Advanced QA and Testing Approach Based on Service System Paradigm

Abstract. *The paper presents the service system paradigm as a tool dealing with some current quality assurance (QA) and testing problems, which are difficult to be handled within product paradigm. The real life experiences show that having a high quality product doesn't always mean to have no serious problems with quality when using this product. The paper illustrates the gap between QA and testing based on product paradigm and service oriented reality and describes the reasons, why QA and testing based on product paradigm are not sufficient in service environment. Further, it reveals the possible way the service paradigm can be used in QA and testing area to solve the mentioned gap. The case study of real problems with quality is illustrated and possible solution based on service system paradigm is proposed.*

Key words: service system, value-proposition, quality assurance, testing, V-model

1 Introduction

There are many standards, methodologies and recommendations for quality assurance and testing (for example RUP, ISO or ISTQB), but serious failures and issues still occur. At present lot of these failures and issues are not caused by defects in product, but by the improper use of the product. Such improper use can have the same or ever worse consequences than proper use of low quality product. The world is shifting from product paradigm to service paradigm as proves the fact that most of the gross domestic product in the most developed countries is generated by services (Factbook, 2008) and more and more organizations adopt service model (Arina and Oy, 2007). Quality assurance and testing aimed at products can guarantee sufficient level of quality for simple product, but does not cover service systems as a complex. There seems to be a gap between current QA and testing approaches and the service oriented reality.

To be able to manage quality in service oriented reality the QA managers and testers need to aim at the whole environment in which the product is settled, not only on the quality of product itself. We believe that QA and testing approach based on service system paradigm is the key to manage quality of service systems in situations where the QA and testing approach based on product paradigm is not sufficient.

In this paper, we focus on the situation, where product is deployed into the service system, and describe the differences between the QA and testing of this product deployment based on product paradigm on one hand and service system paradigm on the other. We introduce the advanced QA and testing approach based on service system paradigm as an extension of well-known V-model, which is the development model used for current QA and test management.

2 Related work

The current QA and testing activities can be divided into two areas:

1. Product testing – the QA and testing aims at the quality of single product.
2. Service testing – the QA and testing aims at the quality of the service.

In product testing area, current tools and methodologies are based on V-model, which is a development model created for software engineering. It defines a uniform procedure for software system development (Brőhl and Drőschl, 1995). V-model is illustrated in the figure 1. According to this model, required testing steps are taken and testing deliverables are created. QA and testing methodologies based on V-model have proved to be useful in product oriented economy, but

because of its focus on product development, the V-model in its current form seems not to be sufficient for managing the quality of the product in context of the whole service system. Our experiences show that by using current form of V-model, testing managers often focus only on the product without sufficient attention payed to the products environment. This approach can lead to failures initiated by improper use of high-quality product.

Take in Figure (1): V-model (based on thoughts of A. Brőhl and W. Drőschl)

In service system testing, the Service Oriented Architecture (SOA) requires new testing approach. The papers such as (Massuthe and Reisig and Schmidt, 2005) aim at the verification of single service and especially verify that the single service will finish correctly when run by the service client. The testing of the whole service system, where the set of agents can provide or consume different services or create another agent, is still the subject of research.

In situation where the finished product is customized for specific organization, the connection of product approach and service approach is required. In this case, the V-model helps the customer's testing team to manage product requirements validation and its verification by acceptance tests execution (product's requirements corresponds to the higher part of V-model, the lower parts of testing are realized by the product developer in the development phase). However, acceptance test of the product does not appropriately include the rest of the service system (relevant part of the organization and its environment), which includes people, organizations, value-propositions, etc. (see figure 2).

Take in Figure (2): The illustration of common testing focus on product deployment into the service system (ovals represent service agents, rectangles represent service targets)

We propose an advanced QA and testing approach which is focused on the product quality in context with the whole organization. We introduce an enhanced V-model, which extends the current form of V-model by the analysis and testing of the relevant part of the service system.

3 Proposed approach

The paper focuses on the deployment of customized product into the service system, which is at present very often the case. For the purpose of this paper, we consider the product to be some sophisticated product such as IT system and the service system to be an organization. However, the described problem is much more general and can be used for another situations, including the deployment of new service or set of services in different types of service systems.

3.1 Service system definition

According to (Spohrer and Maglio and Bailey and Gruhl, 2007) the service system is defined as a tuple:

- People
- Technology
- Shared Information
- and Organizations

connected by value proposition.

The illustrations of service system is in the figure 3.

Take in Figure (3): The illustration of service system (Gadrey, 2002)

Value proposition defines the added value offered to service client by service provider and as such it is the key, that creates the service system. Service provider provides the service and service client consumes the service in order to fulfill value-proposition. The service is not created only by service provider but is co-created in close cooperation between service provider and service client (see figure 4).

Take in Figure (4): The customer input intensity in case of product and service paradigm (Spohrer, 2006)

3.2 The product usage via service system view

In order to realize the gap between the common product oriented QA and testing approach and service system reality, it is necessary to understand how the product is used by the service system. The product is placed into the company with the purpose to provide or enable to provide a service or set of services, otherwise it is useless. The agent, who uses the product to provide the service is service provider of this service. The agent, who consumes this service is service client. The product can also provide some services directly. As was described above, the service clients cooperate with the service provider in order to create the service. In terms of this cooperation, the service clients often use these direct services provided by the product, which means that in this context the service provider is the product. Some of the services are complex, which means that for given entry it is not possible to estimate their exact result. They correspond mainly to organization's goals, objectives or deliverables. The other services are simple services, where the service client expects exact result for given entry. They correspond to simple actions.

To describe the product usage via service system view, we differentiate between two types of services.

We define the **simple service** as: *The simple service is service provided directly between the service provider and the service client, which requires a short-term communication between the service provider and service client. It can be described by simple use case or as a function.*

Note: The service provider is often a tool such as IT or automatized system.

We define the **complex service** as: *The complex service is a service, which is provided by individual or organization (service provider) to another individual or organization (service client), composed by many other services and requires long-term cooperation between the service provider and service client. It often uses some product or set of products. It cannot be described by simple use case or as a function.*

The service provider of the complex service creates this service by using many single services provided by the product and other agents. The same the service clients of the complex service co-creates this service by using several simple services provided by the product and other agents. The usage of the product in context of complex and simple services is visualized in figure 5. The scope of common product acceptance testing in context of service system view are the simple services provided by the product, as illustrated in figure 6.

Take in Figure (5): Illustration of product usage via service system view

Take in Figure (6): Illustration of product acceptance test scope in context of service system view

In case of Enterprise Information System (EIS), the complex service can be for example collection of accounting data, the service provider an accounting department and the service clients all employees of the organization. One of the simple services is the entry of an invoice, where the service provider is the accounting module and the service client is the employee.

Furthermore, we differenciate two types of service systems.

We define **simple service system** as: *The simple service system is the service system defined by the simple service.*

We define **complex service system** as: *The complex service system is the service system defined by the complex service.*

3.3 The extension of V-model

In order to overcome the gap between testing, which ends at the level of product requirements, and service oriented reality, where the value-proposition of the complex service between service provider and service client is arranged, we define new level of V-model. On the left side of our extended V-model, complex service system requirements are included. On the right side, testing of complex service system is included. Also the scope of classic V-model upper steps differ (see figure 7). These additional steps and their connections to the classic V-model are described in more details in the next section.

Take in Figure (7): The extension of V-model

3.4 Testing process

The testing process proposed for situation, where the customized product is deployed into the organization, is described in form of use cases. These use cases correspond to the additional steps in extended V-model. We presume that the product offers only one complex service to simplify the description of given steps. However, the proposed process can be used in the same way in case the product offers the set of complex services.

Use Case 1: Complex service system requirements

Scope: Complex service system

Primary stakeholder: *Analytics*

Other stakeholders: *Testers*, *Complex service provider*, C*omplex service client*

Value: *Analytics* – Value-proposition and other requirements of the complex service are specified.

Testers – Value-proposition and other requirements of the complex service are validated. The basis for complex service system tests are gained.

Complex service provider – The requirements of the complex service are in harmony with complex service provider's expectations, especially value-proposition and requirements of the complex service provider's competences.

Complex service client – The requirements of the complex service are in harmony with complex service client's expectations, especially value-proposition and requirements of the complex service client's competences.

Core scenario:

1.1. *Analytics* identify the complex service

1.2. *Analytics* identify the complex service provider

1.3. *Analytics* identify the complex service client

1.4. *Analytics* facilitate the value-proposition between *complex service provider* and *complex service client*

1.5. *Testers* validate the arranged value-proposition

1.6. *Analytics* specify other complex service requirements (for example SLA) in coordination with *complex service provider* and *complex service client*

1.7. *Testers* validate other complex service requirements

1.8. *Testers* prepare acceptance tests for complex service

Comparison with classic QA and testing approach: This process enables to validate the solution at the level of complex service system architecture, while the classic QA and testing approach starts at the level of single product. Thus we can find issues that are difficult to be found while using the classic QA and testing approach. This approach also enables that these issues can be found in earlier phases of product deployment into the service system.

Notes: At present, there is no ideal method of value-proposition specification. There are several initiatives in this area. IBM for example patented goal-service modeling method where goals are used for identification of individual services (Ang et al., 2008). Another approach represents goal-oriented requirements engineering techniques (Bertrand and Darimont and Delor and Massonet and Lamsweerde, 1998), which could be possibly also used as the basis for value-proposition specification method.

Use Case 2: Simple service system requirements

Scope: All simple service systems that create the complex service system

Primary stakeholder: *Analytics*

Other stakeholders: *Testers*, *Simple services providers*, Simple *services clients*

Value: *Analytics* – Value-proposition and other requirements of the simple services are specified.

Testers – Value-proposition and other requirements of the simple services are validated. The basis for simple service systems tests are gained.

Simple services providers – The requirements of the simple services are in harmony with simple services provider's expectations, especially value-proposition and requirements of the simple services provider's competences.

Simple services clients – The requirements of the simple services are in harmony with simple services client's expectations, especially value-proposition and requirements of the simple services client's competences.

Core scenario:

2.1. *Analytics* identify the simple services

2.2. *Analytics* identify the simple services providers

2.3. *Analytics* identify the simple services clients

2.4. *Analytics* facilitate the value-proposition between *simple services providers* and *simple services clients*

2.5. *Testers* validate the arranged value-propositions

2.6. *Analytics* specify other simple services requirements (for example SLA) in coordination with *simple services providers* and *simple services clients*

2.7. *Testers* validate other simple services requirements

2.8. *Testers* prepare acceptance tests for simple services

Comparison with the classic QA and testing approach: The scope of this process includes all simple services required by the complex service, not only those offered by the product. This also enables to validate the specification for simple services that are offered by another agents than by product and take in account the solution of found issues in the architecture of all simple services.

Notes: In the case of simple service provided by the product, the simple service system requirements are the requirements of the product usage by the product user and it is used as the basis of the software product requirements, design and implementation. In the case of simple service provided by the individual or organization, the simple service system requirements identify the requirements of cooperation between individuals or departments and should be defined by the organization's processes or methodologies or by another directive, for example by law.

Use Case 3: Simple service system acceptance test

Scope: All simple service systems that create the complex service system

Primary stakeholder: *Testers*

Other stakeholders: *Simple services providers, Simple services clients*

Value: *Testers* – Quality of the simple service systems is evaluated.

Simple services providers – Quality of simple services provider's competences is evaluated. Simple service systems correspond to simple services providers' expectations.

Simple services clients – Quality of simple services client's competences is evaluated. Simple service systems correspond to simple services clients' expectations.

Core scenario:

3.1. *Testers* execute the simple service systems acceptance tests

3.2. *Testers* document the issues found in simple service systems

3.3. *Testers* evaluate the quality of the simple service systems

Comparison with the classic QA and testing approach: In this step, all simple services are verified, including the competences and features of simple service providers and simple service clients (as illustrated in figure 8), while classic QA and testing approach focuses only on the simple services offered by the product (see figure 6). This enables identification of issues in simple services that are offered by another agents than by product that would cause defects in complex service system and initiate the corrections before product deployment into the organization.

Take in Figure (8): The scope of simple service acceptance test

Use Case 4: Complex service system acceptance test

Scope: Complex service system

Primary stakeholder: *Testers*

Other stakeholders: C*omplex service provider, Complex service clients*

Value: *Testers* – Quality of the complex service system is evaluated.

Complex service provider – Quality of complex service provider's competences is evaluated. Complex service system corresponds to complex service provider's expectations.

Complex service clients – Quality of complex service client's competences is evaluated. Complex service system corresponds to complex service provider's expectations.

Core scenario:

4.1. *Testers* execute the complex service system acceptance tests

4.2. *Testers* document the issues found in complex service system

4.3. *Testers* evaluate the quality of the complex service systems

Comparison with the classic QA and testing approach: The scope of the service is the whole service system as illustrated in figure 5. While the classic testing process finishes with the evaluation whether the product is high-quality or not (in terms it does not contain critical defects), the testing according to the proposed approach continues with verification whether the complex service system with the product embedded is able to fulfill arranged value-proposition.

Notes: The specificities of complex service system such as constant state of flux or heterogeneity of service system parts require different testing techniques then simple product testing. However, the comprehensive identification of service system specificities and design of service system test techniques is out of scope of this paper.

Service system monitoring

Beyond the testing of service system, which is defined by the complex service provided by the product, testers have to monitor different service systems.

- They monitor the service system, which is defined by the complex service provided by the product, in customization and deployment phase in order to identify the changes in this service system that would require the changes in simple service requirements.
- They monitor the same service system after the product deployment, because, in opposite to the program code, the service systems are in the constant state of flux. Therefore, after some period, the changes in the service system not considered in the product can increase the service system quality.
- Finally, they monitor the global service system, represented for example by an organization, after the product deployment. The product deployment can affect another service systems in the global service system indirectly and the quality of the global service system could decrease.

If any changes evolve, it is necessary to define the affected service system (it can be the simple service system, complex service system or even the whole organization) and to validate the changes against it's value-proposition. This would eventually require the modification of the service system parts in order to assure service system quality.

3.5 Benefits of the proposed approach

The biggest advantage of this new approach is the possibility to validate complex service system architecture and find issues in it before the evaluation of the quality of single product together with evaluation of the product in context of the whole complex service system and testing all parts of complex service system. Because issues found in later stages of program or system development are generally more expensive than if they are found in earlier stage of development (see figure 9), the main goal of QA and testing is to find the issues as soon as possible in the development process. Proposed approach support QA and testing department in reaching this goal and enables to save

huge amount of money.

Take in Figure (9): The relative cost to remove a defect

The coordination of simple services by means of complex services solves the same problem that occurs in project management area, where the single projects are managed within the scope of some program. The project management of single projects have to be coordinated in context of the whole program to achieve the quality of the project management as a whole. In spite of high-quality project management of single project, the program as a whole can fail, in case the projects are not coordinated (for example some project consumes the resources required by another projects with higher priority). This principle is specified in the project management standard published by two significant project management associations (IPMA, 2006), PMI (2008).

4 Example

In this section we illustrate the problems with quality which are difficult to be handled within the product paradigm on real case study of Enterprise Information System (EIS) Implementation.

4.1 Enterprise Information System Implementation

Situation
The functionality of current EIS in consultancy oriented company became insufficient so the management decided to replace current EIS by the solution available on the market and adjust it regarding the requirements of their organization. They were aware of severity of this decision so they carefully selected proved, high quality solution. Great number of seminars for all users have been organized. The implementation of new system had to deal with some technical issues, but there were no critical failures of the system. However, the employees criticized the system for no specific reason and it seemed they lost their motivation not only to work with the system but at all. The efficiency of their work also decreased. From the view of ICT manager, the product was high-quality and he blamed the employees for sabotage the new EIS system.

Based on the fact how professional their employees are and how well they used the old system, the board decided to carefully analyze the situation with individual employees. Their founding were very interesting. The employees had no serious problems with technical quality of the system. There were two main non-technical issues identified. First, nobody explained to the employees why it was necessary to change the EIS so they did not realize the added value of new EIS and refused to invest their energy to learn how to work with new system efficiently. They used new EIS unwillingly. Secondly, the new EIS contained new attendance system, which completely changed the fashion of attendance perception. While in old system the employees filled their attendance into Excel sheet according to how long they worked, the new system automatically logged in real time the time they came and left the work. The system also automatically inserted the half hour lunch-brake after 4 hours of work, according to the Czech law, no matter if the employee left for lunch or not. Finally, the system reported a warning in case the 40 hour week work limit was not achieved. For this reasons the most hard-working employees were forced to change their habit to finish their work at home and they had to stay in the office till late evening to have this work logged. Their weekend work was not taken in account but they were forced to sit in the office even in case they have finished all assigned goals but have not spent enough time on it. This naturally led to situation where the employees started to think more about how much time they have to spend in the office to reach the 40 hour limit rather than how to reach the goals that needed to be achieved. Their efficiency decreased and their motivation was ruined. It is important to say that there were no

serious attendance issues requiring new attendance system.

Proposed approach application
This example clearly shows how the company gained high-quality software and accurate data about employees' attendance at the workplace but the quality of the whole service system (organization) decreased. In the next paragraphs, we describe how the proposed approach can be used to prevent this situation. We focus on one specific component of EIS, the attendance system and title the service system defined by the attendance evidence and evaluation value-proposition as attendance service system.

Using the proposed approach, the process of attendance service system specification would be as follows:

1. *Analysts* would identify the attendance evidence and evaluation as a complex service offered by the attendance service system.
2. *Analysts* would identify the accounting department as attendance service provider
3. *Analysts* would identify the employees as attendance service client
4. *Analysts* would facilitate the value-proposition of attendance evidence and evaluation between *accounting department* and *employees*.
5. Based on the arranged value-proposition, *analysts* would decide that the attendance component will not be deployed and specify the minor required in the current attendance service system in form of requirements.
6. *Testers* would validate specified requirements and prepare acceptance tests for attendance service system.
7. Based on attendance service system requirements, *analysts* would specify the changes required in simple service systems embedded in attendance service system, for example new requirements on attendance excel sheets filling.
8. *Testers* would validate specified requirements and prepare acceptance tests for embedded simple service systems, fox example competence test for employees.

In design and implementation phase (the lower part of the V-model) the team responsible for attendance excel sheets would prepare new templates and the team responsible for competences of the accounting department and employees would prepare the methodologies and train employees.

The process of attendance service system would be as follows:

9. *Testers* would execute simple services tests and evaluate accounting department and employees competences against the specified requirements.
10. *Testers* would execute tests for the accounting service system and evaluate the results against the specified value-proposition.

Beside, the QA department would constantly monitor the changes in the whole acceptance service system and evaluate whether the changes in one part of acceptance service system, for example personal changes in accounting department, require any changes of the rest of the acceptance service system.

Thus the organization would save money and time for attendance component implementation and seminars organization, the attendance evidence and evaluation would more appropriately correspond to the organizations needs and the motivation of employees would increase.

5 Conclusion and future work

We have described the product usage in context of service system paradigm and proposed the advanced QA and testing approach based on service system paradigm. The extension of V-model have been revealed and described together with the comparison with the common V-model usage.

Presented approach enables QA managers and members of testing team to realize the differences between products and service systems and to upgrade their QA and testing techniques so these techniques will more suitably fit to the service oriented environment. Thus the resources required for the product deployment into service system can be reduced, while the quality of the whole service system can increase. The application of proposed approach has been illustrated in real case study.
The paper aims at the situation, where the product is deployed into the service system defined by this product, which is a significant part of service system testing area. In situation where more and more organizations adopt service model, there is strong need to assure the quality of the general service system, which can be composed of many other service systems and it is in constant state of flux. This area is the subject of our future research.

References:

Ang, Jenny Siew Hoon (San Francisco, CA, US), Arsanjani, Ali P. (Fairfield, IA, US), Cherbakov, Lyubov (McLean, VA, US), Galambos, George M. (Montreal, CA), Holley, Kerrie Lamont (Montara, CA, US), Janson, David Hugh (Raleigh, NC, US), 2008, *Goal-service modeling*, US Patent 20080027784.

Arina, T. and Oy D. (2007), "Web 2.0 Business Models", available at: http://www.slideshare.net/infe/web-20-business-models-270855 (accessed 07 March 2009).

Bertrand, P., Darimont, R., Delor, E., Massonet, P., and Lamsweerde, A. v. (1998), "GRAIL/KAOS: an environment for goal driven requirements engineering", *In Proceedings ICSE'98 - 20th International Conference on Software Engineering*, IEEE-ACM, Kyoto.

Brőhl, A. and Drőschl, W. (1995), "Das V-Modell", *Oldenbourg Verlag*, Wien.

Factbook (2008), "Field Listing - GDP - composition by sector", available at: https://www.cia.gov/library/publications/the-world-factbook/fields/2012.html (accessed 05 May 2009).

Gadrey, J. (2002), "The Misuse of Productivity Concepts in Services: Lessons from a Comparison between France and the United States", *Productivity, Inovation and Knowledge in Services: New Economics and Socioeconomic Approaches*.

IPMA (2008), *ICB - IPMA Competence Baseline Version 3.0*, International Project Management Association, Zaltbommel: Van Haren Publishing, Netherlands.

PMI (2008), "*A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - 4th Edition*", PMI.

Massuthe, P. and Reisig, W. and Schmidt, K. (2005), "An Operating Guideline Approach to the SOA", *Annals of Mathematics, Computing & Teleinformatics*, Vol. 1 No. 3, 2005.

Spohrer, J. (2006), "Services Sciences, Management and Engineering (SSME): An Emerging Multidiscipline", *Frontiers in Service Conference,* IBM Almaden Service Research.

Spohrer, J., Maglio, P., Bailey, J., Gruhl, D. (2007), "Steps toward a science of service systems", *IEEE Computer*, Vol. 40 No. 1, pp. 71-77.