time on viewing and selecting the option. Again in case of unimportant information, it was time consuming but not useful. All of these pitfalls were due to the fact that Company A was keeping all the traceability information without prioritization for its variability management.

The variabilities, as discussed in the previous sections, were the core business values of the company. By application of the proposed solution, variability was appropriately managed. Stakeholders' participation at the time of variability identification assists in identifying the correct core business values because they are involved in product development either directly or indirectly, therefore their participation ensures collection of right information, and correct identification of variability paves the way towards effective variability management (Raibser et al, 2008).The representation of these variabilities in terms of variants provides better overview of available possibilities of each variability. Importance of stakeholders participation and variability representation in terms of variants is also reported by (Raibser, 2008), (Berg etal,2005) Variability realization in terms of traceability information helps to capture the in depth information about each variability and then prioritization of this information facilitates in managing the variability effectively and efficiently. We conclude it on the basis of results as shown in (table 4.9). It shows that out of 99 different traces, 51 were the most important that were actually required by the company A and rest of the traces were not important and company A was agreed to discard this information (threshold value for company A was 1-3). This implies that before the introduction of proposed process ,company A was achieving its business values while maintaining 99 different traceability information ,however after the introduction of the proposed process, same business values were achievable with 51 different traceability information...the results revealed that due to prioritization time saving was approximately 56%, because before VVMT, time spent in maintaining all traceability information was 1hrs ad 4 minutes approximately, but after the introduction of VVMT, 36 minutes were required for maintaining traceability information. This indicates that with the introduction of VVMT variability was managed effectively and efficiently.

4.7 Summary of the Result

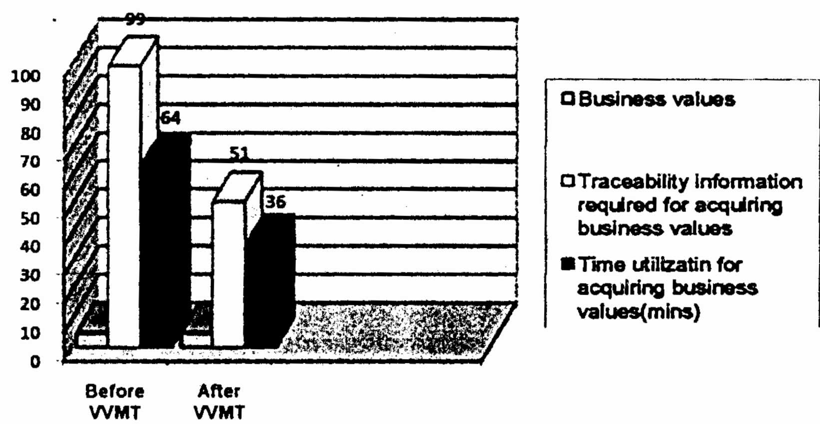

Figure 4.1: Traceability and time utilization before and after VVMT

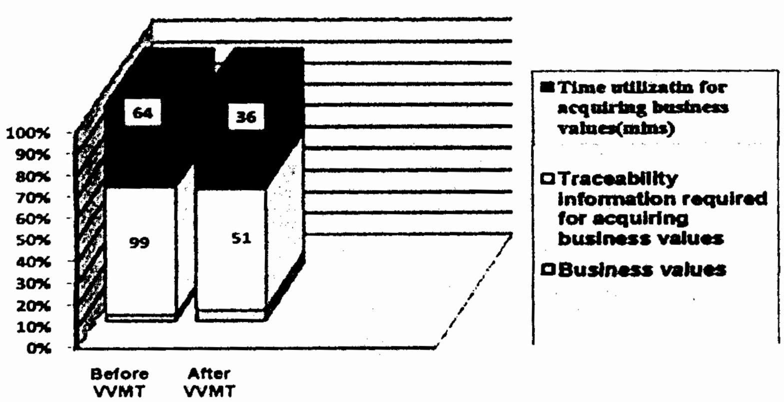

Figure 4.2: Traceability and time utilization before and after VVMT

Chapter 5

Conclusions and Future Work

5.1Contribution of the research

Variability management is central to the success of software product line engineering. It aims at managing the unavoidable changes requested either from internal or external to the organization, mostly due to changing business values of an organization. Various solutions are employed for the purpose of variability management. These solutions however, face different challenges when applied for effective and efficient variability management.

The major contribution of this thesis includes a proposed process for tackling with all major issues of variability management. We have also identified the core issues of variability management for evaluating individual variability management solutions and evaluated variability management processes on basis of these issues, to determine the process that would fare best in a product line variability management environment.

The core issues of variability management for evaluating variability management processes has been identified by reviewing the literature pertaining to variability management activities, issues and problems in managing product line variability and other factors that enable and influence variability management in a software product line development environment. Identification of this set of core issues was necessitated to evaluate the variability management solution and to discover their strengths and weaknesses relative to variability management. Up till now, no evaluation of these solutions exists to determine how they fare in the variability management. Therefore, this contribution assists researchers and practitioners in determining the effectiveness of the available solutions in the context of variability management.

The evaluation of variability management solutions based on identified core issues attempted to offer an insight into how variability management is challenged by various issues. Realization of these challenges assists in identifying effective variability management mechanisms, which can have a direct bearing on the success of the product line development. Evaluation of the variability management solutions shows that they are influenced to varying degrees by the issues identified. From the evaluation, VBE proves to be the fair choice. It comprises of stakeholders participation for variability identification and also provides partial support for variability representation, variability realization, product instantiation and traceability support.

A process has been proposed in this thesis that aims at managing variability in a value based manner using traceability information. It uses the concepts of value based software engineering. For variability identification, it includes stakeholders. The concept of stakeholders' participation for variability (business values) identification is conceived from VBE (Rabiser et al, 2008). The process manages variability using traceability information- therefore it also provides traceability support. Besides these two major improvements, the process supports business values consideration for variability identification and also supports dependency management. This way, we have attempted to fill up the gaps identified during literature analysis. Besides these issues, the process also supports variability representation and realization. This way it incorporates all the important issues pertaining to variability management.

Three research questions were posed in the first chapter of this thesis i.e. what are the core issues of variability management, determining how effective variability management processes are with respect to these issues and what possible measures can be taken to make variability management process effective. The first research question is answered by identifying core issues after conducting an in depth literature survey pertaining to variability management. The second research question was answered by evaluation of the variability management processes and other solutions based on the evaluation parameters. The third research question posed is answered by the proposed process, which facilitates in managing variability effectively and efficiently by introducing concepts of value based software engineering for variability management.

5.2 Recommendations and Future Work

For a more general evaluation of VVMT and to evaluate the effort difference by applying VVMT, we plan multiple case studies with the systematic range of other domains for software product line development. We will try to generalize the business values and their associated traceability links, in order to establish a general frame of reference for product line traceability with respect to business values of relevant stakeholders. Automation approaches for value based variability management using traceability information is also a future topic. Another topic of future will focus on improvement of trace prioritization in order to optimize the value of VVMT. There are many more relevant trace prioritization attributes than criticality and frequency of usage that can be used in the context of VVMT .e.g. trace volatility, cost of trace implementation, interdependency between different traces that belong to conflicting business values, etc.

Value based variability management using traceability information (VVMT) is important to keep track of interdependencies between stakeholders 'values(in terms of business objectives) and product line artifacts. This information helps product line development team in several tasks e.g. cost benefit analysis, conflicts identification and understanding the mapping between system elements and business values etc. VVMT is a promising approach to lighten the problem of high effort of managing variability in a comprehensive way.

References

Ahmed, F., and L. F. Capretz (2007). "Managing the business of software product line: An empirical investigation of key business factors". Information and Software Technology Journal. **49** (2), Elsevier.

Ahn, S., and K. Chong. (2006). A Feature-Oriented Requirements Tracing Method: A Study of Cost-benefit Analysis. In International Conference on Hybrid Information Technology, Korea, IEEE.

Ajila, Samuel A., and Ali B. Kaba (2004). Using traceability mechanisms to support software product line evolution. In IEEE International Conference on Information Reuse and Integration, Las Vegas, Nevada, USA, IEEE.

Atkinson, C., et al. (2002). "Component-Based Product Line Engineering with UML". Addison-Wesley.

Bachmann F., and L. Bass (2001). Managing variability in software architectures. In ACM Symposium on Software Reusability, Toronto, Ontario, Canada, ACM.

Bayer, J., and T. Widden (2001). Introducing Traceability to Product Lines. In 4th International Workshop on Product Family Engineering. Spain. Lecture Notes on Computer Science, volume, 2290, Springer.

Becker, M. (2003). Towards a General Model of Variability in Product Families. In 1st Workshop on Software Variability Management, Groningen, Netherlands.

Berg, K. et al. (2005). Tracing software product line variability: From problem to solution space. In Annual Research Conference of the South African institute of Computer Scientists and information Technologists on IT Research in Developing Countries, White river, South Africa, SAICT.

Bosch, J. (1999). Product-Line Architectures in Industry: A Case Study. In 20th International Conference on Software Engineering, San Francisco Bay, USA, IEEE.

Bosch, J. (2000). "Design & Use of Software Architectures: Adopting and evolving a product-line approach". Addison-Wesley.

Bosch, J. et al. (2004). Variability issues in software product lines. In Revised Papers from the 4th International Workshop on Software Product Family Engineering, Spain. Lecture Notes on Computer Science, volume, 2290, Springer.

Buhne, S., et al. (2005). Modeling requirements variability across product lines. In 13th IEEE International Conference on Requirements Engineering, Germany, IEEE.

Cravens,D.W., G.Greenry., N. F. piercy.,and Slater ,S.(1997), Integrating contemporary strategic management perspectives. In long range planning, **20**(4),unknown.

Czarnecki, K., and U. W. Eisenecker (2000). "Generative Programming: Methods, Tools, and Applications". Addison-Wesley.

Deelstra, S., Sinnema, M. and J. Bosch (2009). "Variability assessment in software product families". Information and Software Technology Journal. **51** (1). Elsevier.

Dhungana, D. et al. (2006). Coordinating Multi-Team Variability Modeling in Product Line Engineering. In 2nd International Workshop on Supporting Knowledge Collaboration in Software Development (KCSD2006), collocated with the 21st IEEE International Conference on Automated Software Engineering (ASE'06), Tokyo, Japan, IEEE.

Estublier, J., and G. Vega (2005). Reuse and variability in large software applications. In 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT

International Symposium on Foundations of Software Engineering, Lisbon, Portugal, ACM.

Gomaa, H., and D. L. Webber (2004). Modeling Adaptive and Evolvable Software Product Lines Using the Variation Point Model. In 37th Annual Hawaii International Conference on System Sciences, Hilton Waikoloa, IEEE.

Gurp, J. Van. et al. (2001). On the Notion of Variability in Software Product Lines. In Working IEEE/IFIP Conference on Software Architecture (WICSA'01), Washington D. C., USA. IEEE.

Heindl, M., and S. Biffl (2005). A case study on value-based requirements tracing. In 10th European software engineering conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Lisbon, Portugal, ACM.

Imtiaz, S. (2008). "Impact analysis from multiple perspectives using traceability information". Thesis, Mohammad Ali Jinnah University, Islamabad.

Inoki, M., and Y. Fukazawa (2007). Software product line evolution method based on kaizen approach. In ACM Symposium on Applied Computing, Seoul, Korea, ACM.

Jaring, M., and J. Bosch (2002). Representing variability in software product lines: A case study. In Second Product Line Conference, San Diego, California. Lecture Notes in Computer Science, Vol. 2379. Springer:

Jirapanthong, W., and A. Zisman (2005). Supporting Product Line Development through Traceability. In 12th Asia-Pacific Software engineering Conference, Taiwan.

Kang, K. C., et al. (1998). "FORM: A feature-oriented reuse method with domain-specific reference architectures". Annals of Software Engineering Journal. Volume *5*.

Kim, S. D. et al. (2005). Traceability Map: Foundations to Automate for Product Line Engineering". In third ACIS International Conference on Software Engineering Research, Management and Applications, USA, IEEE.

Khalifa, S.A. (2004) 'Customer Value: A Review of Recent Literature and an Integrative Configuration' Management Decision, **42** (5): unknown

Lee, J., and D. Muthig (2006). "Feature-oriented variability management in product line engineering". Communications of the ACM. **49** (12). New York, USA.

Loesch, F., and E. Ploedereder (2007). Optimization of Variability in Software Product Lines. In 11th International Software Product Line Conference Washington D. C., IEEE.

Markus, A., and J. Vancza (1998). "Product Line development with customer interaction". CIRP Annals Journal. **47**(1). Elsevier.

McGregor, J. (2003). "The Evolution of Product Line Assets". CMU/SEI Technical Report No. 005, ADA418409, Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA. Retrieved from http://www.sei.cmu.edu/publications/documents/03.reports/03tr005.html.

Metzger, A., and K. Pohl (2006). Variability Management in Software Product Line Engineering. In 27th International Conference on Software Engineering, Shanghai, China, ACM.

Mohan, K., and B. Ramesh (2002). Managing Variability with Traceability in Product and Service Families. In 35th Annual Hawaii International Conference on System Sciences, USA, IEEE.

Mohan, K., and B. Ramesh (2003). Ontology-based support for variability management In 36th Hawaii International Conference on System Sciences, USA, IEEE.

Mohan, K., and B. Ramesh (2007). "Tracing variations in software product families". Communications of the ACM. 50 (12) New York, ACM.

Oliveira, Dc. et al. (2005). A variability management process for software product lines. In Conference of the Centre For Advanced Studies on Collaborative Research, Toronto, Ontario, Canada IBM Press.

Posner, Roy, and MSS., (2009). "Business: Utilizing Business values". Retrieved June, 2009, from http://www.gurusoftware.com/GuruNet/KnowledgeBase/Business/Values.htm.

Rabiser, R. et al. (2008). Value-Based Elicitation of Product Line Variability: An Experience Report. In Second International Workshop on Variability Modeling of Software-intensive Systems, Germany, ICB-Research Report No. 22, University Duisburg Essen.

Riebisch, M., and I. Philippow (2001). Evolution of Product Lines Using Traceability. In OOPSLA Workshop on Engineering Complex Object-Oriented Systems for Evolution, Florida.

Runeson, P., and M. Host (2008). "Guidelines for conducting and reporting case study research in Software Engineering". Empirical Software Engineering Journal. 14 (2), Springer.

Samad, J. et al. (2008). VRRM: Value based risk management process. In International Conference on Software Engineering, Austria.

Schmid, K., and I. John (2001). Product line development as a rational, strategic decision. In International Workshop on Product Line Engineering in Early Steps: Planning, Modeling, and Managing. Klaus Schmid and Birgit Geppert editors. Retrieved from www.plees.info/Plees01/plees01-dokumente/06-john.pdf.

Sinnema, M. et al. (2004). COVAMOF: A Framework for Modeling Variability in Software Product Families. In Third Software Product Line Conference, Boston, MA, USA, Lecture Notes on Computer Science, Vol. 3154. Springer.

Staples, M., (2004). Change control for product line software engineering. In 11th Asia Pacific Software Engineering Conference, Busan, Korea, IEEE.

Svahnberg, M., and J. Bosch (2000). Issue concerning variability in software product lines. In Third International Workshop on Software Architectures for Product Families. London, UK. Springer.

Taulavuori, A. et al. (2004). "Component documentation-A key issue in software product lines". Information and Software Technology Journal. **46** (8).

Theil, S., and A. Heindel (2002). Modeling and Using Product Line Variability in Automotive Systems. IEEE Software Special Issue on Software Product Lines. **19** (4), IEEE.

Weiler, T. (2003). Modeling Architectural Variability for Software Product Lines. In Software Variability Management ICSE 2003 Workshop, Groningen, Netherlands.

Wikepedia(2009).Product Family Engineering Retrieved October, 2009, from http://en.wikipedia.org/wiki/Product_Family_Engineering