A Comprehensive Planning Framework for Selecting and Customizing Quality Assurance Techniques

Frank Elberzhager, Christian Denger
Fraunhofer Institute for Experimental Software Engineering, 67663 Kaiserslautern, Germany
{frank.elberzhager, denger}@iese.fraunhofer.de

Abstract

Different quality assurance techniques can be applied in the software development lifecycle to improve the quality of a product. Today, no holistic approach for planning and customizing inspection and testing techniques in a given context exists. In this paper, we present a framework which consists of the main aspects necessary for a systematic planning of quality assurance activities in different development steps. We present the core elements of the framework: influence and variation factors, characteristics of techniques and a role concept. The framework is a first step towards a systematic planning approach for quality assurance, which has to be refined in future research. Based on the elements, we recommend an initial process how to apply the framework. With this, a basis is developed to support quality managers planning quality assurance techniques in a more systematic way.

1. Introduction

Quality of the final software product is one of the most decisive factors that determine the market success of a company. In general, companies need to carefully consider the time and efforts they spent on quality assurance techniques, as resources are often limited and time-to-market is often short. Thus, quality assurance has to be performed in a most efficient way. Consequently, analytic techniques, which can be distinguished between static and dynamic techniques, need to be customized to the specific company and project constraints. Moreover, in order to have most benefits from quality assurance, it is necessary to consider different techniques and plan their application in the overall development cycle.

Based on goals and context constraints in a certain development context, a quality manager has usually to decide an appropriate mix and timing of different techniques to check the quality of the product or the intermediate artifacts. This mix should be as efficient as possible in fulfilling the goals (e.g., finding defects, ensuring quality attributes) and should fit the development context constraints. To perform such a decision, it is necessary to know about the performance (e.g., how effective and efficient is the technique?) and the characteristics (e.g., which experience is necessary to perform the technique?) of the techniques. From our experience, such decisions are typically based on experience and characterized by a high degree of uncertainty, i.e., there is no explicit, systematic support in selecting the right set of quality assurance techniques. Due to the fact that usually a variety of different ways exist how to apply a quality assurance technique, an adaptation to the concrete context is necessary. Consequently, a quality manager has to decide which variant of the quality assurance technique should be applied. To perform such a non-trivial decision process it is important to know those factors that influence the adaptation, i.e., information about the context have to be gathered to adjust the techniques. However, this information is usually not explicitly known by a quality manager which makes a decision of the most beneficial quality strategy very difficult today.

The state of the art regarding quality assurance does not give conclusive answers. For example, we cannot determine the right mix based on existing empirical knowledge. Such knowledge is often related to one quality assurance technique evaluated in a certain context, which makes it difficult to generalize the results. Furthermore, the current body of knowledge does not consider the combination of techniques over several life-cycle phases. Consequently, we perceive empirical knowledge as a good starting point to gather the main factors important for a structured planning and customization support but this is not sufficient and has to be enhanced with company-specific measurement data.

33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)
0-7695-2977-1/07 $25.00 © 2007 IEEE

Objectives: To improve the efficiency of quality assurance processes and thus, the quality of the software itself, a framework is introduced that provides guidance on how to plan quality assurance activities. Thereby, the focus is not only planning single techniques but developing a lifecycle wide quality assurance strategy.

To support quality managers in planning and determining the best fitting quality assurance strategy in a given context, we defined a framework making relevant factors for a systematic planning approach explicit. The first step to establish such a quality strategy is the identification of factors that might have an impact on the quality assurance and the specific techniques, respectively. In practice, the project constraints are often implicitly determining how the quality assurance activities are performed. However, in order to allow a strategic planning of quality assurance activities over the entire development cycle, it is important to make these factors explicit. Based on the context information, it is possible to develop quality strategies that describe how a technique is used in a given context. For this, the concrete values of the techniques have to be determined, i.e., the potential ways to vary a technique have to be known. To support this step, the variants of the techniques have to be known explicitly, too. In addition, it is necessary to know what the effectiveness of the technique is in general and in the concrete context. The first information is described by available empirical knowledge; the second one can only be gathered by measuring in a concrete development context.

Based on the framework, which summarizes and describes every relevant factor necessary for planning quality assurance, we further propose a process describing which steps are necessary to develop a customized quality strategy in a given context. Basics how to get all necessary information for planning quality assurance are described. This is the first step towards a systematic planning process. The basis for an efficient planning, identification of influence-, variation factors and characteristics, is already defined and described within this paper. The refinement of the initial framework, the process and its steps is part of further research activities.

The remainder of the paper is structured as follows. Section 2 presents the related work and describes the concepts we used in our framework. In Section 3 the entire planning- and customization-framework is described in detail. Section 4 suggests a process how the framework can be applied to get a quality strategy and how to customize certain quality assurance techniques. Finally, Section 5 summarizes the main aspects and presents an outlook.

2. Related work for quality-assurance planning

Considering planning activities for quality assurance, we identified different frameworks that consider similar aspects. Biffl and Halling [6] describe a conceptual framework to support the planning process for software inspections. Three levels of decisions structure different responsibilities, starting from a quality-management level to a detailed inspection-level. For making appropriate decisions while planning the inspection process, the framework suggests capturing existing knowledge from literature regarding context constraints that influence a decision. Inspection process parameters, which can be varied to customize inspections to match the context constraints, are a second source of knowledge. Depending on the decision level, a quality manager, inspection manager and the inspector is assigned to plan the quality assurance or the concrete inspection process.

In [7], the authors present a characterization scheme to support selection of testing techniques. They argue that a huge number of techniques exist to create test cases, but practitioners are often not aware of all variations and especially miss support in deciding which technique is most suited in a given context. The criteria of the characterization scheme are independent from experience or background knowledge of individual testers to allow a more objective selection process. A distinction between project (context) characteristics and technique characteristics is done. With this, the idea is to compare the characteristics of the technique with those of the context to decide which technique is suitable or not in a given situation. Some elements of the scheme are, for example, technique characteristics (e.g., cost of application, inputs, dependencies), tools, test cases (e.g., defect type detected, number of tests) and the objective of the technique. By the characterization scheme of the test techniques, a quality manager is supported in selecting the best fitting test approach regarding the context. Thus, the context in which the quality activities are planned must be carefully understood and considered.

Beside the exemplarily mentioned frameworks we identified several approaches of how to customize single techniques to a given context or with respect to a quality goal. Inspection techniques focusing on certain quality attributes (e.g., [8] focuses on usability) or testing approaches considering reuse topics (e.g., [9] describes reusable test assets) are some examples. Furthermore, we identified approaches concentrating on quality improvement of the product or the

33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)
0-7695-2977-1/07 $25.00 © 2007 IEEE

development process, but no procedure how to plan and customize quality activities was presented (e.g., [10] specifies an improvement by company specific quality goals).

Finally, approaches considering cost estimation models (e.g., [12]) and economic-based frameworks (e.g., [13]) further characterize the planning-context of quality assurance. Basically, they concentrate on the question how much quality assurance is enough in a given situation, but the approaches mainly focus only on one quality assurance technique.

The main conclusion of the analysis of the identified frameworks is their focus on single development phases or single techniques. Neither support how to plan quality assurance regarding the whole development lifecycle is presented, nor is described in detail how to plan the application of different quality assurance techniques in combination. However, several frameworks consist of promising ideas describing what is necessary to consider when planning quality assurance, in general. But no holistic framework exists in which the different concepts are represented.

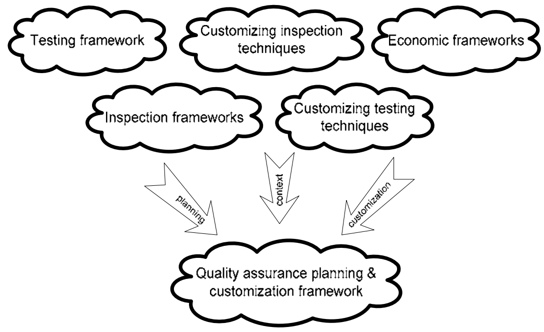

Figure 1: Existing models for planning and customizing quality assurance

Figure 1 presents an overview of the existing models we have identified and shows the main aspects we have used for and adapted to our planning and customization framework. In detail, the "good practices" for quality assurance planning integrated in our framework are (we identified such concepts as good practices that are common in several existing planning approaches):

- Context factors that describe restrictions or constraints the quality assurance has to fulfill and thus, influence the planning of inspection and testing techniques.
- Variants of the techniques and characteristics (the main benefit from the existing techniques) that describe how the technique can be adjusted to a given context.
- A separation of concerns, i.e., a differentiation of decisions or planning levels seems to be beneficial to reduce complexity.
- A role concept that defines responsibilities for decisions on different levels.

3. A generic quality-assurance planning framework

To support a person responsible for planning quality activities over the entire development lifecycle, more is necessary than only to support planning of single quality assurance techniques. Therefore, our framework contains elements from the existing frameworks, but is enhanced by one additional abstraction level. This means, our framework does not substitute planning single techniques, but embeds this activity into a top level planning process. Moreover, the goal is to gather information how quality techniques work together in a most effective and efficient way in a given context. Thus, our framework offers planning support on a high level (development-wide quality assurance planning) together with a planning support on a technique level.

The core of our planning support is a framework that helps to understand the most relevant decision factors during a planning process [1]. From our point of view, the key-to-success of a strategic planning is a profound understanding on how quality assurance techniques

1. are influenced by external context constraints and project restrictions,
2. impact the desired goals, qualities, and restrictions (timing, cost) of the project; i.e., what are the benefits and risks when applying a quality assurance technique,
3. can be customized to a given context and how this is supported, so that their application is most efficient and effective.

The goal of the overall framework is twofold: 1.) It shall support practitioners in *planning* a mix of quality assurance techniques that are optimal to fulfill given goals; and 2.) it shall support practitioners in *selecting and customizing* quality assurance techniques in a systematic way to make them applicable in a given context. The impact of the customization should be visible in terms of the benefits and risks.

3.1. Basic elements of the framework

To clarify the goals it is necessary to start with some basic concepts. These are mainly taken from the existing frameworks and are defined in our context.

Influence factor: A factor that influences the customization of a technique (e.g., the available effort for quality assurance, the quality goals to achieve).
Variation factor: An element of a technique that can be adapted in a given context (e.g., the number of inspectors the reading technique used in an inspection, the test coverage criteria, the effort spent).
Technique characteristics: Aspects of a quality assurance technique that represent relevant information to understand the nature of the technique (e.g., pre-requisites for application, required effort, expected efficiency).
Quality gate: Defined point in the development life-cycle at which quality assurance activities are performed on a certain work product and its quality is checked (e.g., requirements, design, system).
Quality assurance technique: A single technique to ensure the quality of a work product created throughout the development life-cycle, e.g., inspection or testing.
Quality assurance strategy: A mix of quality assurance techniques, determined how and at which quality gates to apply.

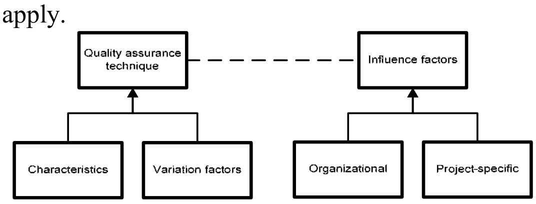

Figure 2: Basic elements of the customization framework

Based on the definition of the terms, Figure 2 shows our framework and the relationship of its elements. In general, a quality goal should be achieved by applying certain quality assurance techniques. Quality assurance techniques are defined by characteristics (e.g., their effectiveness, prerequisites to use). Furthermore, a technique can often be used in different (variant) ways depending on the goals to achieve (e.g., for inspections, different reading techniques exist or phases can be omitted). While the characteristics and the variation factors describe control elements of the technique, influence factors specify the context which can be further defined into organizational (e.g., an overall defined process model) and project-specific factors (e.g., time or manpower constraints). Summarized, a quality assurance strategy has to be chosen and instantiated (regarding variation factors), guided by the characteristics and depending on the influence factors to achieve the defined goals.

3.2 The framework elements in detail

Figure 2 represents only the abstract view on our framework and introduces the core concepts. However, to support systematic planning support, these concepts need to be further refined. First, a top level view on the **influence factors** is described in Figure 3.

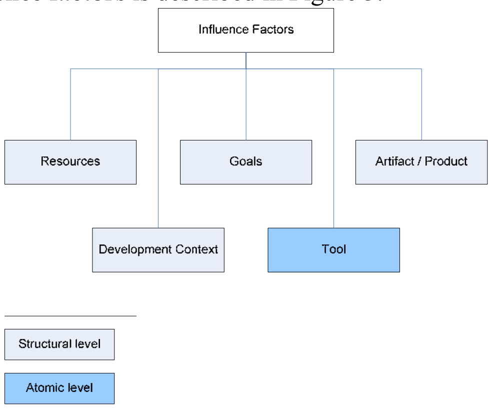

Figure 3: Influence factors of the framework

We identified 5 elements and 21 sub-elements. For each of them, we gave a definition and possible values for easier initial appliance (see Table 1 where some sub-elements of the influence factor resources is shown as an excerpt of the entire set of influence factors). Figure 3 shows structural elements (those ones that are refined into sub-elements) and one atomic element (no refinement is necessary).

Table 1: Excerpt of the refined elements of the influence factor 'resources'

Sub-Element	Definition	Value
Experience	The experience of the team in applying testing / inspection techniques.	High, medium, low
Time	The available time for quality assurance (e.g., inspecting a document, testing a module).	<2, 2-10, 10-20, >20h
Team size	Number of potential testers / inspectors, available for the task.	<2, 2 - 4, 4 – 7, >7

The 'development context' comprises the *development process* (e.g., waterfall, iterative), *programming language* (e.g., java, C++), *notation* (e.g., varying UML diagrams) and *reuse*. The objective is at least to gather enough information to determine the quality gates.

The 'goals' include aspects the quality assurance should focus on or what should be ensured by using certain quality assurance techniques. Certain *quality attributes* such as reliability, functionality or usability can be in focus of the quality assurance and defined *quality levels* have to be reached. Another possible goal is to avoid certain *defect classes* (e.g., insufficient performance, calculation defects). Finally, *secondary goals* (e.g., team building) or testing a *new technology* can be in focus.

The 'resources' comprise every factor that limits the quality assurance or serves as a prerequisite in using a certain technique. The *team size*, the *effort* and the *time* define the available resources. The *experience* and the *knowledge* are team-specific and express which level is necessary to perform quality assurance (e.g., the more experience people have in applying a quality assurance technique, the less do very structured techniques help regarding their efficiency). Questions about resources have to be answered for each quality gate again.

The 'artifact / product' summarizes information about the extent of the product that is analyzed (e.g., code, requirements specifications, design artifacts). In detail, this includes the *size*, the *complexity* and the *criticality* of the product as well as information about *former defect density* which can focus the quality assurance on certain parts of the artifact. The extent of quality assurance can depend on a *complete development cycle or just configuration* of existing parts of the software.

Finally, the 'tool' can limit the decision which quality assurance technique to perform (e.g., if a testing tool can only check certain coverage criteria).

After the project context is characterized (i.e., determination of the relevant influence factors), the variations of techniques can be considered and defined. The **variation factors** describe in which different ways quality assurance techniques can be applied. Figure 4 presents an overview of the possible variation factors captured in our framework. Again, atomic and structural elements are distinguished. In addition, general elements are those, which are important to consider for both inspection and testing. In addition, technique-specific factors exist. All in all, four general factors, two inspection-specific, and three testing-specific factors exist. Again, 21 sub-elements are defined.

The general elements comprise resources, the focus and scope of quality assurance and the entry / exit criteria. 'Resources' include the *effort*, the *team size* and the *experience* that is selected to perform a quality assurance technique. The 'focus' determines if quality assurance should concentrate on certain aspects, for example a *quality attribute* (e.g., maintainability, reliability) or a certain *defect class* (e.g., calculation defects). The 'scope' specifies which *parts of the artifact* should be inspected or tested. Finally the 'Entry / Exit criteria' define when to *start* and when to *stop* the quality assurance.

Next, the inspection specific variation factors are the reading technique and the inspection process. 'Reading techniques' are, for instance, ad hoc, checklist-based or scenario-based reading ([3], [4], [5]). The 'inspection process' consists of the *applied phases* (e.g., planning, preparation, meeting), the involved *roles* (e.g., organizer, inspector) and the *process structure* (e.g., sequential, n-fold). Furthermore, information about what is *documented*, how the *defect detection* is performed (individual, team) and about the *meeting organization* (no meeting, iterative, etc.) can be determined.

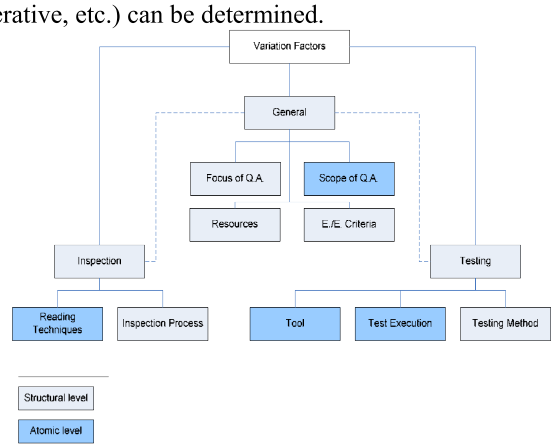

Figure 4: Variation factors of the framework

For testing, the application of a 'tool' and how the 'test execution' (manual, semi-automatic, automatic) is done are variant aspects. Finally, the used 'test methods' differ and are distinguished between *functional, structural, formal* and *other test methods*.

Finally, **characteristics** of the techniques support the decision process by providing information, for example, about their effectiveness, efficiency and applicability. Empirical knowledge serves as an initial basis. From this, heuristics are derived. Heuristics are rules that should help in instantiating a concrete quality assurance technique, i.e., which variation of the technique should be applied. So, the heuristics support in determining the concrete variations of the quality assurance techniques. For this, the influence factors determine which heuristics are applicable in a given context (e.g., if enough time (influence factor) is available for perspective-based reading (variation), choose this technique for checking the design document; if resources and experienced personnel are available, perform inspections of requirements documents). The heuristics should be enhanced by project and company specific measurement data to get a more specific decision support which is one goal of our future research activities (e.g., review of a design document in a project found 50% of the total defects, therefore it is sensible to perform it in another project).

33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)
0-7695-2977-1/07 $25.00 © 2007 IEEE

4. Applying the framework

Based on the outlined factors (i.e., influence and variation factors, characteristics) in the previous section, a conceptual process was developed which shows the basic steps to be taken in defining a holistic quality assurance strategy [2]. For this, characteristics of the quality assurance techniques and influence factors (context) are considered and serve as a starting point for planning activities. The relation between them (i.e., based on the questions which quality assurance techniques are available and how to adopt them to a given context) is established by heuristics which instantiate the relevant variation factors.

4.1 A suggested process

To reduce the complexity of coming to a decision regarding planning and customizing quality assurance, different abstraction levels have been introduced. It starts with high-level decisions defining the overall quality assurance strategy in terms of quality gates. Afterwards, decisions are made about how quality gates are instantiated, i.e., which quality assurance techniques are performed at the quality gates and how the techniques are customized.

In accordance with the decision levels mentioned, we defined two roles with different responsibilities. On the one hand a quality manager (QM), who has to decide the overall quality assurance strategy, i.e. this role is responsible for defining the quality gates. For this, the QM has to gather the relevant context factors, i.e. the influence factors. On the other hand, a technique manger (TM) has to customize concrete quality assurance techniques based on the context constraints, i.e., the TM has to determine certain variation factors of the quality assurance techniques. Both roles can select concrete quality assurance techniques that should be applied at the defined quality gates. This selection is supported by information about the technique characteristics. Table 2 shows exemplarily the main activities and decisions of a QM together with the input this role needs for managing the decisions. The output presents what the role has to determine.

The identification of the context factors and thus, the constraints in which the quality assurance activities must be performed is an essential step that lays the basis for all later activities. The customization framework supports quality managers in determining concrete, context specific values for the identified influence factors and with respect to the goals that should be achieved. Based on this information, quality assurance should be planned over the development lifecycle.

Table 2: The responsibilities of a QM

Role	Quality Manager (QM)
Activity	Defining quality gates based on context factors, determining quality assurance techniques.
Decisions	Based on the influence factors and their values, how should a quality strategy look like?
Required input	Technique characteristics, context constraints, risks in applying a technique
Output	Quality gates, quality techniques
Exemplar questions	In which development phase should quality assurance techniques applied? How much effort is available for quality assurance?

For this, we propose a planning and customization process involving the framework elements mentioned before. With this, quality and technique managers should be supported in their quality assurance activities. The process demonstrates at which steps in the development cycle certain framework elements are needed. The individual steps and how the elements of the framework are integrated are shown in Figure 5.

The whole planning process consists of six steps starting with determining the relevant influence factors, i.e., gathering the context factors. This is supported by guidelines how to apply the framework influence factors. In these guidelines (applied by the QM), for each of the 21 influence factors we describe activities how to get the necessary information, the goal, exemplar sources, the result and a rationale why this factor is important to gather.

Based on the knowledge of the project context and goals (i.e., the specific influence factors), determination of quality strategies can be performed. For this, knowledge about the technique characteristics is important in order to decide which techniques and which variations fit best in the given context. This can be supported by heuristics (e.g., if many defects have been detected in former projects which have been introduced during design, use an inspection technique if resources are available). We developed an initial set of heuristics based on empirical knowledge, but much more research is necessary to get a more beneficial set of heuristics. The outcome of this step is a set of possible quality assurance strategies in a given context (the outcome can have different levels of detail, at least quality gates, or already determined techniques at the quality gates).

Next, each quality strategy has to be evaluated. While doing the evaluation, it is important to balance the risk, the effects and the achievable goals of each strategy before selecting the most fitting quality assurance strategy.

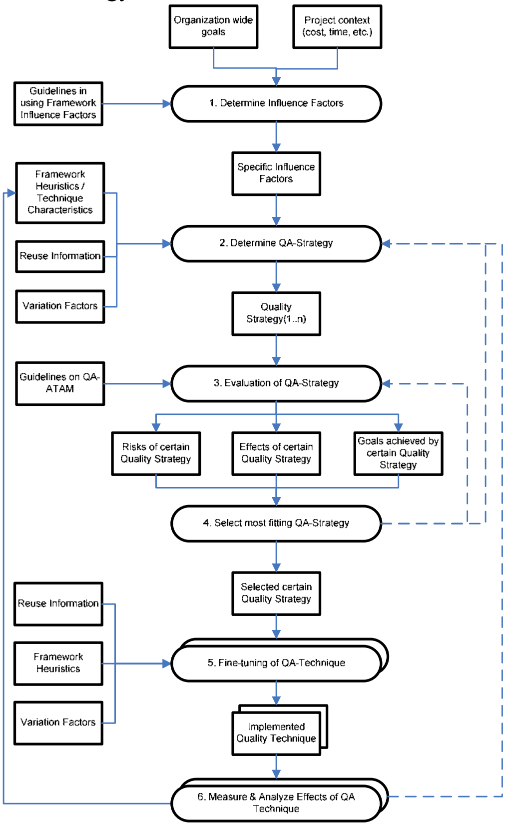

Figure 5: The overall planning process

As shown in Figure 5, it is possible to perform step 2, 3 and 4 more than once which depends on the degree of decisions from a quality or technique manager and how to refine the strategy. However, the output of step 4 is a unique quality assurance strategy which can have different degrees of detail (starting from only consisting of quality gates to a very detailed strategy with refined quality assurance techniques).

In step 5, it is the task of one or more technique managers to tailor the chosen techniques to a certain development step considering all constraints decided in earlier steps. Due to different quality gates, this step has to be repeated at every quality gate. This should be supported again by heuristics which have to be detailed on a technique level (e.g., reading technique perspective-based reading (PBR) or checklist-based reading (CBR) should be used if enough resources are available due to a high defect detection potential). The result of the fine-tuning step is that the technique can be applied and it is known how to do this.

Finally, the chosen quality strategy has to be implemented, monitored and, if necessary, further tailored if project context factors change during the development cycle (e.g., it was much more time needed for an early inspection which leads to less time for testing). Thus, measurement activities of the selected and implemented quality assurance strategy are essential. By doing this, the reliability of the planning process in future projects can be increased due to project specific data. Possibly, new heuristics can be derived. Furthermore, it can also influence the actual selected quality strategy which results in a re-planning of certain steps (e.g., the requirements inspection found many defects and therefore, an additional design review should be performed). This is shown by the arrow from step 6 to step 2 in Figure 5.

4.2 Necessary refinement of the process

Considering the proposed planning process (Figure 5), we only focused on the first step so far (by developing guidelines) and a refinement is necessary to define how to perform the following steps. In our opinion, the evaluation step is the most crucial one and offers high optimization potential. To offer a basis to make the most beneficial decision about the best fitting quality strategy in a given development context, much information has to be considered (risks of strategies, effects, and goals). Furthermore, detailed guidelines how to perform this step should be developed (we want to use ideas from ATAM [14] to develop a "QA-ATAM", i.e., a process should be defined to evaluate the potential quality strategies based on objective indicators or scenarios). Simulations are another idea. With this, it should easily be possible to vary certain adjustment screws regarding the quality assurance techniques and to have a look at the effects. However, a good database is necessary to perform suitable simulations.

Abstracting from concrete process steps, we are interested in refining and developing new heuristics. On the one hand, these should be technique specific and help in deciding when to use which technique. On the other hand, we hope to get information that shows relationships between different quality assurance techniques (e.g., when using inspection technique PBR under certain considerations, structural testing can be omitted or using design reviews with certain effort can reduce effort for module testing at 20%). All in all, we want to offer a comprehensive and well-defined process that supports quality managers in planning and customizing a quality assurance strategy over the entire development lifecycle in a given context.

5. Conclusion and future work

In this paper we presented a framework for planning quality assurance over the entire development cycle. During a literature research, we only identified planning approaches which focus on single quality assurance techniques such as inspection or testing, but do not consider the entire development process and how to use different quality assurance techniques in a holistic way. This was our motivation to define a framework which comprises every relevant impact on such a strategy and how to vary quality assurance techniques depending on the concrete context.

Thus, we started defining 21 influence factors that characterize the context and therefore, have an impact on the planning process. For gathering these factors, we developed a checklist and guidelines to support a quality / project manager in quality planning activities.

Moreover, we identified 21 variation factors which show how quality assurance techniques can be customized to a certain context. This was enhanced with characteristics of the techniques, which were the basis for an initial set of heuristics. These heuristics further support a quality or technique manager in choosing the most fitting quality assurance technique in a given context, but have to be enhanced in future research activities.

Furthermore, we proposed a process consisting of six individual steps in which the framework is integrated. With this, quality and technique managers should be supported in planning and customizing quality assurance techniques. The process starts from gathering the context, suggesting different quality strategies and evaluating them against each other to find the most suitable one for a given development context. The main benefit of the developed framework is its collection of the main factors which are important in planning a holistic quality assurance strategy. To apply the framework in a beneficial way, it is important to derive more heuristics (general and specific ones) which is one goal for future research. This also comprise the question which quality assurance technique is suitable in which development phase.

Next, we will refine certain steps of the planning process. While we developed guidelines for determining the influence factors, more work has to be done for the evaluation step. First ideas how to get a significant basis for judgment are results from a defect flow model [11] or simulation approaches. Guidelines that have to be developed should support this step, too. Finally, once each step of the process is defined (i.e., it is clearly described what to do at each step), we want to evaluate the planning framework in a real context, i.e., we want to know how beneficial it supports people responsible for quality in the development process.

Acknowledgement: Parts of this work have been funded by the LifeCycleQM project funded by the BMBF grant 01 IS E05 D.

6. References

[1] C. Denger, and F. Elberzhager, "A Comprehensive Framework for Customizing Quality Assurance Techniques", *IESE-Report 118.06/E*, September 2006

[2] C. Denger, and F. Elberzhager, "Basic Concepts to Define a Customized Quality Assurance Strategy", *IESE-Report 013.07/E*, February 2007

[3] A. Aurum, H. Petersson, and C. Wohlin. „State-of-the-Art: Software Inspections Turning 25 Years". *Journal on Software Testing, Verification and Reliability*, 2002, 12(3):133–154

[4] V. R. Basili. "Evolving and Packaging Reading Technologies", *Journal of Systems and Software*, July 1997.

[5] O. Laitenberger, "Cost effective Detection of Software Defects through Perspective-based Inspections", *PhD Theses in Experimental Software Engineering*, Fraunhofer IRB Verlag, 2000.

[6] S. Biffl, and M. Halling, "Managing Software Inspection Knowledge for Decision Support of Inspection Planning", *Managing Software Engineering Knowledge*, Springer, 2003

[7] S. Vegas, and V. Basili. "A Characterization Schema for Software Testing Techniques", *Empirical Software Engineering, 10*, Springer Science and Business Media, Inc. Manufactured in the Netherlands, 2005, pp. 437–466

[8] Z. Zhang, V. Basili, and B. Shneiderman, "Perspective-based usability inspection: an empirical validation of efficacy". *Empirical Software Engineering 4*, 1999

[9] A. Mäntyniemi, and P. Mäki-Asiala. "Improving efficiency of testing with test reuse: development of reusable test assets". *Proceedings of the First International Workshop on Quality Assurance in Reuse Contexts* (QUARC), 2004

[10] A. Birk, P. Derks, M. Elf-Mattila, J. Hirvensalo, and R. van Solingen. "The PROFES Improvement Methodology and Experience from its Industrial Application", 4. SQM-Kongress Software-Qualitätsmgmt 'Made in Germany', 1999

[11] M. Ciolkowski, and M. Klaes; „Einführung eines Fehlerstrommodells bei IBS Ziele, Vorgehen und resultierende Klassifikation", *Report in the LifeCycleQM project*, Report Number IESE-195.06/E, 2006

[12] L. Huang, B. Boehm. "Determining how much assurance is enough? A value-based approach", *Proceedings of the 7th international workshop on Economics-driven software engineering research*, 2005

[13] D. Port, M. Halling, R. Kazman, and S. Biffl, "Strategic Quality Assurance Planning", *Proc. of the 4th Int. Workshop on Economics Driven Software Engineering Research*, 2002

[14] R. Kazman, M. Klein, and P. Clements, "ATAM: Method for Architecture Evaluation", *Technical Report SMU/SEI-2000-TR-004,* 2000

0-7695-2977-1/07 $25.00 © 2007 IEEE