Configuration Challenges & Issues: A Configuration Process Model to Reduce the Complexity of Software Product Line

Uzma Afzal
Department of Computer Science, FUUAST, Gulshan Campus Karachi.

Prof. Dr. Syed Irfan Hyder
Department of Computer Science, PAF-KIET Karachi.

Mahwish Rani
The Educators, Garden Campus, Karachi.

Abstract

Configuration of software product line is often a more laborious process than anticipated before. One major difficulty with software product line engineering is unavailability of standardize configuration process model that reduces the adverse affects of configuration related issues and problems such as requirements conflicts of same/multiple stages, complexity of conflicts resolution, adhoc testing and low integration with traditional phases of software engineering process models. We discuss some core product line configuration issues and present a standardize configuration process model "CPM" that is built on the strong base of traditional software engineering process models with the additional power of modern product line configuration to reduce the configuration issues and problems. Some real world scenarios are used to illustrate the proposed model.

1. Introduction

Software vendors want to fulfill all user requirements. In an ideal environment, they satisfy almost all requirements and deploy system in time. However actual environments are quite different. Tight budgets and schedules imply some serious requirement's issues and challenges [16].One of the most critical issues is unvoiced customers belief i.e. they wasted their value able time in specifying requirements for a useless system. One authoritative and widely cited compilation of software quality issues is the 1994 "CHAOS Report". It reported that only 16% projects were succeeded.

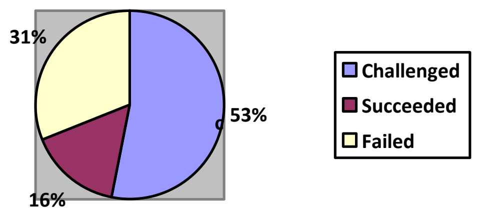


Figure 1: Project Resolution [13]

In 2004 update report showed some improvement but overall situation remains dire.

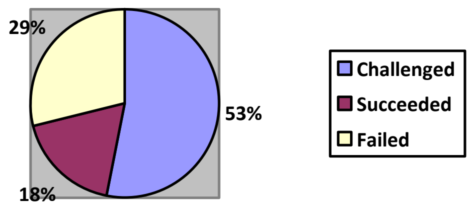


Figure 2: Project Resolution [14]

These problems changed the customers perspective and they switched to 'buy & configure' a software from a reputed vendor rather than to start from scratch (Analysis, design, code and so on) and brought the generic & flexible products development in demand to fulfill the need of multiple customers having similar requirements domain.

A product line is a group of products related to each other by marketing, technical or end-use considerations. Adopting a product line approach allows companies to build different systems with a minimum of technical diversity and to realize significant improvements in time-to-market, cost, productivity and quality [11]. Deriving a product from a product line means selecting and configuring existing reusable assets to fulfill a specific customer's needs. In ideal scenario all customer requirements can be satisfied by exploiting existing assets and their variability. The more realistic scenario, however, is that the customers can have additional wishes and requirements [2].

The configuration of software product line provides benefits to both stakeholders (software customer and software vendors) but introduces some new problems and challenges.

2. Configuration: Issues & Challenges

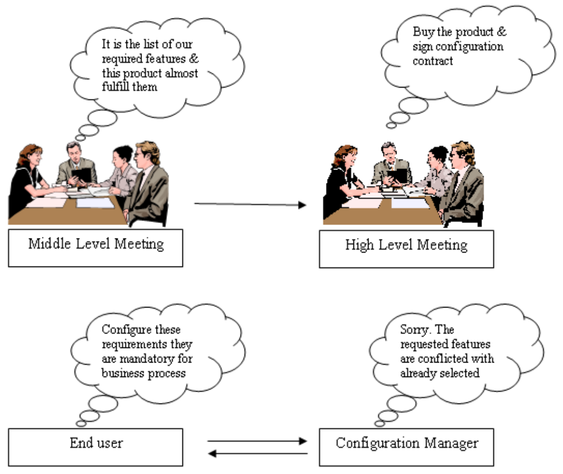
Configuration management (CM) is more important than ever because customers want new designs of products of higher quality at lower prices. Efficient CM can shorten the product life cycle, minimize production cost, and guarantee product quality [18]. Product configuration has proven to be an effective means to implement mass customization [17]. Through a configuration process, product modules or components are selected and assembled according to customer requirements [17].

Product Configuration is a collaborative process. Multiple stakeholders involve in process and different configuration units assign to them. It creates problem when each individual takes his own configuration decisions (for e.g. feature selection) without going in extra detail. Assembling of these asynchronous efforts is one of the biggest configuration issue [10]. There are many techniques available to reduce complexity but they still dependent on features understanding and knowledge of developers [19].

Software product vendors encounter problems when attempting to configure software products. As the competitiveness become more violent, they have to deliver the quality product to gain customers trust. Unfortunately lack of standardize software product configuration process places a big question mark on product quality.

Some issues and challenges of product configuration process are discussed below: -

Requirements are gathered in multiple stages. This is illustrated in figure 3. User involvement is very minimal in first stage. Management defines basic requirements. The 2nd stage takes place after procurement decision. Requirements are constrained by the facilities available in the purchased system [4]. As the configuration moves forward actual user requirements discover (third stage). Requirements that are gathered in multiple stages can be conflicted. Multiple stages of Requirements gathering and involvement of different participant in each level make the conflict resolution complex. Requirements gathering can not be limited to a single stage because it is impractical.


It is the list of our required features & this product almost fulfill them
Buy the product & sign configuration contract
Middle Level Meeting
High Level Meeting
Configure these requirements they are mandatory for business process
Sorry. The requested features are conflicted with already selected
End user
Configuration Manager
Figure 3: multiple stages of requirements

Multiple stages requirements gathering is not the only cause of conflicts. Conflicts are also observed in requirements of the same stage and it becomes worse if configuration is doing for a big scale software product by a large team of developers with their assign responsibilities. Normally, these conflicts are captured in testing phase. However, debugging of these systems is itself a complex and time-consuming task.

Figure 4 shows some hardware/software conflicts. When configuring automotive systems, software developers may want a series of software component configurations that cannot be supported by the configurations proposed by the hardware developers. To each party, their individual needs are critical and finding the middle ground to integrate the two is hard [5].

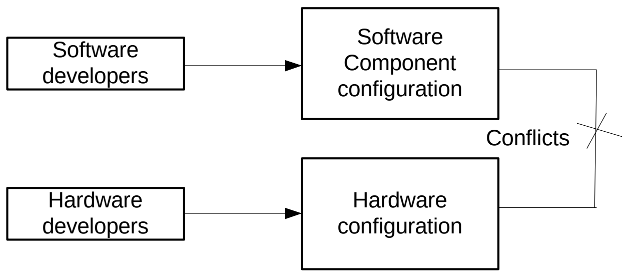


Figure 4: multiple participants involvement

Testing of these systems is another issue and a less standardize requirements gathering process of product configuration directly effects the testing phase. Lack of requirements detail makes system testing difficult [4].

In addition, configuration processes do not support the conventional software engineering processes such as quality assurance and quality management, However they are the most critical and important areas of software engineering and basic building blocks of the software product quality.

Moreover, Software product configuration processes do not support traditional phases of software development like requirements engineering. To derive a product from product line, configuration of existing reusable assets are done to meet a particular customer needs, however sometimes customers have some additional wishes and requirements that are not supported by existing repository of Product line, in this scenario only configuration engineering is not well enough, facility of requirement engineering is also required to satisfy the customer needs [2].

3. Motivation

A scenario is created for the illustration of existing software product line configuration process limitations. " A Software product family name is "PF" that contains multiple components from C_1 to C_n and constraints from Co_1 to Co_n. Two derivate products P1 & P2 configure from PF."

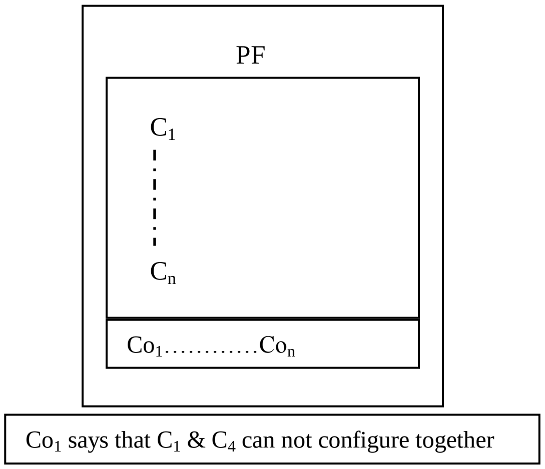


Figure 5: Product Family Repository

Figure 6 describes the configuration scenario with additional requirements. Configuration of P1 starts first. Components C1 and C2 configure respectively. When configuring component C3, customer gives some additional requirements that are not supported by current product repository and these requirements are mandatory for customer business process. Traditional configuration management processes focus on selection and assembling of components from existing repository and do not support the traditional phases of software engineering like requirement engineering so every configuration manger deals this situation in his own way without going in detail that weak/substandard/undefined requirement gathering process can introduce critical problems in later phases and the complexity of configuration will be increased. Lack of a well defined new requirement development phase places a big question mark on product quality, user satisfaction and the ability of software process to entertain additional user wishes. There is a great chance of user's unacceptability of product and no support of the traditional phases (like requirements engineering) is a main cause.

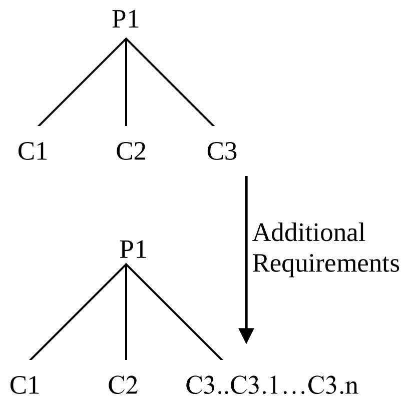


Figure 6: Configuration with additional requirements

Figure 07 illustrates another common scenario. Second derivative of product family configures with the components C1, C3, C4 &C6. Testing of product reveals the fact that P2 has violated one of the constraints Co_1 i.e. C_1 and C_4 can not configure together.

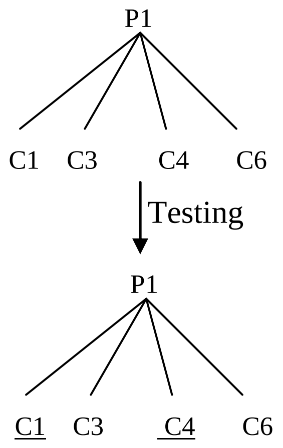


Figure 7: constraint violation of components

Late discovery of the conflicts violation introduces complexity in configuration process and moves P2 in inconsistent state. Transformation of the flawed featured model into constraint satisfying model and conflicts resolution are complex and time consuming processes.

4. Proposed Configuration Process Model

Software Quality Engineering includes three very basic steps [12]:

- Define a Software Engineering process.
- Assure adherence to the process.

- Improve the process.

We have modified these steps for software configuration process.

- Define a Software product configuration process.
- Assure adherence to the process.
- Continuous improve process.

We present a well-defined and quality centric product configuration process to reduce the adverse affects of product configuration process. It is divided into ten phases and three entry points. Each phase has an entry and exit criteria. Phases of the process model are:

(1) Business Requirement gathering [BRG]. Business requirements are gathered here to clarify the business objective. High level management of both side (customer and vendor) are main participants. System selection and scope definition is done. BRG just makes sure that selected system is appropriate to satisfy business objectives. If yes Contract signs and system procures. Actual configuration process initializes here.

(2) Initial Investigation [II]. Initial investigation is conducted in this phase. End user representatives and configuration mangers are main participants. The main objective of II is investigating system up to the level that is required for platform design. Main feature selection, environmental requirements and other interfacing requirements provide base for platform design.

(3) Platform Design [PFD]. Figure 8 shows a representation of platform based configuration. Platform is a set of common components, modules or parts from which a stream of derivative product can be effectively created and launched [15]. PF is the base of different conflicts resolutions. In term of customer's requirements [II], Platform selects first, and then different products derive by selecting different components [9].

In this phase a new platform can also be designed or existing PF from repository can be selected or updated. If any conflicts identify here and no suitable Platform is available, process moves back to initial investigation.

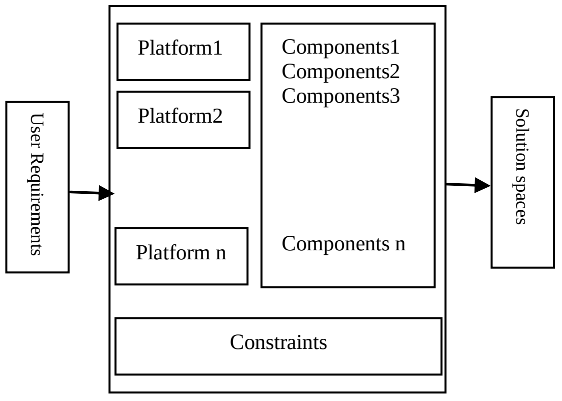


Figure 8: Platform based Configuration [9]

(4) Platform Validation. PF validates to ensure that a consistent PF achieves without any conflicts and problems .PF also validates to check the compatibility with environment and other interfacing system on the basis of initial investigation.

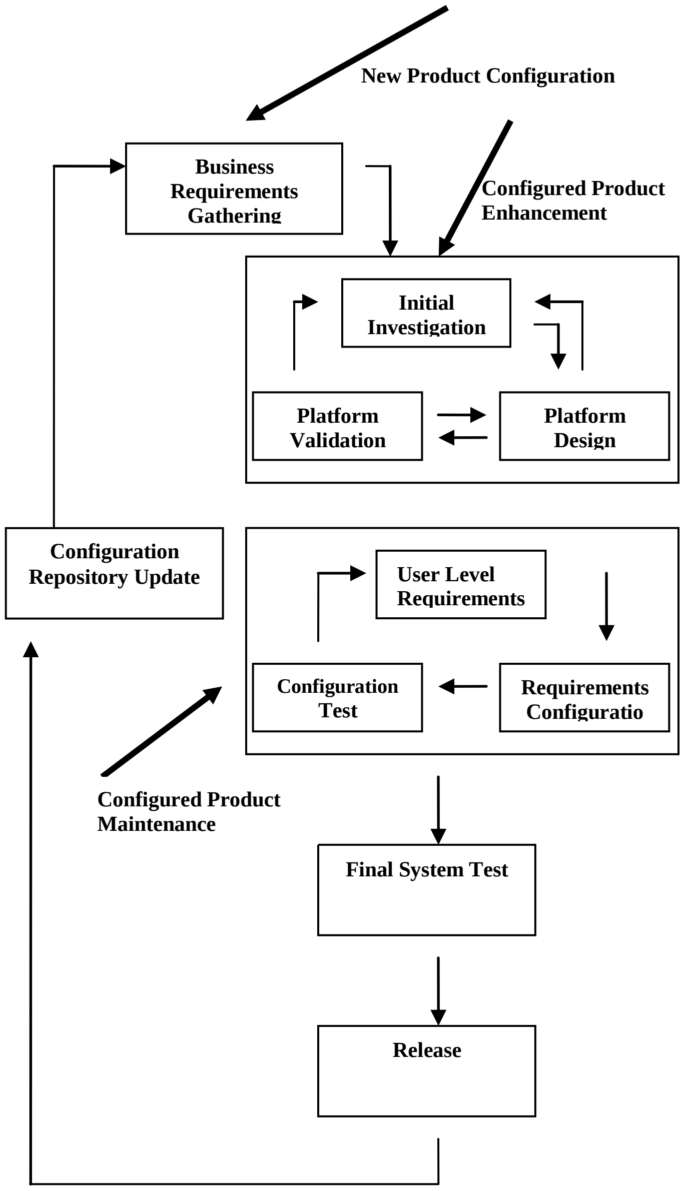

New Product Configuration
Business Requirements Gathering
Configured Product Enhancement
Initial Investigation
Platform Validation
Platform Design
Configuration Repository Update
User Level Requirements
Configuration Test
Requirements Configuratio
Configured Product Maintenance
Final System Test
Release

Figure 9: Proposed process model for the configuration of the software product line

(5) User Level Requirements. Detail requirements elicit from users. Both types of requirements (Requirements that can be directly satisfied by only product configuration & Requirements that require further development) entertain here. All captured requirements document and only those are validated that can be satisfied using selected platform. Test cases are also generated here.

(6) Requirement Configuration. All the requirements that are established in user level requirement phase are configured.

(7) Configuration Test. Black box configuration test will be done here. Configuration tests provide a base for quality product.
(8) Final System testing. Additional integration and system testing are done to test the configured system. User acceptance testing will also be a part of final system testing.
(9) Release. The tested configured system releases.
(10) Configuration Repository update. After the successful deployment of the system existing repository that contains different platforms and features is updated.

Three entry points of the process are:-
(1) New Product Configuration. Project enters through this entry point if configures first time.
(2) Configured product maintenance. It is the entry point for the maintenance of already configured software product.
(3) Configured Product enhancement. Entry point for the enhancement of product that is already configured.

5. Evaluation

We have evaluated proposed process model by using some real world scenarios. We revisited the motivational scenarios and used 'CPM' as a process model. Configuration of P1 starts first. Components C1 and C2 configure respectively. When component C3 configuring customer gives some additional requirements that are mandatory for customer business process.

Using proposed model, process moves into user level requirement phase to fulfill them then actual configuration takes place, configuration tests make sure that requirements are properly implemented without any conflicts and problems. Proposed CPM is highly flexible and provides a good support to traditional phases of the software development life cycle.

In second scenario Product P2 configure with the components C1 & C3 when component C4 configuring, it generates error message because selection of platform prohibits the configuration of conflicted components into single variant moreover validation of plat form removes any remaining ambiguity.

Early discoveries of the problems make the process simple and save a lot of time and efforts.

6. Capabilities & Benefits of CPM

In this section we discuss that how configuration process model helps to reduce the complexity of the configuration process and reduce the effects of issues discussed in section 2.

CPM supports the traditional software engineering processes such as requirement engineering. Sometimes customers have some additional wishes and requirements that are not supported by Product line, in this scenario only configuration engineering is not well enough, facility of requirement engineering is also required to satisfy the customer needs [2]. Requirement engineering process can be easily mapped to CPM through user level requirement phase.

A series of testing [configuration & system] makes sure that configured product is tested up to an adequate extent. Configuration errors capture earlier in configuration testing and we don't have to wait for the system test. We have opportunity to resolve these problems earlier to avoid the configuration complexity due to the late errors identification.

Introduction of the platform design phase reduces the requirements conflicts of the different participants and different stages. Platform is a set of common components and no conflicted requirements can be configured under a single platform.

Base of Platform design is initial investigation phase and all users requirements are configured using the selected platform and requirements that are out of the scope of plat from not configured so there is no chance of multiple user requirements gathering errors. The only way to configure these requirements is platform updation. Moreover from business requirement gathering to initial investigation and initial investigation to user level requirement there is a linked established that makes sure that all user level requirements are supported by initial investigation requirements and all initial investigation requirement supported business objectives so there will be no **inconsistent state** in the product configuration cycle.

Business objectives → Initial investigation → User level requirements

Involvement of users throughout the process, platform design to avoid requirements conflicts, platform validation, configuration testing to discover errors earlier and a final system test to completely check the system validity, make sure that CPM is quality centric.

7. Future work & Open Issues

We have validated the model by using some real world but self created scenario, however it requires some real world projects implementations for widely acceptance in software market. In future we will collect results of real world implementation and study its impact in organizations. We also do research on CPM and other models comparison.

8. Conclusion

In this paper core issues related to the configuration of software product line are highlighted and a motivational example is used to show that how configuration process complexity increased due to the less integration with conventional process models, requirements conflicts and adhoc testing. A configuration process model has been proposed to resolve these issues and real world scenarios are revisited to illustrate the working of the proposed configuration process model.

9. Acknowledgement

Uzma Afzal would like to thank to her mother. There is no doubt in her mind that without her mother's support and counsel she could not have completed it.

10. References

[1] Goetz Botterweck, Steffen Thiel, Ciarán Cawley, D Nestor1, André Preußner. Visual Configuration in Automotive Software Product Lines. Annual IEEE International Computer Software and Applications Conference 2008.

[2] Rick Rabiser, Deepak Dhungana. Integrated Support for Pr-duct Configuration and Requirements Engineering in Product Derivation. 33rd EUROMICRO Conference on Software Engineering and Advanced Applications 2007.

[3] Yaluo Yang,Ming Li,Yayu Huang. The use of configuration conception in software development. IEEE Pacific Asia workshop on computational intelligence and industrial application 2008.

[4] Ian Sommerville. Construction by Configuration: Challen-ges for Software Engineering Research and Practice. 19th Australian Conference on Software Engineering 2008.

[5] J.White, D.C. Schmidt ,D. Benavides , P. Trinidad , A. Ruiz–Cortés. Automated Diagnosis of Productline Configuration Errors in Feature Models. 12th International Software Product Line Conference 2008.

[6] Philip Miller. An SEI Process Improvement Path to Software Quality. Sixth International Conference on the Quality of Information and Communications Technology 2007.

[7] Cheng Thao, Ethan V. Munson, Tien N. Nguyen. Software Configuration Management for Product Derivation in Software Product Families. 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems 2008.

[8] Per Runeson, Peter Isacsson. Software Quality Assurance – Concepts and Misconceptions. Proceedings of the 24th. EUROMICRO Conference 1998.

[9] Yan huiqiang. PB-GPCT. International conference on computer engineering and technology 2009.

[10] Liguo Yu1 and Srini Ramaswamy. A Configuration Mana-gement Model for Software Product Line.

[11] K. Pohl, G. Böckle, and F. v. d. Linden. Software Product Line Engineering Foundations, Principles and Techniques. 1st ed. New York, Springer 2005.

[12] Humphrey, Watts S. Managing the Software Process. Addison-Wesley, 1989

[13] The Standish Group. The C H A O S Report 1994. The Standish Group International 1995.

[14] The Standish Group. The C H A O S Report 2004. The Standish Group International 2004.

[15] Meyer, M. H. & Lehnerd, A. P. The power of product platforms. The Free Press. 2000. New York.

[16] Paul Clements and Linda Northrop. Software Product Lines: Practices and Patterns. Addison-Wesley 2001.

[17] Chunjing Zhou, Zhihang Lin & Chuntao Liu.Customer-driven product configuration optimization for assemble-to-order manufacturing enterprises. Int J Adv Manuf Technol 2008, 38:185–194

[18] I. Choi and S. Bae: An Architecture for Active Product Configuration Management in Industrial Virtual Enterpris, Int J Adv Manuf Technol 2001, 18:133–139

[19] Maryam Shiri, Jameleddine Hassine, Juergen Rilling. Feature Interaction Analysis: A Maintenance Perspective. ASE'07, Atlanta, Georgia, USA.