Fraunhofer Institut
Experimentelles
Software Engineering

First eWorkshop on Quality Assurance for Software Product Lines: Strategic Issues

Authors:
Ronny Kolb
Dirk Muthig

IESE-Report No. 013.05/E
Version 1.0
January 31, 2005

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Sauerwiesen 6
67661 Kaiserslautern

Abstract

The importance of quality assurance for software product lines and the need for effective and efficient techniques and methods for assuring quality in the context of product line engineering has recently been recognized by the product line community. Today, more and more practitioners and researchers are investigating ways of improving quality assurance techniques and processes, as well as develop concrete techniques for ensuring the quality of reusable components or products in a product line. However, strategic issues like organization, planning of product line quality assurance, or economic impacts have not been investigated in detail so far.

On December 8, 2004, therefore, the PuLSE™ team of the Fraunhofer Institute for Experimental Software Engineering (IESE), Germany conducted a eWorkshop to establish viable discussions among experts on experience, ideas, and solutions related to the strategic issues of quality assurance for software product lines, especially its impact on organizations and economics. A eWorkshop stands for an arrangement that enables invited experts and authorities within a certain area to discuss a topic remotely by using a web-based chat tool. The workshop was supported by VSEK (see www.software-kompetenz.de), a German internet portal providing empirical knowledge on software engineering technologies to the German public. VSEK is funded by the German Ministry of Education and Research (BMBF).

This report summarizes the results of the eWorkshop. The main findings of the eWorkshop are (a) quality assurance is more important for product lines than for traditional single systems, (b) product line engineering has a strong impact on an organization and its processes, (c) the two unique product line factors with respect to quality assurance are variability and scale, (d) there are new or different quality assurance activities or processes for software product lines, (e) the emphasis of product line quality assurance should be on unit testing as well as testing and evaluating the product line architecture, (f) a poor quality assurance strategy can negate the economic gains of product lines, and (g) if an organization wants to move towards product line engineering it must spent as much effort on quality assurance as on the construction of a product line infrastructure. Altogether, this first workshop has been a very large success, due to the level of participation of the invited experts and the quality of their statements. All participants voted to have another eWorkshop to discuss role of quality assurance in a product line business case.

Keywords: eWorkshop, Quality Assurance, Software Product Lines, Strategic Issues, Organization, Economic Impact

Table of Contents

1 Introduction

Software development today faces several challenges. There is a critical need to reduce cost, effort, and time-to-market of software products, but, at the same time, complexity and size of products are rapidly increasing and customers are requesting more and more quality products tailored to their individual needs. A promising approach to address today's software development problems and to make the software development process more efficient is the systematic, large-scale reuse of software artifacts over multiple products. Recently, reuse-based software development paradigms such as component-based software development and software product lines have increasingly received attention not just in the software research community but even more in the software industry as they promise – and have shown – to shorten the development time of software systems and to reduce development and maintenance costs.

Software product lines are an approach to software development that is based on the systematic and planned reuse of previous development efforts among a set of similar products. The product line approach enables organizations not only to reduce development and maintenance costs but also to achieve impressive productivity and time-to-market gains [CN01]. Although an effective means of improving the productivity of the software development process, product lines with its inherent reuse of software artifacts in combination with iterative development typically practiced today provides a massive challenge to quality assurance. Furthermore, without suitable strategies and techniques, quality assurance cannot keep pace with development productivity gains and so becomes the bottleneck of product line engineering. As a result, the goal of fast and cost-effective reaction to market and customer needs is hard to achieve. The importance of quality assurance for product lines and the need for effective and efficient techniques and methods for assuring quality in the context of product line engineering has recently been recognized by the product line community. Today, more and more practitioners and researchers are investigating ways of improving quality assurance techniques and processes, as well as develop concrete techniques for ensuring the quality of reusable components or products in a product line. However, strategic issues like organization, planning of product line quality assurance, or economic impacts of quality assurance in a product line context have not been investigated in detail so far.

To establish viable discussions among experts on experience, ideas, and solutions related to the strategic issues of quality assurance for software product lines, especially its impact on organizations and economics, the PuLSE™ team of the Fraunhofer Institute for Experimental Software Engineering (IESE), Germany conducted a eWorkshop on December 8, 2004. A eWorkshop stands for

an arrangement that enables invited experts and authorities within a certain area to discuss a topic remotely by using a web-based chat tool. It allows that people at different locations in the world can change ideas and arguments in a virtual meeting room. The workshop was supported by VSEK (see www.software-kompetenz.de), a German internet portal providing empirical knowledge on software engineering technologies to the German public. VSEK is funded by the German Ministry of Education and Research (BMBF).

This report summarizes the results of the eWorkshop and the issues addressed during the discussion. It recaps the major themes of the workshop and suggests future directions. The intention of the eWorkshop was to bring together researchers and practitioners in the areas of product lines and quality assurance to discuss organizational aspects of quality assurance for product lines and product line quality assurance economics. The discussion started at the Third Software Product Line Conference (SPLC) in August 2004, especially during the First International Workshop on Quality Assurance for Software Product Lines [KMM04]. The results of the workshop have shown that quality assurance in a product line context has – besides special techniques – a strong impact on an organization and its processes. In the eWorkshop, the discussion was particularly focused on the following issues:

Organizational Aspects of Quality Assurance for Product Lines

- Who is responsible for correcting defects in a product line artifact – The product line infrastructure team or the team that detected the defect?
- Who is to blame for if an error in a product is caused by a product line artifact?
- Should there be a separate team for ensuring the quality of the reusable product line artifacts?
- How is quality assurance organized in a distributed, multi-site product line organization?
- How are the different quality assurance activities for a product line managed and organized?

Product Line Quality Assurance Economics

- How should a product line organization invest its resources for quality assurance in a manner that will maximize its gains and minimize its risks?
- What is the best way of spending the limited resources available for quality assurance in a product line?
- Who pays for assuring the quality of the core assets in a product line infrastructure?
- How can costs and benefits be "traded-off" against resulting product quality?
- What are the economic impacts of different quality assurance strategies?

2 Overview

2.1 Participants

The participants in the eWorkshop were invited based on their background in software product lines and quality assurance. In total, 25 experts have been invited to participate in the eWorkshop. Unfortunately, only 6 eventually participated in the workshop. The workshop participants included:

- John D. McGregor, Clemson University, USA
- Henri Muccini, University of L'Aquila, Italy
- Christian Denger, Fraunhofer IESE, Germany
- Hideharu Teranishi, Ricoh Company Ltd., Japan
- Mikael Lindval, Fraunhofer Center Maryland, USA
- Forrest Shull, Fraunhofer Center Maryland, USA

2.2 Preparation

In preparation of the eWorkshop, participants were asked to fill out a reply form. The reply form which was sent to the participants is provided in Appendix A. Originally, it was intended to use the answers to the questions in the reply form to focus the eWorkshop and to guarantee a useful and efficient discussion during the eWorkshop. Also, it was planned to distribute a summary of the filled out reply forms to all participants before the eWorkshop so that all discussants would have had a chance to prepare. However, as only two of the participants provided a filled out reply form before the eWorkshop, it was actually not possible to use the answers for steering the discussion.

2.3 Agenda

The agenda of the eWorkshop consisted of the following discussion points:

- **Introduction**. Each participant was asked to say one or two sentences about his or her background and interests in the topics of the eWorkshop.

- **Organizational Issues.** As part of this discussion point, the following questions were raised:
  - How is quality assurance organized in a product line organization?

  - How do we set up different quality techniques in a product line organization?
  - Are there special activities or processes with respect to quality that are product line specific?

- **Economic Impact.** Regarding the economic impact of quality assurance in a product line context, the following questions were discussed.
  - What is the economic impact of quality assurance to a product line project?
  - How critical is the right quality assurance strategy to the success of product lines in practice?

- **Summary.** Finally, the discussion was summed up and potential topics for future workshops identified.

3 Results

This chapter synthesizes the workshop discussions and summarizes the main results. In the eWorkshop, two major points have been discussed: organizational aspects of quality assurance in a product line context and the economic issues of product line quality assurance. For more information, consult the chat log of the eWorkshop which is provided in Appendix B.

3.1 Organizational Aspects

The results of the First International Workshop on Quality Assurance in Reuse Contexts (QUARC) [KMM04], which was held in conjunction with the Third Software Product Line Conference in Boston at the end of August 2004, have shown that quality assurance in a product line context has a strong impact on an organization and its processes. Using this as a starting point, participants were asked to share their opinions and experiences regarding the organization of quality assurance for software product lines and the impact of product line engineering on quality assurance activities. All participants agreed that there are product line specific issues when talking about quality and quality assurance. However, there was consensus that there is no issue that is fully product line specific, that is which can be found in the context of product lines only. McGregor pointed out that the two unique product line factors are variability and scale. Variability gives many more combinations which have to be tested or inspected in order to insure quality and scale increases the penalty of allowing a defect to progress later in the life cycle. Participants agreed that one way to handle variability is to ensure quality as early as possible, since the earlier an artifact is inspected or tested, the fewer combinations are there. The problem, however, is that the artifacts are generic and hence the question is how to inspect or test generic artifacts. To address this problem, combinatorial techniques such as Orthogonal Array Testing which allow for a very large reduction in test cases with only a very small reduction in defect detection power should be used. In general, traditional quality assurance techniques such as inspections have to be customized in several ways so that they fit the specialties of a product line context. One aspect is to consider what you can and should, for example, inspect on infrastructure components before they are used in other projects.

Regarding the organizational aspects of product line quality assurance, the discussion showed that there are new or different quality assurance activities or processes for software product lines. Muthig, for example, mentioned that there have to be special processes that handle problem reports from projects.

Also there has to be someone who is responsible for deciding whether a problem is caused by a product line artifact or is a project-specific issue. McGregor and Denger remarked that this also has to be present in case of a multi-team development effort and that it is quite similar to this situation. The product line specific issue, however, is that there might be problem reports from many different projects at the same time and that they should be analyzed together. Also, the product line components produced by the infrastructure team might already be reused in many products. So there has to be a process for propagating fixes to defects found in reused components to projects. In general, however, the goal should be to have a strategy on how to find defects in reusable artifacts before they were propagated to all projects. In order to achieve this, the emphasis should be on unit testing as well as testing and evaluating the architecture.

Regarding architecture, participants agreed that in a product line context you aim at a stable architecture and that this will support the reuse of experience from one project to another. Muccini, however, expressed that from his experience it is hard to reconcile results from different phases of the software development process (e.g. requirements, architecture, design) since there is no synergy between the different quality assurance techniques. A very challenging task is to set up different quality assurance techniques and to find a valuable combination between them. In a product line context, however, it is easier than in a single system context to analyze dependencies between different development phases and quality assurance techniques and to transfer the results from one project to another.

3.2 Product Line Quality Assurance Economics

The second major discussion point of the eWorkshop was economics of product line quality assurance. At first, participants were asked what from their point of view the economic impact of quality assurance to a product line project is and how critical the right quality assurance strategy to the success of product lines in practice is. Muccini stated that quality assurance is certainly even more important for product lines than for traditional single systems since it impacts the quality of any product derived from the product line infrastructure. According to McGregor, a poor quality assurance strategy can certainly negate the economic gains achieved from scale. The risk that this happens is – from his point of view – very large, since development projects have always had size problems and a product line is a very large development project. When a defect is injected it can get magnified much more quickly than on a one off system. Muthig agreed to that and concluded that if an organization wants to move towards product line engineering it must spent as much effort on quality assurance than on the construction of a product line infrastructure. All participants agreed to that conclusion. Nevertheless, Muccini pointed out that quality is typically not considered very crucial in small and medium companies. Muthig

agreed and stated that this might be the reason why some people think that you cannot be successful with product line engineering in small and medium sized companies. Muthig and McGregor pointed out that quality engineering as a common theme for successful product line engineering is missing on the presentation slides of the Software Engineering Institute (SEI).

According to McGregor, quality assurance pays for itself if you take the time to count the hidden cost of late defect removal. Especially early quality assurance pays for itself by reducing the effort needed to find defects and by shortening the time to market. There are many sources on this but one of the first and most definitive was Capers Jones' Programming Productivity [Jon86]. Nevertheless, many companies only invest a very small amount of money in quality assurance. McGregor pointed out that we might need some analyses that show the business case for quality assurance. Another important aspect mentioned by Denger is that reliable data is required that shows the impact of quality assurance in terms of quality, cost, and time and so helps to convince people.

Muthig concluded that the fact that early quality assurance pays for itself and shortens the time to market applies as follows in a product line context. First, it motivates inspections at earlier phases in application engineering and secondly it motivates delivering high quality, reusable components to projects for the infrastructure. McGregor objected that he doesn't want to wait to do inspections in application engineering. Instead, the architecture and requirements should be inspected at the generic level. Muccini and Denger agreed to this, especially as waiting until application engineering might lead to missed chances. The question, however, is whether it is possible to reduce the effort for inspections in application engineering, for example by focusing inspections on special things like interfaces or integration issues, when we perform inspections in the infrastructure. McGregor argued that quality assurance can definitely be limited in application engineering to interaction defects. A good defect model that shows the types of defects that happen at what phase in development will identify the types to look for at any given phase. So far, however, there is no such defect classification scheme specific for product lines and whether there are product line specific defects. From McGregor's experience the biggest "defect" in a product line context is misunderstanding between the domain and application engineering teams. Denger expressed the opinion that a defect classification for product lines would not be so much different than one for single systems. At least the types of defects (as defined in the ODC [CBC+92]) might be quite similar or even the same as in single systems (e.g. misunderstandings, interface problems, instantiations, etc.).

Another interesting aspect discussed was whether the architecture of a product line helps to focus quality assurance effort on critical aspects. In general, there is a rule that 80% of the defects result from 20% of the components. According to Denger, the identification of these components is easier with a stable architecture as it is present for a software product line.

With respect to the economic impact, quality assurance can show a positive ROI if used early. This is not specific to product lines, but in a product line, for example, the fan out of a single requirement defect is much greater than for a single system. Participants observed, however, that it is much harder to inspect a product line architecture because of the variability points and therefore more costly.

An important aspect discussed was coming up with a business case for product line quality assurance. Regarding the required data for this, Muthig expressed that the product line community is working on economic models in general and that their exist data from projects strongly indicating that an incremental strategy works for industry. So far, however, the importance of quality engineering aspects for the overall success of product line engineering cannot be proven. Therefore, it is important to elicit quality assurance data from product line projects and to integrate this data into the already existing model of costs and benefits of product line development. The role of quality assurance in a product line business case was considered interesting by all participants and therefore all voted to have another eWorkshop on this topic.

3.3 Summary

The eWorkshop was the first in what we hope will evolve into a series of workshops addressing the problem of quality assurance for software product lines. In summary, the main findings of the eWorkshop are:

- Product line engineering has a strong impact on an organization and its processes
- There are product line specific issues when talking about quality and quality assurance, but there is no issue that is fully product line specific. The two unique product line factors are variability and scale
- One way to handle variability is to ensure quality as early as possible, since the earlier an artifact is inspected or tested, the fewer combinations are there.
- Traditional quality assurance techniques such as inspections have to be customized in several ways so that they fit the specialties of a product line context.
- There are new or different quality assurance activities or processes for software product lines, such as special processes that handle problem reports from projects. This is quite similar to the situation of multi-team development efforts, but there might be problem reports from many different projects at the same time and that they should be analyzed together.
- There has to be a process for propagating fixes to defects found in reused components to projects.

- The aim should be to have a strategy on how to find defects in reusable artifacts before they were propagated to all projects.
- The emphasis of product line quality assurance should be on unit testing as well as testing and evaluating the product line architecture.
- A stable architecture will support the reuse of experience from one project to another.
- Quality assurance is certainly even more important for product lines than for traditional single systems since it impacts the quality of any product derived from the product line infrastructure.
- A poor quality assurance strategy can certainly negate the economic gains achieved from scale and the risk that this happens is very large.
- So far, the importance of quality engineering aspects for the overall success of product line engineering cannot be proven.
- If an organization wants to move towards product line engineering it must spent as much effort on quality assurance as on the construction of a product line infrastructure.
- Quality is typically not considered very crucial in small and medium companies. This might be the reason why some people think that you cannot be successful with product line engineering in small and medium sized companies.
- Quality engineering is considered to be a common theme for successful product line engineering.
- Quality assurance pays for itself if you take the time to count the hidden cost of late defect removal.
- Especially early quality assurance pays for itself by reducing the effort needed to find defects and by shortening the time to market.
- Analyses that show the business case for quality assurance and reliable data that shows the impact of quality assurance in terms of quality, cost, and time are required to convince people to invest in product line quality assurance.
- Quality assurance can be limited in application engineering to interaction defects if inspections and testing are performed during domain engineering.
- It is much harder to inspect a product line architecture because of the variability points and therefore more costly.

Altogether, this first workshop has been a very large success, due to the level of participation of the invited experts and the quality of their statements. All participants voted to have another eWorkshop to discuss role of quality assurance in a product line business case.

References

[CBC+92] Ram Chillarege, Inderpal S. Bhandari, Jarir K. Chaar, Michael J. Halliday, Diane S. Moebus, Bonnie K. Ray, and Man-Yuen Wong, Orthogonal defect classification – A concept for in-process measurements, IEEE Transactions on Software Engineering, vol. 18, pp. 943–956, Nov. 1992.

[CN01] P. Clements and L. M. Northrop, *Software Product Lines: Practices and Patterns*, Addison-Wesley, 2001.

[Jon86] T. Capers Jones, *Programming Productivity*, McGraw-Hill, New York, 1986.

[KMM04] Ronny Kolb, John D. McGregor, and Dirk Muthig (Eds.), "Proceedings of the First International Workshop on Quality Assurance in Reuse Contexts (QUARC 2004", Fraunhofer IESE, Technical Report No. 096.04/E, August 2004.

Appendix A: Reply Form

Background

Please, give a brief (5 lines maximum) abstract of your background and experience in the areas of software product lines and quality assurance.

Workshop Topics

Prioritize the questions you would like to discuss with the other experts. Give marks from 1 to 9 where 1 means highest, 9 means lowest priority.

Organizational Aspects of Quality Assurance for Product Lines

Question	Mark
Who is responsible for correcting defects in a product line artifact – The product line infrastructure team or the team that detected the defect?	
Who is to blame for if an error in a product is caused by a product line artifact?	
Should there be a separate team for ensuring the quality of the reusable product line artifacts?	
How is quality assurance organized in a distributed, multi-site product line organization?	
How are the different quality assurance activities for a product line managed and organized?	

Product Line Quality Assurance Economics

Question	Mark
How should a product line organization invest its resources for quality assurance in a manner that will maximize its gains and minimize its risks?	
What is the best way of spending the limited resources available for quality assurance in a product line?	
Who pays for assuring the quality of the core assets in a product line infrastructure?	
How can costs and benefits be "traded-off" against resulting product quality?	

What are the economic impacts of different quality assurance strategies?

If you see any possible refinements of the above mentioned questions please add them.

Please give a first statement regarding your opinion to the top 3 of your favored questions.

Question 1:

Question 2:

Question 3:

Organization of Product Line Quality Assurance Activities

Does the domain have an influence on how quality assurance activities are organized? If so, what do you think what and how big the impact is?

Assume the product line infrastructure and its artifacts are developed incrementally as part of the development of concrete product line members by product development teams. How would you organize quality assurance activities for the infrastructure artifacts? In particular, how do you make sure that the product line artifacts are generic and reusable enough for other planned or potential products in the product line?

Assume you are responsible for organizing quality assurance activities in a product line organization that has a separate team for building reusable infrastructure artifacts. How do you ensure that the product development teams have trust in the quality of these artifacts and hence really use them in the development of their products?

Impact of Quality Assurance Strategies on Product Line Economics

There are several potential quality assurance strategies for product lines. For example, quality assurance activities can focus exclusively on the generic and reusable artifacts in the product line infrastructure or on the products built from these infrastructure artifacts. Both of these strategies are extremes and have benefits, but also limitations.

What do you think, should more effort be put in assuring the quality of the artifacts in the product line infrastructure or in assuring the quality of the products built with reuse of the artifacts from the infrastructure.

From your point of view, what are the consequences in terms of cost and quality of adopting a specific quality assurance strategy?

What do you think is the best way to spent quality assurance resources so that that is most likely to yield the best return on investment and maximize the chances of successful defect detection?

One possible way of improving quality assurance for product lines is to provide product teams with results and artifacts from quality assurance activities performed by the developers of the infrastructure artifacts or the developers of other products in the product line? Do you think that this approach allows improving quality of the resulting products while reducing the overall effort for assuring the quality of the products?

Quality assurance for generic and reusable artifacts in the product line infrastructure is far more complex than for artifacts used for a single system only. This is mainly because of the genericity of these artifacts and the resulting number of possible combinations that have to be checked. Due to these and other problems, quality assurance activities for the infrastructure artifacts might not only increase the effort for domain engineering and the investment in the infrastructure, but also delay the time-to-market of the first products that have to be built from the infrastructure. Even though quality assurance for the product line infrastructure artifacts might be more complex and hence costly and time-consuming, however, it might be pay off in the long-term, as the artifacts are of higher quality and less effort has to be spent in assuring the quality of the products built with the reuse of the artifacts. From your perspective, do you think that an investment in assuring the quality of the product line infrastructure and its artifacts will really be recouped after producing a few numbers of products using the infrastructure?

Practical Experience with Product Line Economics

If you have experience in organizing and performing product line quality assurance activities:

- How costly was it, compared to traditional single system development, to assure the quality of the generic artifacts in the product line infrastructure in terms of:
    a. Extra calendar time:
    b. Extra effort:
    c. Other costs: (Please specify)

- How much of the overall effort for quality assurance was spent on assuring the quality of product line infrastructure artifacts?

- How much of the overall effort for quality assurance was spent on assuring the quality of the products built from the infrastructure artifacts?

Appendix B: Chat Log

The following is the chat log of the discussion in the eWorkshop:

10 **Moderator**: The meeting is formally open, we're still waiting for anther participant to join

11 **Moderator**: The lead discussant Dirk Muthig will soon welcome everybody and start the discussion

12 **muthig**: Welcome everybody to the first eWorkshop on "Quality Assurance for Software Product Lines" We will focus today on two issues: Organization and Economics. Before we start (still waiting for other participants) and to get familiar with the technology, please, could everyone say 1 or 2 sentences about his or her background and interests in the topics of today.

13 **Teranishi**: Hello, everyone. I'm Hideharu Teranishi.working

14 **denger**: Okay, my name is Christian Denger, and I am working at Fraunhofer IESE in the area of quality assurance. There, I am focusing mainly on software inspections (i.e., qa in the early life-cycle phases). One major interest I have is to look how to customize software inspections to the context characteristics of software product lines

15 **mcgregor**: I am interested in the potential for QA to impact a larger scope in the context of software product lines. I am writing a column on Strategic Issues for the Journal of Object Technology and I think we do not think strategically nearly enough. My interest is how QA contributes to the company strategically.

16 **Teranishi**: I'm working as employee of RICOH (www.ricoh.com). My interest is how to balance QA techniques on my company.

17 **muccini**: Hello everybody. My name is Henry and my main research is on Software Architecture, model-based analysis and testing, and recently on software product line. I would like to understand how testing techniques at the PLA level may help for QA of PL

18 **muthig**: thank you for introducing yourselves. I think technology works now.

19 **muthig**: Okay, let's start. The results of our workshop at SPLC in Boston have shown that Quality Assurance in a product line context has - beside special techniques - a strong impact on an organization and its processes.

20 **muthig**: So, how is - from your point of view - quality assurance organized in a product line organization?

21 **mcgregor**: For the most part I do not want to "see" QA as a separate entity. I think it should be integrated into all of the functions of the company. I see QA as activities rather than a functional unit. Whether it is a product line or not.

22 **denger**: Well, from my point of view there are several ways of performing the QA process. Basically, two extremes: Do as much QA as possible on infrastructure components (reusable) ones or do as much as possible on project specific components, the big question is how to balance the efforts to achieve the biggest impact

23 **denger**: 21: What do you mean exactly by a functional unit

24 **muthig**: 21 that is a general statement. Do you think the only PL-specific are special techniques focusing on variability?

25 **mcgregor**: I mean a team whose responsibility is QA.

26 **mcgregor**: When I write a process every phase includes validation and verification activities. Many times these are performed by the team that produces the output from that phase.

27 **denger**: 25: I see, and I agree, having a central "unit" for QA issues might be a valuable thing. What are from your experiences the characteristics such a team should have in a PL context?

28 **muthig**: 22 balancing means that we cannot test reusable components completely but have to see what must be done in a project. any suggestions how balancing can be supported?

29 **muccini**: I agree with 21. In my experience with Software Architecture, however, I have seen how hard is to reconcile results from different software process stages (requirements, Architectures, Design, ...) and to create a unique QA process

30 **Moderator**: 29 Please note that if you add the reference first then an automatic link will appear to that statement

31 **muthig**: 29 in a product line context, you aim at a stable architecture. I think that this will support reusing experience from one project to another. do you agree?

32 **muccini**: 31 I partially agree... what I mean is that, up to day, there are some testing techniques at the Requirement level, something at the product line architecture level, some at the implementation level. However, there is no synergy between them

33 **mcgregor**: 24 I think that variability and scale are the two unique PL factors. Variability gives many more combinations and scale increases the penalty of allowing a defect to progress later in the life cycle.

34 **mcgregor**: 32 But QA is a lot more than testing, correct?

35 **denger**: 32: I agree, the big question is here how to find a valuable combination. I.e. how can we balance the efforts between QA activities on different levels

36 **muccini**: 34 Certainly

37 **denger**: 34: Definitely

38 **muthig**: 32 you are right. Would you agree that analyzing dependencies between the different stages can be transferred to next projects in a product line (more easily than in single system context)?

39 **muccini**: 38 YES

40 **Moderator**: 35 I think it's also a question of how to adapt, for example, inspections to product line architectures, or can you apply regular inspections to PLAs?

41 **muthig**: 34 so let's step back and address the question: how do we set up different quality techniques in a pl organization?

42 **denger**: 40: From my point of view you have to customize the inspections in several ways so that they fit to the specialties of PL context.

43 **denger**: 42: One aspect is that you need to consider what you can and should inspect, for example, on infrastructure components before they are used in other projects (i.e. which quality aspects)

44 **muthig**: 33 I agree if we look at the techniques to be applied. I would like to move the discussion towards differences in the overall organization. Are special activities, process wrt Quality that are pl specific?

45 **denger**: 42: Another issue is, how to address variabilities in a sufficient way with different techniques (how to test generic components, how to inspect them). With respect to inspections I wonder if you have any experiences with respect to industrial use of inspections in such a context

46 **muccini**: 44 The problem of moving from the framework engineering level to the application engineering level is typical and unique of PL

47 **muthig**: 46 right - so what does it mean for quality-related activities?

48 **mcgregor**: 44 This goes to the argument about unit testing. The sooner we inspect/test a piece the fewer combinations there are. So one way to handle variability is to ensure quality at the earliest possible point.

49 **Moderator**: 44 Do you imply that there might be new QA activities that need to be added in order to do QA on PLAs?

50 **mcgregor**: 46 It means we focus on emergent behavior. That is when two unit-tested items are combined what is different than when they were separate.

51 **denger**: 48: Yes, assuring quality as early as possible is important. Have you any experience how this is done, e.g., how to inspect generic components?

52 **Moderator**: 50 is this really a PLA specific issue?

53 **mcgregor**: 51 Yes, there are combinatorial testing techniques that let us choose parameters wisely and likewise we can focus inspections using these techniques

54 **mcgregor**: 52 Not really

55 **muthig**: 49 I think that we have, for example, special processes handling problem reports from projects. Someone must decide whether it is a problem with a pl artifact or a project-specific issue.

56 **mcgregor**: 54 but it may be magnified in a PL context

57 **mcgregor**: 55 But don't you have that when you have a multi-team development effort and someone has to route the problem to the appropriate team?

58 **muthig**: 56 I agree. To reuse quality effort one should solve this problem. otherwise a pl organization will loose some of the benefits coming through ple.

59 **denger**: 53: That sounds interesting, could you elaborate a bit more on what you mean by choosing parameters?

60 **muthig**: 57 okay but you get problem reports from many projects at the same time and you should analyze them together.

61 **denger**: 57: I agree, it is quite similar. The issue is that in a PL context, the components produced by the infrastructure team might be already reused in many projects so we have to have a process to handle this issue

62 **mcgregor**: 59 Techniques such as Orthogonal Array Testing (OATS) allows for a very large reduction in test cases with only a very small reduction in defect detection power. This allows you to choose parameters on unit tests to detect the most defects with the fewest tests. We use this approach on our Guided Inspection process as well where we inspect based on test cases.

63 **mcgregor**: 61 But the process you are mentioning here would be how to propagate fixes rather than finding defects, correct?

64 **muccini**: Please, do not forget in general that when we deal with PL we have components but also an architecture (software/hardware). Guaranteeing the quality of components is, on my opinion, the first step in guaranteeing the quality of the PL architecture

65 **denger**: 63: Yes that's correct. I think that this is also an important aspect to consider, when we find defects in reused components. However, we should definitely aim for a strategy on how to find the defects before they were propagated to all these projects.

66 **muccini**: 64 since we aim on reusing the same architecture for different products, another important point is to guarantee the quality of the architecture of the entire PL

67 **mcgregor**: 65 I agree, hence my emphasis on unit testing but also on testing the architecture.

68 **muthig**: 62 my observation is that we all agree that there are PL-specific issues when we talk about Quality. But if we point to a specific thing, we cannot find a thing that is fully pl-specific

69 **Moderator**: 67 How do you test the architecture?

70 **muccini**: 69 there are some techniques in which you extract abstract test cases from the architecture and then, through traceability, you drive the implementation test selection process (based on the SA)

71 **Moderator**: We're going to vote on statement 68. Observe that you need to press send once the vote is requested for

72 **mcgregor**: 69 We use our Guided Inspection process and develop a set of test cases. These are either applied manually in an interactive session or they are applied to working models of the architecture built using Rapide or some other executable ADL. The test cases come from the requirements not the architecture itself.

73 **Moderator**: /vote Do you agree on statement 68 as a summary of the discussion?

74 **denger**: 70: This means that you derive the test cases early but run them, when you have the implemented system, correct?

75 **mcgregor**: yes

76 08.12.2004 17:37:51 | Vote question: Do you agree on statement 68 as a summary of the discussion?

76 **muthig**: [voteyes]

76 **mcgregor**: [voteyes]

76 **muccini**: [voteyes]

76 **Teranishi**: [voteyes]

76 **denger**: [votenotsure]

77 **mcgregor**: 74, No

78 **mcgregor**: 74, we derive test cases and run them on a MODEL of the architecture not the final system That is too late.

79 **Moderator**: 76 We would like to follow up with the person who voted not sure, do you want to say why you voted that way?

80 **denger**: 78: That would have been also my concern about it

81 **denger**: 79: I have to confess, I wasn't sure if I understood the summary correctly

82 **muthig**: Before the discussion gets to close to specific techniques. I would like to move on to the second main topic of today: Economics

83 **muthig**: Before and while moving to product line engineering, people talk often about economics. So what is - from your point of view - the economic impact of QA to a PL project?

84 **muccini**: 83 If QA is important for traditional systems, QA for PL is certainly even more important, since it impacts the quality of any product derived out of the pl

85 **denger**: One aspect that might be discussed is whether the architecture helps us to focus on critical aspects in the system where we should focus QA. There is this general rule that 80% of the defects result from 20% of the components. With a stable architecture the identification of these components should be easier.

86 **mcgregor**: It is a positive. Applied early QA techniques pay for themselves by eliminating hours spent chasing down defects. There are statistics that show the fan out of a single requirement defect. So QA can show a positive ROI if used early.

87 **muthig**: 84 I rephrase the question: How critical is the right QA strategy to the success of product lines in practice?

88 **Teranishi**: For my point of view, it requires additional efforts to reuse the result of QA activities.

89 **denger**: 83: see 85

90 **Moderator**: 86 But is it any different compared to single system development?

91 **mcgregor**: 87 A poor QA strategy can certainly negate the economic gains achieved from scale.

92 **mcgregor**: 90 In a PL the fan out much greater so it is not different except on scale.

93 **muthig**: 91 how large is from your point of view the risk that this happens?

94 **Moderator**: 92 I would guess that it is much harder to inspect a PL architecture because of the variability points and therefore more costly

95 **muccini**: 87 I hope this may answer your question: if you guarantee the QA of a component, you (may) guarantee the QA of the system where the component will be deployed. However, if you may guarantee the quality of the entire pl, you may guarantee that any product has certain quality attributes

96 **denger**: 94: I agree, from my point of view in the PL context there are much more things to consider than in single systems (especially the variabilities).

97 **muccini**: 94 and 96 I agree

98 **mcgregor**: 93 I think very large. Development projects have always had size problems. A PL is a VERY large development project. When a defect is injected it can get magnified much more quickly than on a one off system.

99 **denger**: 95: Note sure about this. The question is how the components integrate in the system they are used in. I think you might be able to ensure a certain level of basic quality but you cannot assure good quality of the overall system

100 **muccini**: 99 Yes

101 **muthig**: 98 I agree and conclude that if an organization wants to move towards PLE it must spend as much as effort on QA4PL than on the construction of a PL infrastructure. What do you think about that?

102 **mcgregor**: 101 yes

103 **Moderator**: 99 What is "basic quality" and "good quality"?

104 **denger**: 101: yes

105 **Teranishi**: I agree >101

106 **mcgregor**: 103 "It depends" The context of the domain, government regulation, ... There are several factors that go into this.

107 **muccini**: 101 I agree with you too. However, in practice, or at least in small and medium companies, quality is unfortunately not considered very crucial!!

108 **muthig**: 102 referring back to very first statement - everybody in a PL organization must be concerned with PL quality.

109 **mcgregor**: 108 absolutely

110 **muthig**: 107 I agree but maybe this is the reason why some people think you cannot be successful with PL in SMEs. Do you agree?

111 **denger**: 108: I think that is a basic statement, you should always have the quality in mind, independent of PL

112 **muccini**: 110. Yes, we should provide some easy to use tool.. this is what works (usually) in many companies

113 **muthig**: 109 do you think you can educate everybody in a large industry organization wrt pl quality? I have my doubt because many engineers simple are not able to view from a PL perspective.

114 **mcgregor**: 110 I don't agree that SMEs can't do PL but I do agree that some think that is the case. QA pays for itself if you take time to count the hidden cost of late defect removal.

115 **denger**: 107: I think in SMEs the quality needs might be different. We need to consider what they perceive as a crucial quality level and how PLE can contribute to achieve this.

116 **mcgregor**: 112 I don't think there is such a tool. I think that many different tools are needed at different points in the life cycle.

117 **muthig**: 114 we know from MarketMaker that it is possible. On SEI slides on common themes of product lines I think Quality Engineering Capability is missing.

118 **muccini**: 114 John, I agree with this. However, in my experience in Italy, where we have many medium companies, they invest only a very few amount of money in quality (unfortunately)!

119 **Teranishi**: 111, Yes. We should pay attention even in non-QA departments.

120 **muccini**: 116 Agreed

121 **mcgregor**: 117 I agree. Maybe you and I could address that later.

122 **denger**: However, the overall question was on economics. So can we reduce QA efforts when applying PLE

123 **muthig**: 121 okay

124 **mcgregor**: 118 Maybe we need some analyses that show the business case for QA. The data I have seen have shown shorter time to market doing early QA.

125 **muccini**: 124 Interesting...

126 **muthig**: 124 that is very interesting. where are the data from?

127 **Moderator**: 124 Is that data for single system development or PL development?

128 **mcgregor**: 126 I would have to go back and look. That is not new nor is it PL. But of course there is the canonical IBM data that shows that removing a requirements defect at system test time requires 100 times more effort than removing it at the end of the requirements phase.

129 **muthig**: 128 okay I know this data wrt cost. but does this data say also something about delivery time?

130 **muccini**: I have a question: how can we convince medium and small companies about the usefulness of PL development? I do not know in your country, but in Italy it may be REALLY important!

131 **mcgregor**: 129 "It" is really multiple sources. And yes it shows that many times the schedule overruns would be eliminated if defects had not been present. I believe some of the data is in Steve McConnell's Professional Software Development book.

132 **Moderator**: /sethomepage fc-md.umd.edu

132 **muthig**: 130 it is also important for us, Fraunhofer in Germany. the business case is that the initial investment is very low so we follow an incremental or reactive approach

133 **muccini**: I am trying, for example, to convince Terma GmbH that they should use PL, but sometimes it is really hard!

134 **denger**: 132: In addition, I think showing with reliable data the impact (savings in terms of money, time etc) is important to convince people.

135 **muthig**: 131 There are two points where this applies in a pl context: (a) it motivates inspections at earlier phases in application engineering, and (b) it motivates delivering high quality, reusable components to projects for the infrastructure

136 **Moderator**: 132 Coming up with a BC for PL seems to be really important. What would you need to show in such a BC and do you have the necessary data to show that?

137 **mcgregor**: 135 I would not want to wait to do inspections in application engineering. The architecture and requirements should be inspected at the generic level. So I would not limit domain engineering to components.

138 **muccini**: 137 Yes

139 **muthig**: 136 we are working on economic models in general in the community. We also have data from our projects strongly indicating our incremental approach works for industry. However, we cannot prove yet the importance of quality engineering aspects for the overall success.

140 **denger**: 137: I agree, waiting until application engineering might lead to missed chances. The question I have is: when we perform inspection in the infrastructure can we then reduce the effort for inspections in application engineering (at least focus the inspections on things like interfaces and other integration issues?)

141 **muthig**: 137 I agree - I did not mean waiting but also doing it when adding project-specific requirements.

142 **mcgregor**: 139 Maybe we need a focus of the BC for PL. We are piloting our economic model with a company. I can try to elicit QA data specifically from them.

143 **denger**: 142: It would be interesting to see that data. What kind of data are you collecting there?

144 **muthig**: 142 that would be nice, also to analyze the Q-processes (wrt. to our initial discussion today)

145 **mcgregor**: 140 I think you definitely can limit QA in application engineering to interaction effects. Building a good defect model that shows the types of defects that happen at what phase in development will identify the types to look for at any given phase.

146 **muccini**: 145 Is there any ontology on defects which may be found in pl?

147 **mcgregor**: 143 We are building a model of costs and benefits of development like what Dirk and I published this summer in IEEE Software. We did not break out QA specifically at the model level but we could ask them about doing that.

148 **mcgregor**: 146 There is not one that I know of. Might be a good topic for one of my students.

149 **muccini**: 148 Good

150 **muthig**: 148 We are working on this topic - coming from single systems and extending it by pl aspects.

151 **denger**: 145: That's also the approach we use. But we had the same question, are there any PL specific defects

152 **muccini**: So, it seems we should investigate this point!

153 **mcgregor**: 151 My experience shows that the biggest "defect" is misunderstanding between the domain and application teams.

154 **denger**: 153: yes, I think that is an additional facet that is a result of PLE.

155 **denger**: 153: However, the question is, whether a defect classification would be so much different. At least the types of defects (as defined in the ODC) might be quite similar or even the same as in single systems (misunderstandings, interface problems, instantiations, ...)

156 **muthig**: I feel the discussion for today is coming to an end. Is there something you are willing to continue working on? I, for example, would like to continue looking at the role of QA in a PL business case.

157 **mcgregor**: 156 Yes, I think making a BC for QA in a PL would be interesting. I would work on that.

158 **muccini**: I have a question: if we identify the architecture for an entire pl, should we apply that architecture to any product in the pl? This is important to scale results

159 **muthig**: 157 should we plan for another eWorkshop focusing on business case?

160 **Moderator**: Let's vote on that

161 **Moderator**: /vote Let's run an eworkshop on BC for QA in PL

162 **muthig**: [voteyes]

162 **muccini**: [voteyes]

162 **denger**: [voteyes]

162 **Teranishi**: [voteyes]

162 **mcgregor**: 158 We use the architecture as the central point around which all else is coordinated.

163 **mcgregor**: [voteyes]

164 **Moderator**: Great result! We will summarize the results from this eworkshop and send out to you soon.

165 **muccini**: 162: Agreed. However, should we suppose the same architecture to apply in any product?

166 **mcgregor**: 165 Our architecture is built to do that.

167 **muccini**: 166 ok

168 **mcgregor**: Enjoyed the discussion. Thanks to IESE for organizing it.

169 **Moderator**: We will then start planning for an eworkshop on BC for QA in PL. It would be great if you could take the time and think about what's needed for such a BC and whether you think we have that data already. If not, how we get it.

170 **muthig**: Thank you, for your time and the interesting discussion.

171 **Moderator**: Thanks for participating today! We appreciate it!

172 **denger**: Thank you

173 **muccini**: Ciao and thanks

174 **Teranishi**: Thank you

175 **muthig**: Cheers

176 **Moderator**: It is now 6:30 PM here in Germany and we will sign off in a minute

Document Information

Title:	First eWorkshop on Quality Assurance for Software Product Lines: Strategic Issues
Date:	January 31, 2005
Report:	IESE-013.05/E
Status:	Final
Distribution:	Public