The perception of software quality and testing in Czech software companies

Anna Havlickova

Faculty of Informatics and Statistics

University of Economic, Prague

Czech Republic

xbora08@vse.cz

Abstract: *The aim of this article is to introduce the survey results of the current state of testing and quality management in software companies in the Czech Republic. The article includes answers to questions about the view of quality management in these companies and their employees' level of knowledge of software testing as well as an analysis of these results and a proposal for solving the identified issues. Moreover it describes the current software testing education system and its characteristics as it has a direct impact on this perception.*

Key words: software quality management, software testing, software quality assurance, staff training, research of quality in software companies

1. Introduction

In today's fast moving world, the quality of software is one of the main competitive advantages and conditions for successful businesses. Over the past three decades many different software quality assurance standards have arisen, as well as tools for test management and specialized conferences addressing these standards and tools. Many books have been written on how to test better and how to improve the quality of the developed software. Software testers are now often specially trained, publish information at conferences, as well as forums and blogs, and have a number of research papers on the Internet.

This article seeks to inquire what the current awareness about software quality management[1] is of employees of software companies who are actively involved in shaping quality[2], and how much attention is being paid to testing during the software development process in the Czech Republic.

In this article, following questions are discussed:

1. How do employees understand the concepts of software testing and quality assurance?
2. Are they able to identify the strengths and weaknesses of the testing techniques they use?
3. Where do they draw their knowledge of software testing and quality management from?
4. How do they see the quality of testing in the company they work for?
5. How big is the budget for testing in relation to the entire software development budget?

Also the article pays attention to how this level of awareness was achieved through education and trainings.

In introduction, this article pays attention to some of the current scientific discoveries and role of software quality in competitive corporate world. In second chapter, it describes the current education

---

[1] *By quality management we also understand testing as a tool for evaluating the state of quality*

[2] *By employees who are actively involved in shaping quality we mean:*

- *Testers, test analysts and other persons involved in the testing process*
- *Programmers, software architects, analysts and other persons involved in the development process*
- *Quality assurance specialists*
- *Project managers, test managers and other persons who manage the development process or some part of it.*

and training system in the area of software testing and quality management. The used methodology (a survey) is explained in the third chapter and the results are presented in the fourth. In a conclusion you can find the proposed improvements.

1.1 The role of software quality in business

With the rapid expansion of software into most medium and large enterprises and the decreasing cost of software acquisition through new technology and development methods, the development of new applications is no longer concentrated on rapid development and is now increasingly focused on fundamentally innovating manufacturing procedures to gain a competitive advantage. Gradually, it is becoming more important for companies to produce high quality products, rather than just producing them in the shortest time possible.

The aspect of software quality is moving to the forefront as an important element affecting the competitiveness of one company over another with nearly equivalent software. The low cost maintenance of operations and the assurance of cheap and quick future changes are becoming a major asset. In contrast, software defects often get media attention and counterintuitive and unreliable software will deter many customers.

In their book, the Economics of Software Quality, Jones and Bonsignour state that the bigger the developed software product is, the greater the savings obtained by achieving high quality are compared to a low quality product. (Jones et al., 2011) Even though this is not the main reason for the pursuit of high quality, this fact is another argument that quality strengthens the competitiveness of the company.

Proportions of cost savings from achieving high quality over low quality in overall software development are shown in tab. 1. High quality includes taking appropriate measures to prevent errors, using effective methods for detecting defects in the source code (code inspection and static analysis) and highly efficient testing[3].

Tab. 1: Proportions of cost savings from achieving high quality over low quality in overall software development (Jones et al., 2011)

Functional Points[4]	10	100	1000	10,000	100,000
Savings	14%	16%	19%	22%	25%

This supports the often mentioned argument that software quality is for free, as by increasing the cost of preventing defects on one hand, it reduces the costs of debugging on the other (Slaughter et al., 1998).

Especially nowadays, when companies are trying to minimize their costs and investments related to information technology and prepare for an uncertain future caused by the crisis in the Europe, the quality, reliability and low maintenance costs of software are assets that distinguish strong software companies from competitors and win contracts from European and American firms looking for cheap but reliable partners for software development.

The systematic improvement of processes and the education of employees involved in software development and testing is the key to ensuring software quality. Evidence for this assertion is that the idea of the systematic and continuous improvement of processes and the education system can be found in a number of methodologies and standards such as COBIT, ITIL, RUP, and more.

---

[3] By highly efficient testing it is meant testing that, measured by the following scales, achieves values near 1:

Defect Detection Efficiency (DDE) = number of detected defects before deploying the software / total number of defects

Defect Removal Efficiency (DRE) = number of bug fixes / total number of defects

[4] Functional points are used to measure the size of a program. The number of functional points is determined by a complex method that does not follow a simple algorithm that can be explained here but the functional points is considered more accurate as the measure of size than a number of statements.

In spite of an importance of knowledge and education, Desai states that problems with their transfer and reuse are among the ten major problems in software testing.

The ten major problems in software testing according to Desai (Desai et al., 2011) are:

1. Low reuse rate of software testing knowledge
2. Barriers in software testing knowledge transfer
3. Poor sharing environment for software testing knowledge
4. A serious loss of software testing knowledge
5. Impossible to achieve the most optimum distribution of human resources quickly
6. Determine whether adequate testing has been done- Types of testing
7. Test Coverage-Number of test cases
8. Determining whether or not outputs are correct
9. Determining what you can say about the software when testing is completed- Go or No Go
10. Measuring performance characteristics

With the increasing emphasis on the role of software quality and the process of its development in the company, the question of the level of knowledge and experience of those who actively participate in shaping the quality of software in the company comes to the fore. Quality management must be seen as an integral part of company management, including its activities in the field of information technology. Software development is to be rid of its privileged status as an elusive art, and to begin to be seen more as a normal process where its performance and return on investment can be measured and managed. The objective of development and inseparably also of operational management, because there is a clear causal relationship of cause and effect, is to achieve the maximum return on investment and the optimal ratio of the discounted value of acquisition and operational costs to the performance of the software.

More on managing costs and the economic effects of business intelligence can be found in (Vorisek et al, 2010), (Jones et al., 2011).

1.2 The aim of the paper

Software in the Czech Republic has become a common product in the last decade and it is subjected to an increasing number of demands on quality, price and supply reliability. And, as it will be explained in the following chapter, increasing the quality of testers and other staff who are actively involved in creating software quality has a direct effect on these attributes.

The aim of this paper is not only to discover what attention is paid to testing in software development, and what the awareness of software testing is for employees of software companies who are actively involved in determining the quality of the products, but also it describes possibilities of education in this area as an education is known to have direct impact on knowledge and propose how to improve the knowledge and skills of employees through better education.

The study thus provides insight into the results and implications of contemporary education in the area of software testing.

2. Education in the area of software quality

Companies increasingly use detailed test scenarios[5] and test automation tools when assessing software quality, and thereby delegate even larger portions of test execution to inexperienced staff or machines. On the other hand, it increases the opportunity for the specialization of experts in the field of testing. Knowledge, skills and experience in specific business areas and software testing continue to play an important role.

[5] *The test scenario is a description of a test in the form of a sequence of steps, along with a description of entered data and the expected responses of the system. Test scenarios can be divided into test cases where each test case is focused on testing one functional flow of the specific functionality.*

Merkel shows that significant differences are observed in the performance of software testers, the best testers are at least half again as effective as the average in the team. It means that varying abilities and knowledge greatly influence individual results (Merkel et al., 2010).

In his study, Iivonan identified the common characteristics of high-performance testers[6] in three Finnish software companies (Iivonen, 2009)[7]:

- Experience with the product
- Experience in the business area
- Experience with programming
- Experience with special testing techniques
- Writing good bug reporting
- Maintaining perspective
- Enjoying finding defects
- Care, diligence, patience and endurance

From that we can infer that a software testing education needs to be focused not only on the theory of quality assurance and testing processes and techniques but also on the knowledge of the specific business domain and practical testing experience.

The education can be obtained by these sources and methods:

Tab. 2: Possibilities of education [author]

Sources of education:	**Methods of education:**
• University studies • Employer • Self education • Certification and requalification	• Training and courses • Conferences and colleagues (mentors) • Internet (professional articles, blogs, forums) • Books • Practice

While self-education is uncontrolled and its quality depends on the chosen method of acquiring knowledge, attending specially designed courses within higher education should provide a reasonable assurance of education quality.

Table 3 provides an overview of the specific knowledge and skills in software testing and quality management that can be gained from those public universities in Prague that have study programs focused on information technology.

[6] *Highly efficient testers were identified by analyzing the contents of database systems for reporting defects based on the number, severity and proportion of resolved bugs.*

[7] *The knowledge of tools is missing from the list, as defects found with automation tools cannot usually be matched with particular testers.*

Tab. 3: Overview of the required knowledge and skills and if they are taught at five significant public universities with study programs focused on information technology.[8] *(author)*

Available knowledge and skills	ČVUT[9]	VŠE[10]	CUNI[11]	ČZU[12]	VŠCHT[13]
Software testing	Yes	Yes	Yes	**No**	**No**
Software quality management (prevention, costs, metrics)	Yes	Yes	**No**	Yes	**No**
Tool(s) for test management	Yes	Yes	**No**	**No**	**No**
Tool(s) for functional testing	Yes	Yes	**No**	**No**	**No**
Tool(s) for performance testing	Yes	Yes	Yes	**No**	**No**
Tool(s) for unit testing	Yes	Yes	**No**	**No**	**No**
Tool(s) for code analysis	**No**	**No**	Yes	**No**	**No**
Programming	Yes	Yes	Yes	Yes	Yes
Database systems	Yes	Yes	Yes	Yes	Yes

We can see from the table that software testing and quality management is not yet a fully integrated part of computer science courses at all those universities which produce future IT managers, software engineers and analysts and who lack knowledge about shaping software quality and practical skills with tools for achieving it even today.

3. The survey

I explained the importance of software quality and how better educated and skilled employees play an important part in achieving it. The following survey was conducted to determine and event test the quality of testing teams in Czech software companies as any previous surveys had not done that.

An online questionnaire was created in order to perform the survey and it was published on the web http://studiekvality.vse.cz, where data was gathered over a period of 5 months. During that time, over 300 software companies of various sizes operating in the Czech Republic were asked to have someone fill out the questionnaire. It was preferable for the questionnaire to be filled out only once for each company by an employee involved in testing, or at least coming into contact with it every day. As the name of the employer was not entered anywhere, this preference had no way of being ensured. Also, information about conducting the survey was posted on several websites and discussion forums that deal with different areas of software development.

The questionnaire was anonymous, respondents filled in only the general characteristics of the company and their jobs. The questions were selected and formulated based on foreign publications, the author's knowledge and goals.

The questionnaire was divided into five parts:

1) Characteristics of the company you work for, and your job:

   This part was mandatory for all participants and served to sort the answers, for example, according to the job position that the participant holds, or according to the size of the company where he/she works (or they work).

2) Your approach to software quality:

---

[8] *Data from the 2011/2012 catalogues of university courses accessible on the university websites to the date 15.11.2011*

[9] *ČVUT – Czech Technical University in Prague*

[10] *VŠE – University of Economics in Prague*

[11] *CUNI – Charles University in Prague*

[12] *ČZU – Czech University of Agriculture in Prague*

[13] *VŠCHT – The Institute of Chemical Technology, Prague*

This part was mainly focused on how the participant perceives software quality. They were asked questions regarding their understanding of the basic principles of quality and testing.

3) Quality management in your company:

In this part there were questions that served to find out how testing is done in the company. What tests are done, who does the testing, whether they are using any tools.

4) The level of testing in the Czech Republic:

In this part there were questions about the availability of literature on testing and deficiencies in quality management in the Czech Republic.

5) Quality management on projects - part only for managers:

All questions in this part were optional. This part was related to finding out how much of the software development budget is spent on the testing process.

Given the scope of the survey it was expected that the results would be published gradually in articles focused on specific topics.

3.1 Characteristics of participants

In total, the questionnaire was completed by 84 participants who come into contact with testing and participate in creating software quality.

The participants were only employees of software companies, of which 48% came from companies with up to 100 employees and 52% from companies with over 100 employees. As the value of 100 divides participants in two almost equal parts, I will use it when comparing responses of employees coming from smaller and bigger firms. Table 4 provides an alternative distribution for better understating of varying sizes of companies in these two groups.

Tab. 4: Classification of participants according to the size of the employer [author]

Company size	Representation in %
Up to 40 employees	34.5 %
41 - 100 employees	13,5 %
101 - 300 employees	23.4 %
301 - 3000 employees	16.7 %
Over 3000 employees	11.9 %

Among the participants were 41 members of the test team, 16 project managers and 27 other employees, mostly in positions of developer or analyst. A breakdown of the individual positions represented by percent can be seen in Figure 1. More than two thirds of the participants can be considered to be skilled because they have worked and have come in contact with software testing and quality management for 2 years or more.

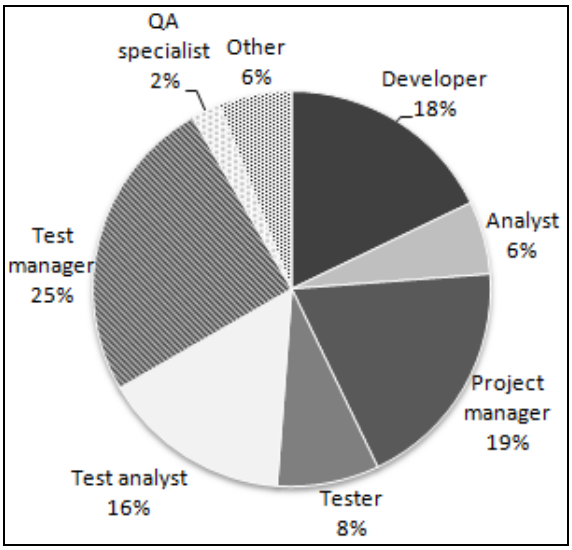

Fig. 1: Job position of research participants (author)

4. Survey results

The survey proved that even when software testers are skilled in finding defects and delivered software is usually satisfactory to users; many software testing teams have a very low level of understanding of the basic concepts related to the quality of the software. There is also low availability of controlled and structured training on quality management and testing for these employees. In following chapters it is explained in more detail.

4.1 Understanding the definitions of software testing and quality assurance

To find out how the testing is perceived, the questions and options listed in Table 4 have been selected. No exact wordings of definitions among the options were given on purpose, because the aim was not to choose a credible definition of testing, but to find out how much respondents understood testing.

Tab. 5: The definition of testing according to survey participants [author]

In your opinion, which of the following definitions of testing is the most correct one?	
a) Testing is the process of finding defects	9.5 %
b) Testing is the process of finding information about the state of quality	16.7 %
c) Testing is the process of checking the product against its specification	26.2 %
d) Testing is the process of ensuring quality	47.6 %

Options a), b) and c) correspond to different definitions. Option a) corresponds to the original concept of testing described in 1979 by Myers (Myers, 2004). Option c) corresponds to the definitions of Abrana and Moore (Abran et al., 2004) and Hailperna and Santhanama (Hailpern et al, 2002), although the definitions of the two sources differ on whether the product is to be checked only dynamically, or also statically. All the above definitions had been criticized and rejected by experts as too restrictive in the past. Testing is not always focused on finding bugs, but it may occasionally have the opposite goal: to find what works. Likewise, a product can be tested without the presence of the specification, although not as efficiently.

Option b) represents the current accepted definitions of quality, such as versions by Kaner or Bach (Bolton, 2009).

Unlike the previous option, option d) was added as a play on words, because inexperienced testers tend to confuse detection ("zjištění" in czech) with ensuring ("zajištění" in czech) when they recall a definition of testing.

As was observed, half of the respondents actually chose option d) many of them occupying the role of test manager. It is quite interesting, as testing alone cannot ensure quality.

As most participants tended to choose more complex instead of simple answers, one might conclude that they are aware that testing can have more roles or goals, and that it is part of a much more complex process. The outdated view of testing as a process of finding bugs was only chosen by a few respondents, but at the same time we can see that the perception of the role of testing among respondents differs significantly.

For a more detailed analysis of the perception of the role of testing it is necessary to take into account how participants see a software quality assurance process[14]. The correct definition is in option c).

Here, the vast majority of participants chose option c), even though participants from companies up to 100 employees also had a tendency to label option d). Interestingly, none stated that they do not know what the term exactly means, and rather chose the wrong answer.

Tab. 6: The definition of software quality assurance according to the survey participants (author)

In your opinion, which of the following definitions of Software Quality Assurance is the most correct one?	
a) I do not know what this term means	0 %
b) Software Quality Assurance is a just different name for testing	3.6 %
c) Software Quality Assurance is an approach to development processes, which seeks to prevent defects and to minimize the cost of their removal	65.5 %
d) Software Quality Assurance is a comprehensive approach to testing, which aims to ensure thorough testing of a product and the correct removal of defects	30.9 %

We can get other notable results by comparing respondents' answers to these two questions about the definitions of software testing and quality assurance. In total, 42.5% of those who say that testing is the process of ensuring quality, also claim that quality assurance is just a comprehensive approach to testing. The remaining 57.5% perceive quality assurance as a way to prevent errors and minimize the cost of their removal. If we merge the previous two answers, we get a quarter (27.4%) of all respondents claiming that testing is the process of preventing defects and minimizing quality costs. The existence of control is certainly a positive impact on prevention, but it is more a minor side effect than the purpose of testing.

From the observed results it is possible to deduce that a significant proportion of the employees who are constantly coming into contact with testing and who are actively involved in shaping quality, fail to properly explain the concepts of software testing and quality assurance.

4.2 Determining the strengths and weaknesses of testing techniques

To determine whether employees of software companies are able to correctly identify the strengths and weaknesses of testing techniques they use, participants were asked two similar questions.

They could choose from four proposed techniques which of them is the most effective in detecting defects and which of them is the most effective in verifying whether the product does what it should. The selection of techniques for both questions was the same:

a) Scenario testing
b) Free testing
c) Regression testing
d) Code review

[14] *Software Quality Assurance is defined as a process that ensures that the standards, processes and procedures used during development and maintenance of software are appropriate and properly respected. (NASA, 2004) Apart from testing it deals with the prevention of errors, implementing and improving business regulation and its main idea is that a good development process ensures a good result. (Sun and col., 2011)*

For an explanation of the following result analysis you can find brief descriptions of these techniques in table 7:

Tab. 7: Characterization of testing techniques [author]

Scenario testing – Testing described by detailed step by step scenarios with input data and expected results. Running these tests does not require any extensive knowledge of the tested system and is quite suitable for junior testers. The tester is spoon-fed, which is a benefit, but he is also limited by pre-described procedures which leads to testing only basic and simple scenarios. The tester doesn't actively seek defects, but just checks the functionality over the specification. This testing is systematic and reveals a lot of visible defects. However, these tests are expensive to create and maintain. In addition, we can conclude that the rate of finding errors with the repetition of the same steps will be lower than when the tester often changes the testing procedure in order to find more defects and performs an unexpected action.
Free testing – Testing in which the tester uses his or her knowledge and experience to search for defects. It is suitable for expert testers.In free testing, the tester only has one goal, to find the most serious defects. The cost of preparing and maintaining these tests is minimal, mostly comprising the salary costs of an experienced tester.
Regression testing – Testing that seeks to uncover new defects in previously tested or used functionality after changes have been made to a system. The probability that a particular test finds a defect is very small, but with a large number of tests the probability increases. The cost of doing regression testing varies; the most effective regression tests are automatic tests that are executed very often.
Code review – formal or informal inspection of the source code by persons or tools. The cost of a code review can be substantial, since it is a time-consuming activity that requires professionals with advanced knowledge of programming. An alternative could be a tool that searches for common mistakes, but its options are always limited. A code review allows the detection of the most of common errors, and moreover those defects that other techniques cannot detect.

Answers to the question, which technique is the most effective in verifying that the product does what it should do, did not bring any surprises: the most frequent one was scenario testing (66%), followed by regression testing (23%). These two techniques are useful for verifying if the product has been successfully implemented.

Selecting a technique which is the most effective in detecting defects was more difficult, with a clear difference in opinion between employees of smaller and larger firms. While many, especially those from companies up to 100 employees, identified free testing as the most effective in detecting defects, the overall winner was surprisingly once again scenario testing.

But scenario testing is, by its nature, very ineffective in detecting defects, as these tests are relatively slow, expensive and also focused on basic functionality where the probability of defects is generally lower.

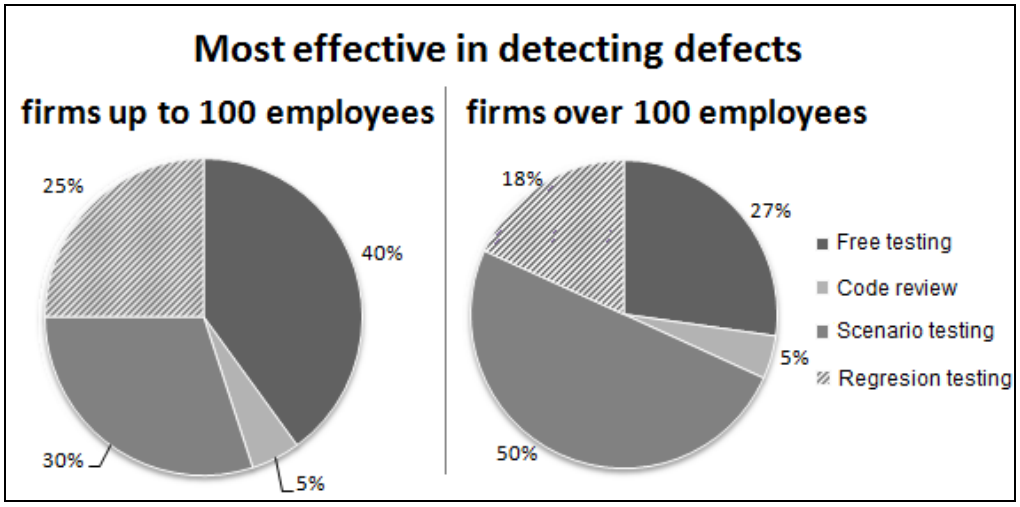

Fig. 2: Comparison of responses of employees of smaller and larger firms (author)

Given that the research also showed that scenario testing and free testing are the two most widely used techniques, it seems that participants do not fully understand the testing techniques they use and have problems identifying their strengths and weaknesses.

4.3 Education methods

To determine the predominant method of training, participants were asked how they gained their knowledge about testing. Respondents could only choose one option, the most predominant one in their education besides work experience.

Tab. 8: Source of knowledge (author)

Where did you gain knowledge about testing and other processes related to software quality management?	
a) From training courses	14.3 %
b) From books	17.9 %
c) From internet	46.4 %
d) From colleagues	20.2 %
e) Not interested in it	1.2 %

The survey showed that participants are actively interested in the area and are trying to learn. However they gain knowledge using mostly an unsystematic and uncontrolled access to information such as the Internet or colleagues. Such sources are very unreliable in terms of the origin and quality of information.

A relatively small proportion of the participants chose training as the main source of information, whether provided by the employer or in the form of higher education. It shows that participants have a very limited possibility of attending courses or that the courses do not provide sufficient education.

4.4 Quality of software testing in companies

Another question to answer was how participants perceive the quality of software testing in the company where they work. In this respect, the following two questions were asked:

- According to your opinion, is testing sufficient in your employer's company?
- What are the biggest shortcomings in software quality management in your employer's company?

Most of the participants evaluated testing in the company rather positively, as shown in the following table:

Tab. 9: View of testing in companies [author]

According to your opinion, is testing sufficient in your employer's company?	
a) Definitely yes	13.1 %
b) Probably yes	47.6 %
c) I do not know	3.6 %
d) Probably not	25.0 %
e) Definitely not	10.7 %

More than half of participants consider testing in their companies to be sufficient, or probably sufficient. This positive view can have many causes, from high-level testing to underestimating the testing process. Agreement in opinions on quality between the staff and the company is expected and desirable. So it can be more striking that a third of the participants who are involved in shaping quality, do find testing in their company to be insufficient. This result indicates dissatisfaction of some employees with the level of quality that the firm produces, and the discrepancy of opinions on quality between these employees and their managers who decide how to invest in quality.

The second question regarding the biggest shortcomings in software quality management in companies was optional and less than half of the participants responded to it. The most frequently mentioned were an underestimation of quality, a lack of investment and a lack of know-how along with a fractured methodology.

A large proportion of the participants thus feel that quality is under-valued in their company. It would be appropriate to ask here whether these employees tried to do something about it and how successful they were in this activity. These questions were not part of the original questionnaire, but may be subject to further research to find more deeply rooted problems.

4.5 Investments in software testing

Software testing is an important part of development and quality assessment, which, according to Boris Beizer, may consume more than fifty percent of the total development budget (cited in Merkel et al., 2010). The survey results in software companies operating in the Czech Republic confirm this; however, such budgets are very rare.

Questions related to the size of investments were optional and in the part that was marked as intended only for managers. The survey results show that both smaller and larger firms invest a relatively small portion of the project budget to testing. The graph in Figure 4 shows how respondents answered the question, what part of the whole development budget (on their last project) went to testing and test preparation. The distribution of responses for both smaller and larger firms is similar, even though they differ in the most frequent response and in whether their firms are willing to pay more than half of the budget for testing.

In this case, respondents were also divided according to the nature of the project for which the budget was discussed. As expected, the modification of an existing product generally has a smaller budget than the development of a completely new product.

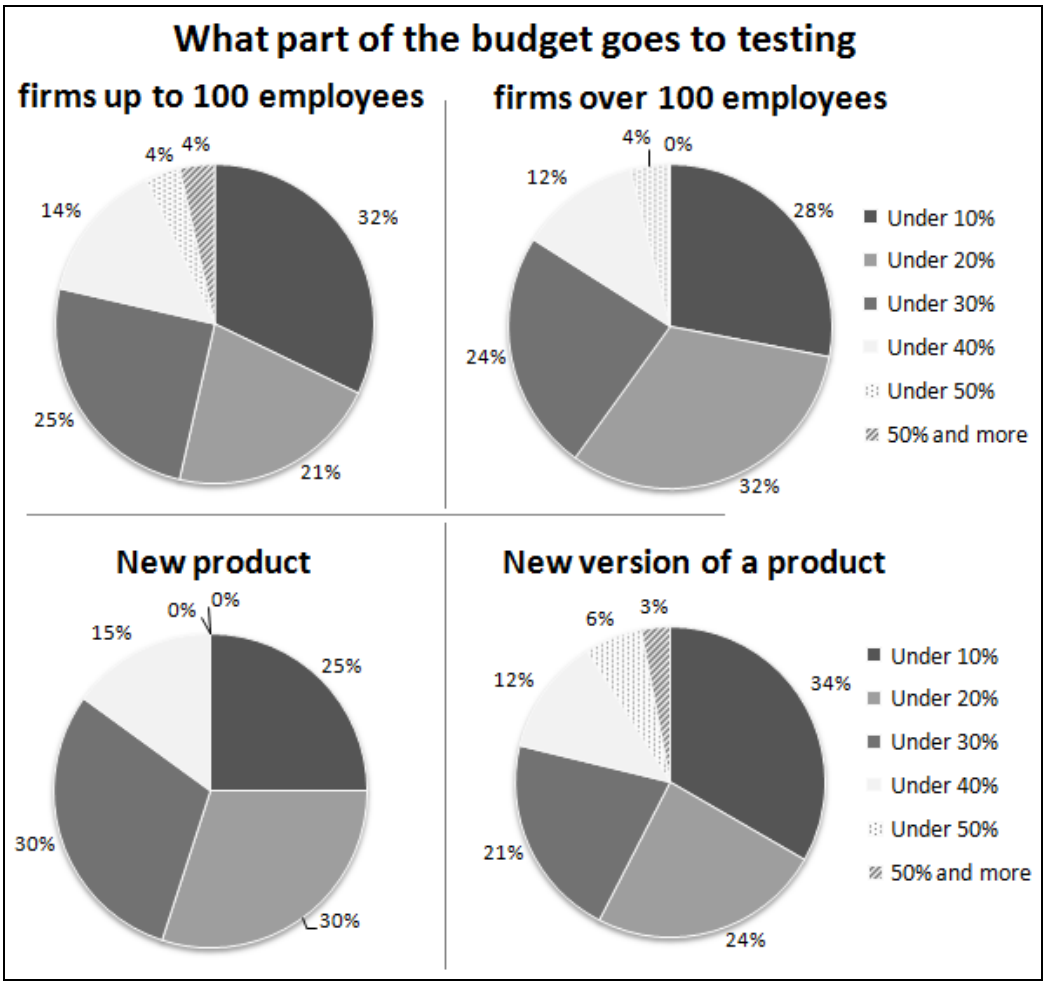

Fig. 3: What part of the entire development budget went to testing and test preparation?

We can see how quickly the mentioned budget was spent in the following graph (Figure 5). As we see, more than a half (56%) of larger companies starts with the testing process from the very beginning of the project, while smaller companies prefer to start with testing only when the executable version is available. Since the period before the existence of the executable version of the process of testing involves planning and preparation for tests, it can be safely assumed that smaller companies often have to minimize preparation time and test planning. This is possible by selecting smaller projects and testing techniques that do not require detailed documentation.

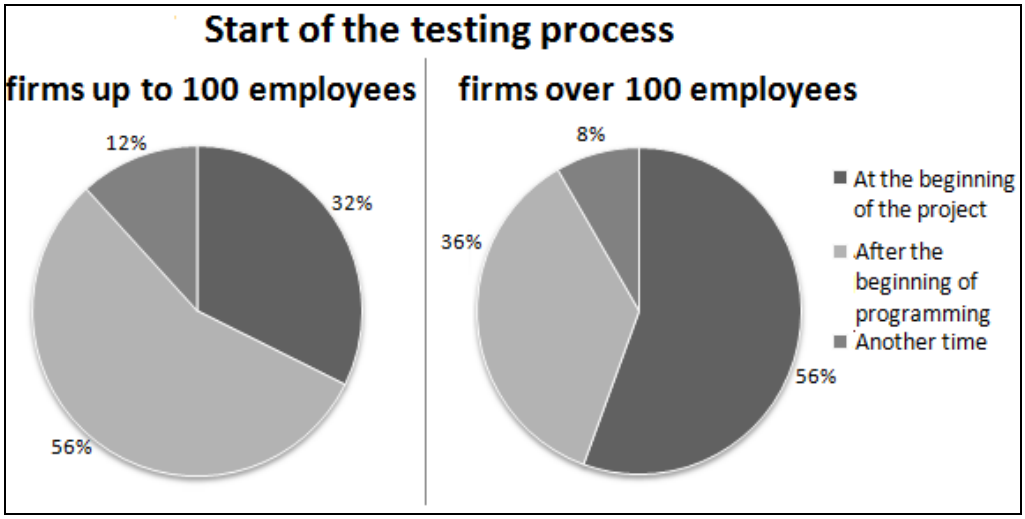

Fig. 4: Comparison of when companies begin the testing process

The testing process begins before the creation of a new executable program version in all major methodologies such as RUP[15], Extreme Programming[16], the V-model[17], and more. Early and continuous quality control is one of the recognized six best practices[18].

The main reason why it is the most effective to begin in advance of the actual programming is the complexity of test preparation and that the early detection of defects minimizes the cost and reduces the number of problems with budget overrunning and time consumption.

Several international studies have shown that a significant number - around 50% - of all found defects occur in the requirement phase (Sheldon et al., 1992) reported a 41% (Nakashima et al., 1999) reported a 50%, (Budd, 1984) indicates 56%).

At the same time, the later the defects are discovered, the more expensive it is to fix them. This growing trend can be up to exponential (Vrbka, 2008) (Patton, 2002), especially when we include not only the price of repairs but also other related costs such as losses associated with the existence of the defect.

From this we can see that although the budget for testing is relatively small, money is often inefficiently allocated to fixing defects and repeated retesting and not on defect prevention.

5. Conclusion

This research has been focused on the level of software testing and quality management in software companies developing software in the Czech Republic, where quality, efficiency and reliability are important aspects of development of competitiveness.

[15] *Rational Unified Process (RUP) is a commercial traditional methodology promoted by IBM.*

[16] *Extreme Programming is one of the significant agile methodologies.*

[17] *V-model – one of the project life cycle models, used in a number of corporate methodologies.*

[18] *Best practices – set of six practices that were adopted by a number of methodologies. They were originally identified as the common properties of solutions applied by successful firms in an effort to prevent problems in software development.*

In the previous chapters the five questions posed at the beginning of the article were discussed and answered:

- How do employees understand the concepts of software testing and quality assurance?
- Are they able to correctly identify the strengths and weaknesses of the test techniques used?
- Where do workers draw on their knowledge of software quality management?
- How do they perceive the quality of testing in the company where they work?
- How big is the budget for testing in relation to the budget for the entire development?

It was found that the employees of Czech software companies who are actively involved in shaping quality have a very low level of understanding of the basic concepts related to the quality of the software. There is also low availability of controlled and structured training on quality management and testing for these employees. This is not apparent in most cases in perceived software quality as customers are accustomed to software defects and are content when the software is able to serve its purpose. But as it was explained the quality of testing and development teams impacts the efficiency so it takes more money and time to develop the software.

An improvement in efficiency of software development depends greatly on a management, namely a project management and a quality management, it is therefore important that they are informed of scientific discoveries in this field done by universities.

As it is in the Czech Republic nowadays, the community of project managers and quality managers has separate conferences and literature from university experts researching the same area.

To improve this situation I would propose the implementation of following steps:

- To extend the number of academic and vocational courses with particular emphasis on preventive quality management methods and to explain to managers the economic aspects of the investment in quality of the software development.
- To extend the number of practical courses aimed at the use of theoretical knowledge about software quality management in practice.
- To expand the range of training for employees in companies actively involved in shaping the quality of software, or motivating employees to educate themselves with a recommended list of quality resources.
- To create joint conferences for both professional and university researchers to share their knowledge and experiences.

Of course, also testing and quality management teams should be motivated by their employers to gain and apply the knowledge from these courses.

6. Bibliography

Abran, A., Moore, J., 2004: *SWEBOK, Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, Version.
http://www.inf.ed.ac.uk/teaching/courses/seoc/2006_2007/resources/SWEBOK_Guide_2004.pdf

Bolton, M., 2009: *Insource or Outsource Testing: Understanding Your Context*. STAREAST 2009.
http://www.choucairtesting.com/portals/2/Documents/StarEast2009/Presentations/W11.pdf

Budde, R., 1984: *Approaches to Prototyping* . Berlin : Springer-Verlag, ISBN 0387134905.

Desai, A., Shah, S., 2011: *Knowledge management and software testing.* ICWET '11: Proceedings of the International Conference & Workshop on Emerging Trends in Technology. Hailpern, B., Santhanam, P., 2002: *Software debugging, testing and verification*. IBM Systems Journal, roč. 41, č. 1, www.research.ibm.com/journal/sj/411/hailpern.pdf

Merkel, R., Kanij, T., 2010: *Does the Individual Matter in Software Testing?* Technical Report: Center for Software Analysis and Testing Technical Report 2010-001, Faculty of Information and Communication Technologies, Swinburne University of Technology,.
http://www.swinburne.edu.au/ict/research/sat/technicalReports/TC2010-001.pdf

Myers, Glenford J., 2004: *The Art of Software Testing*, Second Edition, Hoboken, New Jersey: John Wiley & Sons, Inc., http://www.51testing.com/N_download/lib/TestingTechDL/ArtofSoftwareTesting.pdf

Nakashima, T., et al., 1999: *Analysis of software bug causes and its prevention.* In *Information and Software Technology*. Volume 41, Number 15. Elsevier Science Publishing Company, Inc.,. p. 1059-1068. ISSN 0950-5849.

National Aeronautics and Space Administration, 2004: [online]. *NASA-STD 8739.8 Standard for Software Assurance,* http://www.hq.nasa.gov/office/codeq/doctree/87398.pdf

Iivonen, J., 2009: *Identifying and Characterizing Highly Performing Testers – A Case Study in Three Software Product Companies.* Diploma thesis, Faculty of Information and Natural Sciences, Helsinki University of Technology, http://www.soberit.hut.fi/espa/publications/MTI.pdf

Jones, C., Bonsignour, O., 2011: *The Economics of Software Quality*. Addison-Wesley Professional.. ISBN: 978-0132582209.

Patton, Ron, 2002: *Testování softwaru*. Brno: Computer Press. ISBN: 80-7226-636-5

Sheldon, F., et al., 1992: *Reliability Measurement from Theory to Practice*. *IEEE Software* [online]. July 1992. http://www.cs.utexas.edu/~hestness/papers/sheldon-softwarereli.pdf

Slaughter, S., Harter, D., Krishnan, M., 1998: *Evaluating the Cost of Software Quality*. Communications of the ACM, Vol. 41, No. 8,. http://www.cse.buffalo.edu/~hungngo/SCE-Papers/p67-slaughter.pdf

Sun, H., Jiang, N., 2011: *The Hierarchy of Software Testing Knowledge.* Third International Workshop on Education Technology and Computer Science, IEEE 2011.

Voříšek, J. et al., 2010: *Principy a modely řízení podnikové infomatiky*. Vysoká škola ekonomická v Praze,. ISBN: 978-80-245-1440-6

Vrbka, Z., 2008: *Zpráva o stavu a potřebách v oblasti testování informačních systémů*. Brno,. Technical report. , Fakulta informatiky Masarykova univerzita. http://kirlab.fi.muni.cz/documents/RR_2008_Zprava_o_stavu_a_potrebach_v_oblasti_testovani_informacnich_systemu.pdf

JEL Classification: A20, C80, Y80.

This article should be cited as:

Havlickova, A., 2012: The perception of software quality and testing in Czech software companies. *Journal of Systems Integration* 3 (4), pp. 15 - 28. [Online] Available at: http://www.si-journal.org. ISSN: 1804-2724