TOOLS SUPPORT IN SOFTWARE PL ENGINEERING

[1]Anuranjan Misra [2]Arjun Kumar [3] Akanksha Bana [4] S.K.Pathak

[1]Professor, CSE, Department, Bhagwant Institute of Technology, Ghaziabad, India
[2] Asso.Professor, ABES Engineering College, Ghaziabad, India
[3]Assistant Professors, CSE, Department, Bhagwant Institute of Technology, Ghaziabad, India
[4] Asso.Professor, ABES Engineering College, Ghaziabad, India

ABSTRACT

Tool support has become a key success factor in software product Line Engineering due to vast increase of complexity in today's software intensive system. The objectives of my paper are to develop a quality model to provide a framework for evaluating quality of SPL tools and secondly to validate this proposed quality model on two product line tools by conducting an experimental study which will be useful for researchers, academicians, professionals and vendor organizations to learn, improve or upgrade the functionality of the tools.

INTRODUCTION

Software Product Lines (SPL) has emerged as a significant approach for efficiently developing various software products.SPL is the most promising and leading trend to successfully build different products as it offers high productivity, decreased market time, substantial product cost savings and massive quality improvement. Effective tool support is compulsory in PLE. Tools need to be flexible and adaptable in order to support particular requirements and needs of engineers in different development contexts. Moreover, Product line practice in the software industry is a not too much old concept and since last some years, it has taken great momentum as almost all world's leading organizations.

PROBLEM STATEMENT

Software Product Line Engineering is a methodology for producing multiple products of the same family by sharing common core assets which includes documentations, codes, test plans and architectures and other artifacts. As Product Line Engineering (PLE) comprises of many heterogeneous activities, Software has become more complex in terms of its inherited features. A huge number of CASE (Computer Aided Software Engineering) tools have been produced to assists tasks in a software development process since the end of 1970's.

METHODOLOGY

This study consists of two phases in order to achieve the objectives mentioned earlier. The first phase is to formulate the quality model for evaluating the quality of SPL tools. The Second phase is to evaluate the quality of two products line tools in order to validate the newly proposed model.

PRODUCT LINE QUALITY MODEL

To evaluate any software product, it is pivotal to select some relevant quality characteristics by using any quality model that is closer to evaluate such type of products. Although a number of research studies have been conducted on the quality of different software systems that were developed from traditional software development methodologies, but despite of all that, literature on how to measure quality of product line products and tools that behave as the major role for products line development have not been considered by researchers.

EVALUATING PRODUCT LINE TOOLS

In the scope of this phase, experimental study was performed to evaluate in detail the two tools of product line on the basis of their non-functional factors. Quantitative analysis of the tools was performed on the basis of four major quality requirements i.e. Usability, performance, memory consumption and reliability, and comparison will be performed between two particular product line tools. All these quality requirements were tested with different strategies as their nature is different.

1 Selection of Tools

For evaluating tools of Software Product Line, the very major task was the selection of the tools. For the accomplishment of this task, the first major preference was to download such tools that were commercially used a lot. Second preference was given to such tools that were available free for download.

2 Installation of Tools

Installing the tools was a very challenging experience. These tools were free to download, but their installation process was really ambiguous, especially of Feature IDE. However, tools were downloaded from their official websites. Installation guides found from their official websites really proved helpful in installing these tools.

3 Usability testing of PL tools

The purpose of usability evaluation is to identify the problems with the user interfaces like the tools ability to be easier to use, learnable and understandable when the tools are used in a specified environment.

3.1 Evaluation designing

For usability testing evaluation, evaluators were categorized in 3 groups i.e. advanced and new users.

- **Non-expert Users**: The users had no experience in product line engineering.
- **Expert Users:** Users who had some domain knowledge of software product line engineering.
- **Author:** Author who acted having no domain knowledge.

Sr.	Evaluators	No. of users
1.	Expert system	2
2.	Non expert users	2
3.	Author	1

3.2 Test environment

Author’s site was chosen as test environment for usability testing. In order to get best results, controlled environment set as both tools were downloaded on separately machines.

3.3 Test conduction

According to the tasks lists the users had to interact with the tools and performed the tasks that were given in the list. Before the experiment, author gave brief description of the tasks and provides some guidelines for performing the task.

4 Reliability testing of PL tools

Reliability was the second quality factor to evaluate on feature IDE and Varmod. Here the discussion is about the reliability evaluation on the product line tools; it contains the information about the assessment methods for reliability.

4.1 Evaluation designing

The testing of reliability scenario was different. For the reliability evaluation, strategy was to rum the software for thousands of hour in real scenarios and then observed its behavior. To measure the reliability of software product line tools, author did run the tools in segment of hours for the period of 40 days. Chunk of maximum 5 hours were used for one day evaluation.

4.2 Test environment

The author sites were chosen for reliability testing. There were two important reasons for selecting author’s site as a testing environment for reliability evaluation. First reason was that only the author acted as evaluator for observing behavior of both product line tools on each day and second reason was the load shedding limitation.

4.3 Test conduction

Author observed the tools on each day and noted the running time of tools and the number of crashes found in excel sheet. After a certain time period, the total number of hours running and the identified faults for each tool separately was calculated to find the mean failure of both tools.

5 Performance testing of tools

It means to test software under expected workload to ensure that it will perform well in such conditions. There can be different dimensions of measuring the performance testing of the tools such as response time, scalability, loading time and many more feature and functionalities are supported by the software system.

5.1 Evaluation designing

For testing of tool’s performance capability, author set some asks to be performed on both tools. Each task had varying level of difficulty. The author acted as observer to check the performance of tools.

5.2 Test Environment

The author site’s is chosen as author only participated in testing performance of PL tools as only running time had to calculate which is to be done by author.

5.3 Test Conduction

To test the reliability of Feature IDE and Varmod tasks were divided into two levels: low and medium level.

6 Memory Consumption testing of PL tools

The last quality factor to evaluate on Feature IDE and varmod tools is ‘memory consumption’. How much memory each tool consume in performing the projects? The objective of this evaluation is to find out which tool between both tools takes more memory in execution environment by selecting no. of features as testing criteria to check the memory consumed by each tool.

6.1 Evaluation designing

For memory testing, firstly checked the official websites of both SPL tools to get information about the resource utilization and memory consumption, installation packages, Supporting applications and plug-ins, additional required platforms, system resources and operating systems.

6.2 Test conduction

The goal of this evaluation is to check that whether system is able to fulfill requirements of size or not in a specific workload.

RESULTS

In this section, the obtained results of the quality factors have been analyzed and discussed possible reasons behind the results which differentiate or resemble the results of the tools.

1 Usability

Usability testing results were collected by after observing the tool’s behavior upon performing of tasks by evaluators.

2 Reliability

Reliability testing results were collected after observing the tool’s behavior upon performing different tasks on each day for different durations.

3 Performance

Performance testing results were collected after observing the tool's behavior on the basis of response time upon performing different tasks.

4 Memory Consumption

Firstly, specifications for the both SPL tools to be installed on any machine were explored by the author. Then, to observe the memory consumed by both tools in modeling diagrams were mentioned.

CONCLUSION

This study proposes a model for evaluating the quality of tools and the products of software product lines. There are two contributions that are provided by this work including proposing the quality model for product line tools and products evaluation. Some quality characteristics were chosen from ISO model as it is the base of our newly proposed model. Secondly, offering the comparison between software product line tools to suggest which tools suite is better for particular situations.

REFERENCES

[1] Bosch, J. (2002). Maturity and evolution in software product lines: Approaches, artefacts and organization. In Software Product Lines, pp: 257-271.

[2] Clements, P. and L. Northrop. (2002). Software product lines: Practices and patterns, Third Edition, Addison-Wesley, Boston, USA, pp: 26-34

[3] Chen, L., A.Babar and N.Ali. (2009). Variability management in software product lines: a systematic review. In Proceedings of 13th International Software Product Line Conference, pp: 81-90

[4] Kowalski, G. J. and M.T. Maybury. (2000) .Information storage and retrieval system: theory and implementation (Vol.no8)

[5]Shahid, M. and M.Qaiser. (2010).Software Product Line: Survey of tools. M.S. Thesis, Deptt Computer Science, Linkoping University, Sweden

[6] Zhang, J., X. Cai and G. Liu (2008).The role of aspects in software product lines. In computer Science and Information Technology, pp:588-592.