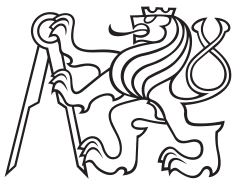
Bachelor's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Economics, Management and Humanities

Analysis of Current Practice in Estimating Software Testing Efforts

Petr Heřmanský
Software technology and management

May 2016
Supervisor: Ing. Miroslav Bureš, PhD.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra ekonomiky, manažerství a humanitních věd

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Heřmanský Petr**

Studijní program: Softwarové technologie a management
Obor: Manažerská informatika

Název tématu:

Analýza současné praxe v odhadování pracnosti testování software

Pokyny pro vypracování:
Proveďte rešerši aktuálně používaných technik pro odhadování pracnosti testování software. Proveďte strukturovaný průzkum mezi vybranými analytiky a manažery testování z praxe, jehož výsledek bude zhodnocení, jaké techniky jsou nejčastěji používány, jaká je přesnost dosahovaných odhadů, jaké oblasti testování se daří a naopak nedaří odhadovat a související aspekty podle zadání školitele. Do průzkumu zahrňte minimálně 10 specialistů z praxe. Analyzované techniky srovnejte podle klíčových faktorů, jako jsou přesnost odhadu, učící křivka, pracnost odhadu a dalších podle zadání školitele.

Seznam odborné literatury:
1. Koomen, T., et al. TMap Next: for result-driven testing. UTN Publishers, 2006.
2. Abran, A. Software project estimation: the fundamentals for providng high quality information to decision makers. Wiley, 2015.

Vedoucí bakalářské práce: Ing. Miroslav Bureš, Ph.D

Platnost zadání: do konce letního semestru 2016/2017

L.S.

Prof.Ing. Jaroslav Knápek, CSc.
vedoucí katedry

Prof.Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10.2.2016

Acknowledgement / Declaration

I thank to Ing. Miroslav Bureš, PhD. the supervisor of the bachelor thesis for his valuable advice and suggestions, which were very helpful. Also I thank to my girlfriend and family for support during studies and my colleagues for factual advices.

I declare, that I have done assigned the bachelor thesis alone led by supervisor. I used only literature, that is listed in work. Furthermore I declare, that I have no objections against lending or making public of my bachelor thesis or it is part with agreement of department.

In Prague 24. 5. 2016

.....

Abstrakt / Abstract

Tato bakalářská práce je zaměřená na analýzu současné praxe v odhadování pracnosti testů v softwarovém projektu v České republice. Práce je rozdělena do dvou části.

V první části se zaměřuji na odhady testování. Definuji pojem odhad, co je odhad pracnosti v testování a čím se zabývá a v závěru uvádím rady, které je dobré dodržovat při vytváření odhadů, především při odhadech pracnosti testování. Dále jsou zde popsány základní techniky pro vytváření odhadů, které jsou rozděleny do dvou sekcí a to založené na zkušenostech a založené na modelu.

Druhá část práce se skládá z analýzy současné praxe v odhadování pracnosti testování softwarového projektu v České republice, kterou jsem vytvořil na základě dat získaných z dotazníku, který je součástí práce jako Příloha A. Zákxladní oblasti na které jsem se v dotazníku zaměřil jsou jaké techniky se používají, jaká je přesnost vytvořených odhadů, které oblasti v testování se daří odhadnout a které hůře, odchylky v odhadování pracnosti a podle čeho se rozhoduje jakou techniku pro odhad pracnosti zvolit. Odpovědi na otázky mi poskytli test manažeři, analytici a vedoucí týmů s průměrnou praxí více než 5 let. Je zde i stručný popis jak odhadovat náklady spojené s odhadnutou pracností a vybrané kognitivní chyby v úsudku s dopady na testování.

Klíčová slova: Testování softwaru, odhad, pracnost, odhadovací techniky.

Překlad titulu: Analýza současné praxe v odhadování pracnosti testování software

This bachelor thesis is aimed at the analysis of the current practice in estimating software testing efforts in the Czech Republic. The work is separated into two parts.

The first part of my thesis focuses on estimations in a general manner general estimations, not only on the testing. It defines the term estimation, presents the purpose of the estimation and at the end its the best practices which should be respected in these estimations, especially in the effort estimation. Furthermore, there is a selection of estimation techniques. They are categorised in the two groups “Experienced-based techniques” and “Model-based techniques”.

The second part contains an analysis of the current practice in the estimation of software testing efforts in the Czech Republic. This analysis is based on data obtained from a survey, listed in Appendix A. Areas tackled by the survey are the techniques used for estimation, the accuracy of the estimations, the parts of the testing providing more accurate estimation and those providing less accurate ones, the deviation in the effort estimation and the decisions as to the most adequate and appropriate estimation technique. The answers to my questions were provided by Test Managers, Test Analyst and Test Leaders, detaining more than 5 years of experience in the domain. Also included is a basic description of cost-estimations related to the estimated efforts and selected cognitive biases and their impact on the tests.

Keywords: Estimation, effort, software testing, estimation techniques.

Contents /

Tables / Figures

Chapter 1
Introduction

Recently software has become the most expensive component in information technology. And a perfect thing, which we want to achieve in the software development, cannot go alive without a proper test cycle to try to catch all bugs which comes with developing. So here is the questions: "why do we do testing? What testing can do?" well, there is one good fact: "Today a typical application of 1,000 Function Points will contain 5,000 defects and deliver about 750 defects to customers using the normal waterfall approach. It is theoretically possible therefore to cut the defect potential down to 2,500 and deliver only 25 defects to customers using state-of-the-art defect prevention and removal methods." available from [1, p.189]. However we can not avoid all bugs but this implies a potential defect deliver reduction of 97%, which is awesome number. Therefore having a well prepared estimation is very important area, because testing is one of the best way to improve quality of the software. Quality, among other things, is key point for improving competitiveness. And also according to the estimation, we can decide about the destiny of our software project.

1.1 Aim of the thesis

This bachelor thesis is aimed on software testing effort estimation and its practice in the Czech Republic. Practice will be monitoring with a survey and it is covering mainly Prague because big corporates have headquarters there. Big corporates have usually big projects and this projects and its effort estimation for testing giving me a quality data. The bachelor thesis is divided into two section.

In the first section is the definition of term estimation and what is the purpose of estimation. Then there is described what is effort, what is test estimation effort and at the end there are listed the best practices which are good to respect in effort estimation. Further, there is description of the basic estimation techniques which are divided into two sections namely on Experienced-based and Model-based techniques.

The second section comes from structured exploration among selected Test Managers, Test Analyst and Test Team Leaders. They have been asked to complete a questionnaire, which is in the Appendix A. The result of this survey should be overview of what is the practice like in the Czech Republic, which techniques are used for estimations, what is the accuracy of estimations, which parts of testing leads to more accurate estimation and which one to less accurate, deviation in effort estimation and decisions what technique to use for an estimation.

1.2 Assumptions and limitation of the work

The assumption is to study and understand selected estimation techniques. Then try to reproduce this technique in the sense of effort estimation. Another assumption is to collect quality data from test managers, test leaders or test analyst to give overall overview of current practice in estimating software testing efforts in the Czech Republic.

The limitation of the work is specified range work. There is not enough space for describing every technique to detail, therefore techniques are described briefly with the most important and basic information. For more study about single techniques is recommended to see one of the books which are in the Appendix. [2–5]

1.3 Target group

The main target group of this work are people who are drown into software testing world and want to know more about current practice in estimating software testing efforts in the Czech Republic, but not only this. It is also overview and simple description of the most using techniques for estimation effort.

1.4 Motivation

I have chosen this topic because I am currently working in company engaged in software testing. My specialization is automation testing but I wanted to extend my knowledge to be more complex about testing generally, especially in estimation, because it is the the point of the project. I found a lot of material what techniques are using for estimation generally, but there was no publication about current practice in estimating software testing efforts in the Czech Republic. Motivation is to collect quality data to draw a conclusion about current situation of estimations. Also brings basic overview of the most used techniques for estimation effort.

Chapter 2
Basics of effort estimation

Estimation as a management activity is to create approximate estimate of cost, schedule and scope of the project. In project management it is called triple constraint or golden triangle. To make a good estimate requires experience and expertize to convert qualitative measures to quantitative form. It is important for success of the project. Factors like project size, amount of risk, availability of test environment, availability of resources and more are affecting the accuracy of estimation. The best estimate should:

- Represent collective estimation of experienced practitioners.
- Provide specific and detailed documentation of the costs, resources, scope and people involved.
- Cost, effort and duration of each activity estimated.

In 1883 glorious Lord Kelvin says: "If you can not measure it, you cannot improve it." [1]) This give us answer why it is important to measure time for testing effort from beginning and then the chance for our estimation is almost perfect. Then we can adjust our estimation down or up by counted time.

2.1 Triple constraint in estimations

Sometimes it is called golden triangle or iron triangle. Each constraint cost, schedule and scope of the project with quality as the central theme plays a key role in developing project. Each constraints forms the peak, so if we summarize it, triangle says: [2])

- Project must be delivered within cost.
- Project must be delivered on time.
- Project must agreed the scope.
- Project must meet customer quality requirements.

[1]) `http://zapatopi.net/kelvin/quotes/`
[2]) `https://www.projectsmart.co.uk/understanding-the-project-management-triple-constraint.php`

Figure 2.1. Triple constraint or sometimes known as golden triangle. [1])

1) *Cost*: Every project has finite budget for realization.
2) *Schedule*: Every project has deadline for delivery the product. As the saying goes "Time is money", especially commodity as time slips away too easily.
3) *Scope*: If scope of the project is either not fully defined or understood then many project fails on this constraint.

The sense of the triangle is that value of one factor cannot be changed without influencing other factors. The difficulty of satisfying expectations for all three constraints is sometimes expressed as *pick two*. This means that in any set of three desired qualities, only two can be delivered.

For example: If we reduce project's time then we have to either increase cost (in testing it is usually increasing the number of testers) or reduce scope of the testing (not to run all tests usually according to priority of the test).

2.2 Estimating effort in general

Drawing up an estimate effort to executing tests in project is surrounded by uncertainty. So it is recommended to spare it in a series of steps.

Firstly, it is good to draw up a draft version of the estimate. There are some questions what we have to ask before estimation. For example, why is the estimate being made? How precise must be the estimate to give enough of confidence? Answers to these question gives manager or leader information, how much effort must make in creating the estimate. So we can say that it is not wise to invest a lot of time and energy to create an estimate, that is only overview, how much time takes converting manual tests to an automated version. On the other side, if client wants a very precise estimate of performance tests then we must collect detailed data as much as it is possible. with this idea, manager must decide if the time required for drawing up an estimate is worth.

Now is time to choose technique for estimation. Depends on this technique, it will take more or less time to collect needed data. Trying to manage the uncertainty factor, it is recommended to use two or more techniques side by side. Then put the results

[1]) `https://projectmanagers.org/triple-constraint-project-management/`

from these estimates together and crosscheck them with stakeholders and bring one final estimate. Important thing is that the manager or leader must always keep the stakeholders and team informed of the chosen estimation technique, the assumptions, preconditions and decisions that are being used or made. Then the stakeholders can react on questionable assumption or preconditions have been made.

Secondly it is the reviewing and refining the draft version. All sides that are directly involved, such as the client side and the testers, should be invited to refine and review the estimate. The goal of this is to to make single estimate that everyone accept as a feasible estimate. However, it will always be multiply by factor of initiation of any test project. This is simply because for a good estimation, to be accurate, we need informations as much as possible. This is something what is lacking at the start of any *new* project. This graph below 2.2 called "Cone of Uncertainty" demonstrates, how difficult is an initial estimate no matter how qualified is the person who doing the estimation. What this graph shows is that even when all requirements analysis has been completed, our estimate can only be at the most 50% accurate.[1])

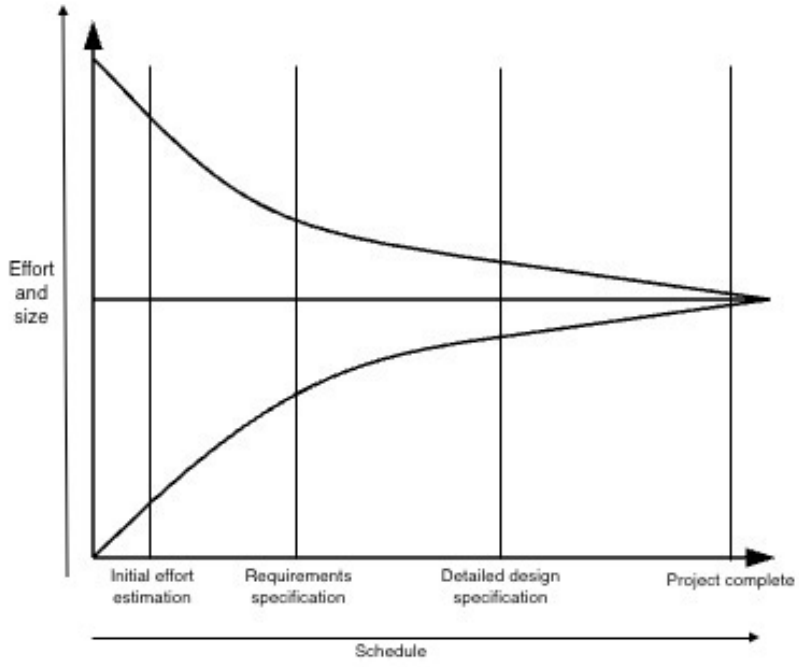


Figure 2.2. Cone of uncertainty. [2])

Thirdly the estimate now can be delivered to all interested parties, such as stakeholders declared in the test strategy. Everyone is then aware of the estimation, strategy and assumptions. In formal organization is necessary to have the estimate approved by the client (or may be better the one who pays for the project). On the other hand, in informal organization will be verbal approved satisfactory. If the estimate is different

[1]) This can be different when we estimating effort for project, that is really similar to the project before, mainly with stable of testing environment, people who works on the similar project, etc. Then the estimate may be more accurate and the Cone can look much more narrow at the start.
[2]) Taken from MIT open courseware.

from the expectations of the client then a discussion can be initiated to control and reorganize the scope and depth of the tests.

Usually management will hope for reductions in the estimated cost or duration or both of this. Ideally, the final test effort estimate represents the best possible balance of organizational and test project goals in the area of quality, schedule and budget.

2.3 Test effort estimation

In my thesis I am dealing with testing effort estimates. It is not easy to draw up an effort estimate at the vary begging of the test project, since there is only general or not concrete data at the time. There is a choice of test estimation techniques described in chapter 3. An effort estimate describes boundaries in sense of time and money usually in MD. [1]) However, it is not simple to draw up an estimate which is not too tight, but also no too liberal. Too tight estimate creates unrealistic expectations, usually the scope is too big for allocated time on the basis of the effort what we estimated, while one that is too liberal leaves too much spaces for extra, unplanned tasks. It is almost always based on some assumptions (e.g. the number of defects or the delivery from developers). All assumptions made by during estimation should be documented. There are some raw testing definitions [2]:

- *Test estimation:* Test Estimation is the estimation of the testing size, testing effort, testing cost and testing schedule for a specified software testing project in a specified environment using defined methods, tools and techniques.
- *Testing size:* The amount (quantity) of testing that needs to be carried out. Sometimes this may not be estimated especially in Embedded Testing (that is, testing is embedded in the software development activity itself) and in cases where it is not necessary.
- *Testing effort:* The amount of effort in either person days or person hours necessary for conducting the tests.
- *Testing cost:* The expenses necessary for testing, including the expense towards human effort.
- *Testing schedule:* The duration in calendar days or months that is necessary for conducting the tests.

Test effort estimation is a skill required of test manager or test leader. However, this skill is not easy to learn quickly. It requires understanding all project test phases and lot of practice. Estimate often comes from historical data or similar project what we have done before. There is various estimation techniques, choosing the right ones in particular, requires experience. By *TMap* [3] „estimation can be made at a number of levels.“ From the top it is MTP estimate, estimate per test level, estimate per test phase, estimate per test activity.

[1]) Man-days

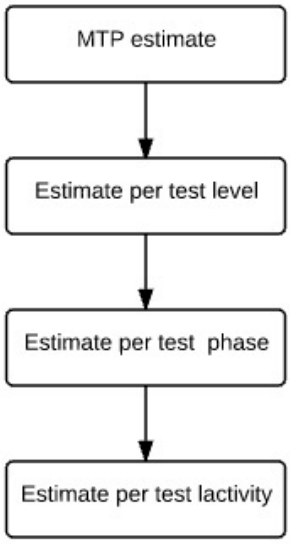


Figure 2.3. Estimation levels.[1])

The success of a test project effort estimation depends on variables that are almost impossible to influence: [5]

- The testability of the information system.
- The time required by the developers to solve the issues.
- The availability of the test environment.
- The stability of the test environment.

2.3.1 Factors before test effort estimation

Test estimation effort should consider all factors that can influence effort and duration of the testing activities. The list of these two factors is taken from [6] and it can be specific for different projects:

- *Required quality level of the system.*
- *Size of the system to be tested*: There is a direct proportion. Time consumed on testing depends how large is a project. In some project, it is possible to know about size of the project in terms of function points.
- *Types of testing required*: Sometimes, it is important to consider using multiple types of testing on system. For example, we need instead of functional testing perform load testing to see delay of application or system.
- *Material factors*: How extensively test automation and tools will be used, availability of test and developers environment(s), ease or difficulty of acquiring test data and quality of project documentation.
- *People factors*: Including experience and competence of managers and technical leaders, skills attitude in the test team, stability and relationship of the test team, test and debugging environment support and much more.
- *Complexity*: Complexity of the process, technology and organization.
- *Assimilation or development*: New tools, technology, processes, techniques, or a large quantity of testware.

[1]) Taken from [3].

- *Complex timing of component arrival*: Especially for integration testing and test development.

Scripted or exploratory testing: It may be feasible to only execute test cases which are already write or do exploratory testing or do both. If we are interested in scripting testing and do not have test cases, we should estimate time for creating and maintaining them. Also scripted tests usually requires prepared data. We should estimate effort for preparation the data. But if we use scripting or include automation testing well, we can save a lot of time in the future for manual testers, who can have more time to refine the test cases and test suites to have them up to date.

Test cycles: Test cycle is complete round of test (build verification test suites followed by attempted execution of all tests in the test suits followed by entering all find bugs into tracking system). In practice, one test cycle is not sufficient, if yes, testers would lose their job.

Quality of the software delivered for testing. Quality of the software delivered for testing is one of the most important factor that test managers should consider in their estimation. It is related with work of developers. If the developers embraced with best practices which means they are using automated unit testing and continuous integration, then as many as 50% of defects can be removed before delivery to the test environment.

2.4 Best practices in testing effort estimation

According to [5], consequences of all variables above are not immediately apparent. It is useful and recommended to split the estimate effort into three phases: Test project preparation, Test project execution and Test project closure.

- *Test project preparation*
  It is much easier to estimate testing effort for this phase of project which involves a quick scan; a risk analysis and a test strategy; a budget and a plan; a test plan and a test organization, including meeting and reporting structure then for execution tests, because it is less dependent on other parties.
- *Test project execution*
  Estimate effort for execution of the tests is much more difficult. Testers have to designed the tests and also execute them. We do not know how good will be documentation by which test cases are done. Testers also find issues that are related to developed software and it is not possible to estimate in advance how many issues will be found or how much time the developers require to bring the system up to the required quality. And, of course, more issues can lead to more test runs then were expected.
- *Test project closure*
  How much closure influencing the estimate depends what follows from the project. If the maintenance has to be set up within the test project, it means extra effort for maintaining and therefore extra budget requirements.

Working on various project helps with accuracy of estimation for test cycle. The main thing is that estimation should be realistic and accurate as much as possible. There is some general tips to raise the accuracy of the estimation:

- *Add buffer time*: Many unpredictable things can happen to our project so every estimation should have included some buffer time. It may delay the project if there

is some leave of team member (typically long leave). Resource planning in estimation plays key role. The availability of resources will help to make sure that the estimation is realistic. The estimation should consider some fixed number for resources and estimation should be re-visited and updated accordingly.

- *Consider bug life cycle*: In estimation we have to include a bug cycle. The actual cycle may take more days then estimated. This depends on stability of build. If build is not stable, developers need more time to repair bugs. It can extend the time for test cycle very easily.
- *Use the past experience reference*: Experiences from past project play vital role in preparing estimation of current project. If there is some similarity with project what we have already estimated, we can use estimation from there. We can analyze how the previous estimates were good and how much they helped in deliver of final product.
- *Stick to our estimation*: Estimation is just estimate because things can go wrong. In early stages of project, we should re-visit the estimations and make modification if needed. We should not extended our estimation after we release final version, unless there are major change requests in environment.
- *Team spirit*: If we know strength and weaknesses of our team, our estimation will be more accurate. We have to consider that same resources may not bring the same productivity. Some people from our team can test faster compare to other.
- *Are we going to perform automation or load testing?* If we want to include performance or automation tests into our plan then we need to put considerable time on it. These estimations should be considered differently.

2.5 Conclusion

Testing effort estimate can be based only on the information at the time it is prepared. At the start of the project, these informations are very limited or information can change over time. In order to keep the accuracy, estimates should be reviewed and updated to reflect new or changed information whenever the information being available.

Chapter 3
Estimation techniques

Basically estimation can be done either with top-down or bottom-up approach.

Top-down approach (see Figure 3.1) is based on high-level analysis, for example, starting with a proportional of the total project effort. This approach is suitable if there is little information available, for example on the start of the project. The only information might be the goal of the test project, overall information about requirements or a general plan of the development. Base on these information an initial estimate can be created. Using this method can only lead to initial estimate of a test project. Then the test manager must plan to review the estimate at certain points and if it necessary, adjust it. This method has some weak points which are:

- Technically complex steps, for example, with setting new tools that are not stable in environment, in the test project may be overlooked. These issues can increase total budget required.
- Testers, who are usually the most familiar with project, are not involved into estimation of hours required.

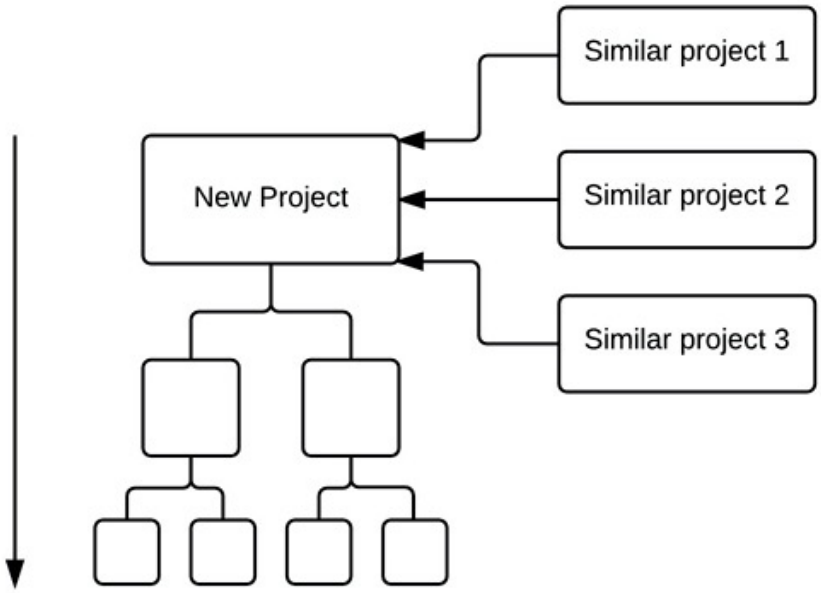


Figure 3.1. Schema of top-down estimation.[1])

[1]) Source: Taken and redesign from [5].

While *bottom-up approach* is based on detailed information and estimations of smaller individual tasks in project. Tasks necessary per test phase are detailed described. It is based on the technique WBS [1]), which is described in section 3.4. It is less suitable approach for beginning of the most project, because it requires detailed information to be successful. And as we know, it is not possible to have all information at the beginning. However, it is possible to adjust the estimate during the project with more detailed information being available. If certain tasks are not ready yet, it can lead to wrong assumption, usually too tight. To avoid this, we can extend estimate by leaving extra time for the uncertain task. Because of this technique is based on estimations of smaller tasks it will reflects in accuracy. This method has also its weaknesses:

- It takes more time to draw up a bottom-up estimation. The time must be planned for.
- It is little bit risky, because some certain overall tasks may be overlooked, for example test management or configuration management. For this type of estimation is quite useful to make a checklist.

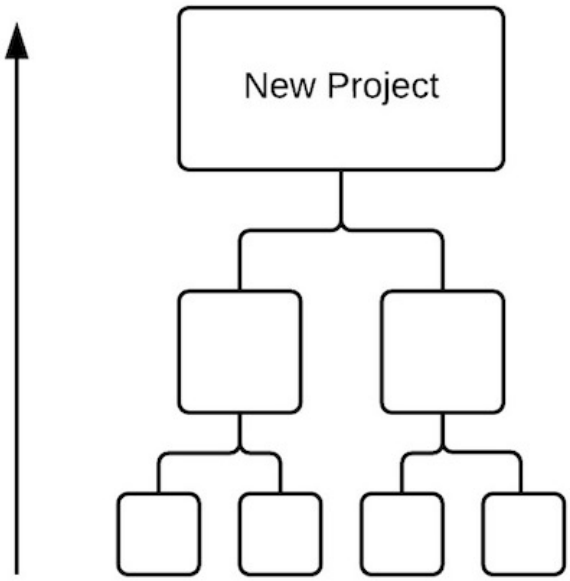


Figure 3.2. Schema of bottom-up estimation.[2])

Estimation techniques can be categorized in different ways. However this does not mean that some techniques belongs only to one category. According to ISTQB [3]) I will categorize techniques in two big categories, which are 'experience-based techniques' and 'model-based techniques'. But nowhere is written that techniques which are in the experienced based bucket cannot be in the model bucket as well. For example experienced based techniques do not calculate solely on people's experienced but may include other sources of information.

[1]) WBS - Work Breakdown Structure
[2]) Taken and redesign from [5].
[3]) International software testing qualifications board — www.istqb.org

I have chosen some basic techniques for further description:

- Experienced-based
  - Intuition, guesses
  - Expert estimation
  - Three point estimation
  - WBS
  - Extrapolation previous data
  - Team estimation sessions (Planning poker, Wideband delphi)
- Model-based
  - Estimation based on ratios
  - Using percentage
  - Metrics driven models (Number of test iteration)
  - Test point analysis

Experience-based estimation techniques

3.1 Intuition, guesses

This technique is also known as 'finger in the air'. It is not formal technique but this approach is commonly used. It give us quickly estimation with a little or practically no preparatory work. Maybe this does not sound like a good approach but there are situations in which we can use it. For example, when experienced test manager or leader estimating maintenance release he or she may be able to make a qualified estimate and based on experience of previous maintenance releases of the system. This works because the situation is consistent. The problem comes when one or more aspects of testing activity are different from expectations from before. Without time to investigate the detail of the differences, the estimator's assumption may not be correct.

Where is this approach not suitable? For example, when comes new technology or feature to our application. This means changes for testing personnel and may other aspect will introduce more risk. This will bring uncertainty to the guesswork that could reduce value of the estimate.

This technique can work well in stable and consistence situations. However in most of the cases it is best to avoid it, especially when new feature or technology is going to be deployed. If we are forced to make this estimation ("I need an estimate now, just give me your best guess") these tips may be find useful:

(i) Give a range values ("It is likely to take between 2 to 4 weeks.").
(ii) Add 'confidence rating' to the estimate ("It is likely to take 3 weeks, but I am only 20 %confident about that").
(iii) Try to delay the estimate. We can make more investigation and perhaps use other estimation techniques to accurate the estimate.

3.2 Expert estimation

This technique, sometimes called expert judgment, is reasonable to use when the environment is complex.

It is almost sure that someone with a lot of knowledge and experience with the information system or system what we want to test can obviously provide an important contribution in the estimation of test project. Explorers or people who are not so familiar with the environment can faces the consequences of variations from past project, such as consequences of using new technology or new interfaces. Here comes expert with his or her knowledge and experience to help with drawing up a reliable estimate. Experts can be recruited within organization or can be hire as extern if a project is entirely new to the organization.

There are also weaknesses of this technique. An expert may be biased (more about biases is in the 5.1). He may have particular point of view that he may show in the significant influence on the estimate. It is possible that expert overestimated skills of the other employees in the sense of 'everybody knows that!'. Therefore judgment of the expert is not always objective. To get more objectivity it is better to involve more experts. It creates the balance between experts subjective opinions. Quality of estimate depends on simplicity or complexity of estimated task itself, but mostly on expert experience. The inconsistent result from estimations are reducing in iterative approach until an acceptable estimate is achieved.

There are various ways to control expert estimation:[5]

- The quickest method is to compare experts individual results and take the average.
- Another way is to organize a brainstorming with experts. Then the test manager gets not only an average, but well—considered basis for estimate. However results depends very much on moderator, for example, if he let too liberal running of the brainstorming, one of the experts can force his opinion to others. Also a wrong choice of experts can influence the estimate.
- The test manager can use Delphi technique or its derivative called wideband delphi technique which are described in section Team estimation sessions which are described in 3.6.

3.3 Three point estimation

This technique is a nice example of combining experience-based and model-based techniques.

First of all we must point out that this technique requires three estimates on each task. Then the technique, based on estimates, statistically determine how much effort will be required with a given probability to success in test plan. The three estimates are:

- *The most pessimistic estimate* — estimate where we have to count with the worst conditions and unplanned difficulties.
- *The most realistic estimate* — estimate where we working with usual amount of unplanned difficulties.
- *The most optimistic estimation* — estimate where everything is going fluently and there is no unplanned difficulties.

For a top-down estimation approach we estimate three values for the total effort and then calculate a mean value of the three estimates above. In this case it is a weighted mean value because we use the realistic estimates 3 or 4 times. See example: [1]).

Let define W as a weighted mean, P as a pessimistic estimation, R as a realistic estimation and O as a optimistic estimation. The result we can see in equation (1) [2]).

$$W = \frac{(P + (R * 3) + O)}{5} \tag{1}$$

Next we have to calculate an approximated standard deviation (2). Let define SD as standard deviation, P as a pessimistic estimation and O as a optimistic estimation.

$$SD = \frac{(P - O)}{5} \tag{2}$$

Then we can calculate a single estimate in which we can have approximately 95% trust (3). Let define E as estimate, SD as standard deviation and W as a weighted mean.

$$E = W + (2 * SD) \tag{3}$$

For example. If we estimate a testing effort as follows:
Pessimistic = 100 days
Realistic = 80 days
Optimistic = 60 days
Then we can calculate:

$$\begin{aligned} W &= \frac{(100 + (80 * 3) + 60)}{5} = 80 \\ SD &= \frac{(100 - 60)}{5} = 8 \\ E &= (80 + (2 * 8)) = 96\ days \end{aligned} \tag{4}$$

This calculation gives us about 95% confidence level that the testing effort estimate for project is 96 days. We can even easily count other levels of confident by adjusting final formula as we can see in 3.1 [3]).

For an approximate confidence level of:	Use this formula
68%	E = W + SD
90%	E = W + (1.645 * SD)
95%	E = W + (2 * SD)
99.7%	E = W + (3 * SD)

Table 3.1. Table of estimation for other level of confidence.

[1]) This example is taken from ISTQB syllabus. [7]
[2]) Dividing by 5 because there are 5 values (we are including 3 times realistic estimate)
[3]) Table is taken from ISTQB syllabus [7]

Task	P	R	O	Weighted Mean	Standard deviation	Standard deviation^2
A	9	6	4	6.2	1	1.00
B	5	3	2	3.2	0.6	0.36
C	10	7	5	7.2	1	1.00
D	8	6	4	6	0.8	0.64
E	7	5	3	5	0.8	0.64
			Total:	27.6		3.64

Table 3.2. Example of estimates for five tasks. [2]).

If we want to combine this technique with bottom-up technique we need to calculate weighted mean, standard deviation and squared deviation for each task. Then we have to add all the weighted mean values together to derive overall weighted mean estimate and add all standard deviation together and calculate the square root of this value to give us standard deviation to total test effort. To make it clear see example in table 3.2 [1]), where P is pessimistic estimate, R is realistic estimate and O is optimistic estimate.

Then using the sum of weighted mean and standard deviation, we can calculate standard deviation for total effort a final estimate. Standard deviation for total effort is represented by the (5), where SDT is standard deviation total and total project effort with 95% confidence is represented by the (6), where TPE is total project effort.

$$SDT = \sqrt{3.64} = 1.91 \tag{5}$$

$$TPE = 27.6 + (2 * 1.91) = 31.4 \tag{6}$$

So with bottom-up technique we reach the result for five tasks with quite simple approach. The project will take 31.4 days with a 95% confidence. If we wanted more accurate or less accurate estimation we just change the value which we multiply the standard deviation for total effort.

We can see number of benefits of this technique. Obviously the estimation is more accurate than with one-point estimation. Estimators adding considerations for risk continuously without adding them after. So there is no need to get back to the estimation only for refining. It is a way of expressing uncertainty in the estimate. The grater standard deviation is the more uncertainty is there.

3.4 Work breakdown structure

As the name implies, this is method where we dividing project into small manageable tasks. It is a tree structure, which shows subdivisions of effort required to achieve the objective. Identifying the main deliverable of a project is starting point for deriving WBS. When document from project manager and involved people is ready, test manager and team leaders start breaking down the high-level tasks into small tasks. These small tasks are derived by functionality and time is estimated separately for each chunk. This

[1]) Example and table is taken from ISTQB syllabus [7]

give us a good supervision and it can be nicely estimated. Also we can use this derives to identify potential risks in a given project. If we have some branch which has not good specification, then its represents a scope definition risk. For this type of branch, we should consider more time for executing tests, because we do not know exactly what it should do. If tests are failing in behind, referring WBS will quickly identify deliverables impacted by a failing branch. Additionally, it is a dynamic tool and can be revised and updated as needed.

It is much easier to estimate small tasks. High level estimates are then determined by adding together the estimates from the smaller tasks. This is shown in the figure 3.3.

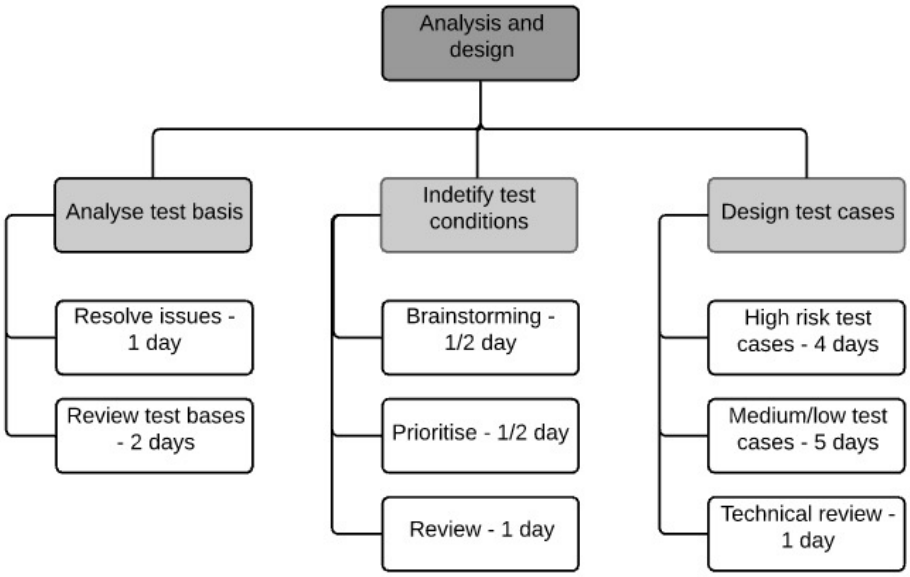


Figure 3.3. Schema of work breakdown structure.[1])

There are, of course, some advantages and disadvantages for this technique:

- Advantages
  - The WBS gives us complete view of project test scope and prevents confusion what should be tested.
  - The WBS describes dictionary and common notations which served as a reference for all involved persons.
  - The WBS is effective tool for resource management.
- Disadvantages
  - Requires quite a lot of work and time to build and maintain the WBS.
  - The WBS support rigid structure, so its not so flexible for initiate or lead changes during project life cycle.

[1]) Source: Author.

3.5 Extrapolation previous data

In estimating a new project, we can estimate relying on historical data. Using metrics from previous testing of similar project can be a good way to derive realistic estimate. For example, if we know that there were 100 new functional tests designed and used and total effort to execution was 50 hours. We can easily calculate that testing effort for one functional test was 30 minutes. For our new project, if we have 75 new functional tests we can make simple estimation that for execution these tests we will need $75 * 30 = 37.5$ hours.

However this assumes that all aspects will be similar with the project before. This means, for example, that the code is written with the same people following the same principles, the same complexity and under the same deadline pressure. If this is not achieved then we can adjust time the estimate by relevant differences into consideration. For example, if the developers will be under bigger deadline pressure, we can assume, that there could be higher possibility of defects, which means longer time to testing. With this assume we may allow instead of 30 minutes per test 35 minutes per test and this will increase total effort to $75 * 35 = 43.75$ hours.

To be able to use this techniques, we need data which are available, applicable and accurate. The data represents experience but not necessarily our experience. We can apply our own experience to adjust the measure when applicable. The more detailed data we have, the better our estimates can be. But also if the data is detailed to much, we may spend too much time and this may not be worth the extra accuracy achieved, even though, some other aspects of a project can change and this undermines some of our assumptions. Data is coming from industry data or benchmarks from our or other organization.

3.6 Test estimation sessions

In the test estimation sessions the whole team makes agreement on the final estimation. It is more reliable then estimates from individuals, however, more people are involved and organize more people usually takes much more effort.

Team is usually small number of people or representative subset (involving people that have knowledge about all aspects of the project). The goal of this team is to achieve consensus about estimate that everyone agreed. There is description of some specific techniques which belongs under team estimation sessions.

3.6.1 Planning poker

The idea behind planning poker is very simple "planning poker is a consensus-based estimation technique for estimating, mostly used to estimate effort or relative size of tasks in software development"[1]). The 'estimation team' involves product owner and selection of the test team or if the team is small then all members are invited.

Each estimator has deck of cards 3.4. They are quite often close to Fibonacci sequence or it can be, for example, a shirt sizes ranging from extra-small to extra-extra-large. However the Fibonacci sequence is recommended because of its proportional grows. The proportional grows reflects uncertainty with a test story. This means that high estimate usually means that the story is not well understood or may be it should be broken down into smaller tasks. The 'joker' (question mark) means 'unsure' or 'I need a coffee'.

[1]) `https://en.wikipedia.org/wiki/Planning_poker`

Then the product owner or stakeholder reads a user scenarios to the estimator. The estimators discuss the feature and asking question until everything is clear. When the feature has been fully discussed, each estimator choose and place the card down on the table at the same time so as not to influence another estimators by laying card earlier then others. If all estimators agreed on one particular value it becomes the estimate. If not, the estimators must discuss the differences in estimates. Now the estimator with the lowest and also the highest estimate explain their thinking. After this comes another round of discussion and the poker round is repeated until agreement is reached either by consensus or rule, which was provided before starting the poker. For example, define the limit of the rounds.

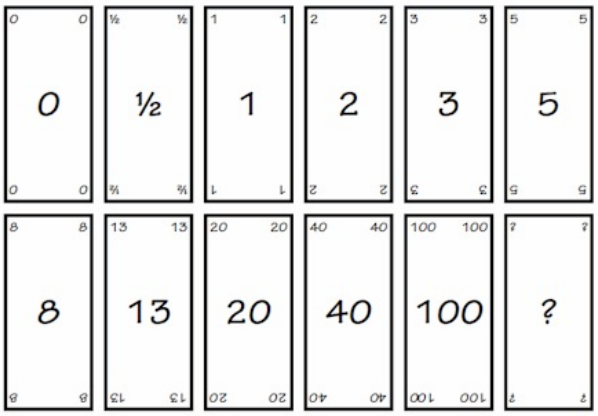


Figure 3.4. Planning poker cards.[1])

3.6.2 Wideband delphi

The wideband delphi process 3.5 was developed in 1950s at RAND corporation as a forecasting tool. Name Delphi comes from ancient Greece. It was the most important oraculum with perfect ability to predict. It has proven to be a very effective estimation technique and suits very well on software projects.

Project manager selects moderator and the team with three to seven members. First meeting is kickoff meeting, where the estimation team creates a WBS and discuss assumptions. Then each team member creates the effort estimation for executing tests by his allocation. Further comes second meeting called estimation session, in which team reviews the efforts each other and try to achieve some consensus with discussion. Then project manager summarizes the results with the team.

- Advantages:
  - More participants can be involved then in face to face estimation methods.
  - Allows sharing information and reasoning among team.
  - Free of social pressure, personality and individual dominance.
- Disadvantages:
  - Individual estimate can be influenced by group.
  - Tendency to eliminate extreme position and force to middle estimate.
  - Time consuming because it requires adequate time and resources.

[1]) `http://blog.garethjmsaunders.co.uk/2010/04/17/agile-planning-poker/`

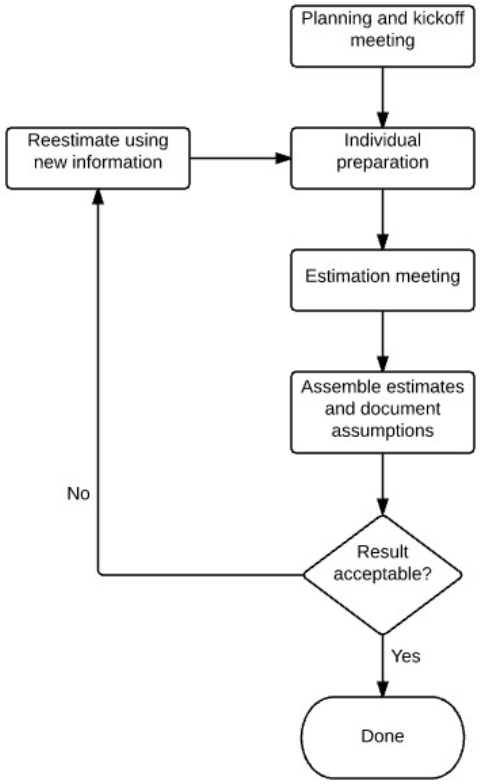


Figure 3.5. Wideband Delphi schema.[1])

Model-based estimation techniques

3.7 Estimation based on ratios

When we want to use effort estimation based on ratios, it is important to collect the greatest possible amount of results from previous projects. Project needs to be similar, this means, that the project is similar in key properties. For instance the project has the same development method, same experienced developers, same development platform, the same software environment, etc. This method is very helpful and useful within organization with a lot of similar projects. Use own ratios are the best ones and can be used at all estimation levels. However, the ratios are so specific, that they can be used only in one organization and often even within the area of the project, application or system.

Below is a basic distribution of ratios between tests and other development activities. We can use these observations as a starting point but then keeping revisiting the ratios. We can modify them more and more adequately to our project, application or system.

[1]) Source: Author.

There are some ratios which are derived from observations: [3]

- Functional design (FD) = functional detailed design.
- Realization, consisting of the technical design (TD), programming (P), unit and unit integration test (UT and UIT).
- Functional test. This concerns the testing of functionality quality characteristics, with the FD as the test basis.

Observed ratios in an average risk profile are as follows:

- In an environment with a formally complete FD (7)

$$FD : Realization : Functionaltest = 2 : 5 : 3 \tag{7}$$

- In an environment with an incompletely detailed FD, experienced builders who filled the FD's themselves, and a starting test approach (8).

$$(FD + TD) : (P + UT + UIT) : Functionaltest = 1 : 3 : 3 \tag{8}$$

- In a test environment with a formally complete FD, waterfall development method, experienced builders, and a functional test that does not have maximum test coverage. The test approach is structured (9).

$$FD : Realization : Functionaltest = 1 : 2 : 1.2 \tag{9}$$

Ratio can be used even to estimate the various phases of testing. There are also some data from observation of actual practice:

- For a system with a good but complex specification, the observed ratio is as follows: Preparation 6%, Specification 54%, Execution 21%, Completion 2%, and 17%for control and setting up maintaining infrastructure taken together.
- The following ratio was observed for a system with an inadequate test basis: Preparation 21%, Specification 33%, Execution 24%, Completion 5%, and 17%for control and setting up and maintaining infrastructure taken together.

3.8 Using percentage

This technique belongs to top-down estimation method, because of using the estimates for project as a whole. This technique applies a formula, which is a percentage expression, to determine the proportion of effort. So for every phase of the project is percentage evaluation of the effort, for example, project effort can be split between development, testing and other activities in proportions: 45%, 30%, 25%. This means that if we have effort evaluated on 200 days, then testing according to the ratio takes 30%, which is 60 days.

This technique can work well within organization where proportions are taken from the similar project what have been done before. If the percentage figures are not updated with effort of the recent project, it can lead to making the data out of date.

A common variation is set to tester-developer ratio. Lets assume that the ratio is 3:5, this means that for a 5 developers we will need three testers. However, if the development effort and resourcing also keeping some of best practicing in development, such as doing unit test and, some of integration tests and continuous integration, the ration can be different. It depends on working environment of the organization.

3.9 Test Point Analysis

This technique makes it possible to estimate only a system or acceptance test. TPA measure software testing size and reflects the complexity of testing activities to ensure the quality of the software. Because of complexity, it needs to reflect the effort required to perform testing activities like planning, designing, executing tests, report and tracking issues or defects. There are not included development tests, it is out of scope. TPA can be used if the scope of the test project is already known or it can be used if the number of test hours, which are allocated, is determined in advance. This can helps us to prevent risks by comparing objective TPA estimate and predetermined number of test hours. A TPA can be also used to calculate relative importance of different functions or create a global estimate in early phase of project.

When we making a TPA, three elements are relevant: size of the information system to be tested, test strategy and level of productivity. Two first elements together determine effort of testing work to be executed (expressed in test points). Productivity is the amount of time needed to execute a given volume of tests. The test estimate result in hours give us the number of test points multiplied by level of productivity. The three element are described below.

3.9.1 Three elements of TPA

There are three elements of test point analysis [3]:

- ***Size***

  The first element to be considered is size of the information system what we are testing. In TPA, the size of an information system is determined primarily on the number of function points assigned to it. Function points or sometimes called use-case points has following factors which have little or no influence factor on the number of function points. These three factors are relevant to test:

  Complexity: More conditions almost automatically means more test cases and therefore a greater volume of testing work.

  Coupling: The degree of coupling is determined by the number of data sets maintained by a function and the number of function which make use of those data sets. These other functions must be tested if some changes are made on maintenance function.

  Uniformity: If there are multiple functions with same structure in the information system and tests can be used with no more than small adjustments.

- ***Test strategy***

  During development and maintenance, quality requirements will have been specified for the information system. During testing must be established boundary which quality requirements are complied and which are not. However, there is not unlimited time for test all combinations of these requirements. The importance of each characteristics influences the thoroughness of the related test activities. The importance of the various characteristics should be determined in consultation with client or based on product risk analysis when the test strategy is being formulated. The test strategy represents input for the test point analysis and than it is translated to time which is required for testing.

  In addition to the general requirements of the information system, there are differences between the various functions in terms of requirements to be met. For example from user's perspective, a function which is utilized throughout the day will be probably much more important than a processing function which operates only at night.

Each function determine the user importance of the function and the intensity. These two factors are, of course, based on test strategy.

The test strategy specifies which quality characteristics must be tested with what thoroughness. TPA and strategy determination are closely related and in practice are often performed simultaneously [4].

- ***Productivity***

  Productivity is not a new concept to anyone who have already made some estimates based on function points. For TPA, productivity means the time required to realize one test point, determined by the size of the information system and the test strategy. The productivity is build from two components: skill and environment factor. Skill factor is based primarily on knowledge and skill of the test team. The environmental factor indicates the degree to which the environment influences the test activities to which the productivity relates. This involves aspects such as availability of the test tool, the amount of experience the team has with the test environment, quality of test basis and the availability of testware [3].

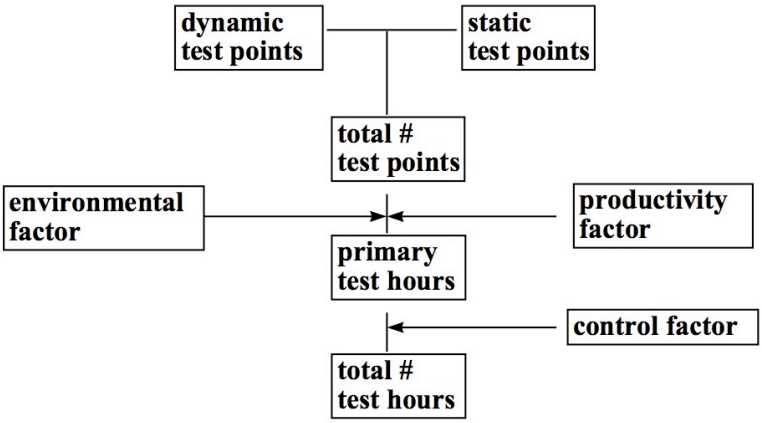


Figure 3.6. Schematic representation of test point analysis. [3]

3.10 Conclusion

So far, in this chapter was described number of techniques for drawing up an estimation. These techniques all have their weaknesses and strengths. Using several techniques can compensate theirs imperfections. The time for using several techniques must be, of course, included in the budget and plan. Lets see an example:

Using the bottom-up technique a test manager has drawn up an estimate for complex environment. There is a small chance, that some technical component was forgotten. But there is a possibility that some aspects at the level of the total effort are being overlooked. Now it comes second technique called expert estimation. The test manager asks one or more experts to draw up a top-down estimate. By coming in at the high level, the experts can found some space in our estimate such as integration.

If estimations comes with the similar result, than we can say that our estimation is good enough. The situation is different if the results are vary. This brings uncertainty and the variation must be deeply investigated. It is possible, for example, that bad techniques has been used for this situation. Seeking the variations of the results and solving them will probably lead to a better estimation.

In the tables 3.4 and 3.3, we can see overview of our techniques [1]).

Model-based techniques			
	Usability	Strengths	Weaknesses
Ratios	Use if quality data are being available	Fast	vulnerable to variables
Using percentage	Use if the organization is staying to one technology and methods many years	Efficient	Depends on employees
TPA	Use only for system and acceptance testing	Tune technique with a lot of detailed information	Restricted to system and acceptance testing

Table 3.3. Model-based estimation techniques overview. Inspiration of the table is taken from [5] edited by author.

[1]) In comparison, I also included the top-down and bottom-up approaches.

Experience-based techniques			
	Usability	Strengths	Weaknesses
Top-down	Use for a first estimate at a high level with low detail available. Consider at the begging of the project.	High level	Degree of detail
Bottom-up	Use when detailed informations are available	Detailed basis	Low involvement of testers
Expert	Use when situation is more complex	Usually fast estimate	Only as a good as a expert is
Intuition	Use in stable and consistent situation	Fast technique	Uncertainty
3-point	Use if we have time for estimation.	Very accurate	Time consuming
WBS	Use when environment is complex	Smooth communication	laborious
Extrapolation	Use if comparable data are available	Optimal use if projects are similar	Representativity of similar projects
Team sessions	Good in smaller team	Estimation comes from the most competent people	Time consuming

Table 3.4. Estimation techniques overview - experienced based. Inspiration of the table is taken from [5] edited by author.

Chapter 4
Costs in relation to the estimated efforts

The costs for testing are mostly influenced by the human effort, and cost estimation methods focus on this aspect give estimates mostly in *MD* [1]).

Accurate estimate of costs is critical for both developers and customer. Underestimate the costs can lead to approving proposed to management and this extend the time, underdeveloped functions or poor quality. Overestimating may result in not winning the contract. Also it is important because:

- It can help to classify and prioritize what test will be execute with respect to overall business plan.
- It can help with determining what resources should be committed to a test project and how these resources will be used.
- Tests can be easier to manage and control if resources are matched to real needs.

The costs for testing involves determination of:

- effort (in man-days)
- tests duration (in calendar time)
- cost (in current currency)

Although the effort and costs are closely related, it does not mean that we can simply make a transformation between them. Effort is often measured in *MD* of the testers, test analytics and test leader or manager. Therefore the effort estimate can be converted into cost figure by calculating an average salary per unit of the people who are involved and then can be multiply by the estimated effort.

And what is good estimate? Good test effort estimate should have:

- It is approved and supported by the project manager and customer.
- It is accepted by the all stakeholders.
- It is based on previous experience from similar project.
- It is based on estimation cost technique with a credible basis.
- It is based on on enough detail with description of all possible risk areas and probability of success of the project.

Is test worth? The answer is yes. The numbers vary by project and environment but basically the costs to fix the defects average what has to come to be known as "1:10:100" rule.[2]). This means that the defect that,for example costs 100 crowns to fix in requirement or design costs 1,000 crowns to fix in a traditional test phase and 10,000 crowns to fix after the product goes into the production use.

[1]) Man-days. It means 8 working hours.
[2]) `http://qa4software.blogspot.cz/2009/08/110100-rule.html`

4.1 Process of cost estimation

If a test analyst, leader or who doing the estimate could use an efficient test estimation technique to estimate the effort and implementing the estimation in test project, then the costs can be calculated easily. For example in top-down estimation technique, the cost estimation process involves few steps:

- Firstly the test analyst or leader have to draw up a high-level estimate effort for sections what will be tested. This effort is calculated in *MD*.
- The one who estimates should separate this estimates between manual testers, analyst and someone who control the test project (it can be project manager, test manager, test coordinator or test leader). This can vary in different projects. Basically we should try to separate it between all participants on the project with differing wages.
- Then the total costs will be counted from effort and salary of the participants. There is a simple example in section 4.2.

4.2 Example of cost estimation

Firstly we have to make estimation how much effort the testing will need. Then we can do estimation how much this effort will cost. The estimation effort can be done by techniques which are described in chapter 3. The estimation how much will cost is shown below. It is very simple crude example.

We can just based on our experience with a similar project. Yes, there is no problem with that, but then as we said, is good to compare our estimation with another test estimation technique. We will do that with help of three point estimation. Now we estimate an optimistic time, realistic time and pessimistic time for how long the estimate takes. Then we count some number according to 3.3. The result is, for example, 40 MD, just for testing. Then we can calculate that analytics needs 10 MD to analyze feature and make test cases and 2 MD for test manager to manage it together and make reports. The total effort of this feature is 52 MD. But we need to have in our mind, that 40 MD is for test execution, 10 MD is for analytic work and 2 MD is for manager. These three different positions have also different salary. So if we take the salary for the tester, for example, 100CZK per hour, analytic salary 200CZK per hour and 400CZK per hour for test manager we can estimate, that the cost for testing will be 32,000CZK for testers, 16,000CZK for analytic and 800CZK for test manager. The total cost of test this feature will be 48,800CZK. Then if the estimate with our experience estimate is equal, we are good to go. If not, we have to investigate, where our estimates differ and try to solve it and make one possible estimation.

As we said, it is a very crude estimate. But even a crude estimate is better then no estimate at all.

Chapter 5
Research of current practice in estimating software testing efforts

The research was conducted between experienced analytics, test leaders or test managers who did or currently doing test estimations on projects. Into the research was involved 13 specialists from practice.

The outcome from the research should be which techniques are mostly used, how accuracy of the estimation is reached, which areas go very well and which go very bad to estimate.

Also there is described how the cognitive biases of the human thinking can affect the estimation effort. There are three basic biases that most affect estimates — causes distortion of the estimates.

5.1 Cognitive biases

In the previous chapters, there were described techniques for estimation based on mathematical calculations, exact procedures or expert (best feeling) judgment. Or even some mix of these attributes. However, it is important to include one more attribute and it is human aspect.

The estimation is very dependent on acknowledgment and experience of the estimator. Human is, in his opinion, influenced by number of cognitive biases. *"A cognitive bias is a mistake in reasoning, evaluating, remembering, or other cognitive process, often occurring as a result of holding onto one's preferences and beliefs regardless of contrary information. Psychologists study cognitive biases as they relate to memory, reasoning, and decision-making."* [1])

5.1.1 Anchoring effect

Anchoring also known as focalism is a cognitive bias that describes the human tendency to anchor to the first piece of offered information when making decisions. Then, according to this information, is set result of the decision. For example, the initial price offered for a used car sets the standard for the rest of the negotiations, so that prices lower than the initial price seem more reasonable even if they are still higher than what the car is really worth.[2]).

While testing effort is estimated, the test manager should collect all needed information for the estimation. At the same time, he should not to give misleading information, which can affect the estimates because of anchoring effect. The important thing is to avoid of the expectation influence. If there is some vision of how long the tests should take then the final estimation could be affected by this anchoring effect.

[1]) `http://www.chegg.com/homework-help/definitions/cognitive-bias-13`
[2]) Example is taken from wikipedia `https://en.wikipedia.org/wiki/Anchoring`

5.1.2 Outcome bias

The main aspect which leads to the errors is desire for approval of a plan by the stakeholders. If the vendor convince customer and the customer succumbs the outcome bias, then in later phase of the project, vendor will have to face budget and/or time increase.

Another aspect which makes errors in estimation is optimism. The estimators, because of the optimism, just overlook option that there is a possibility of multiple requirements increase compared to the initially assumptions and plans.

5.1.3 Illusory truth effect

The illusory truth effect is the tendency to believe information, which is not approve but a lot of people believes in it. So human can easily succumb the illusion of the truth.

Nice example is an ad, which we see over and over again, and this persuade us (not always) to buy the thing. Repetition is one of the easiest and widespread methods of persuasion. What are the impacts on estimation testing effort?

The estimation depends on human psychological well-being. The estimator should not be in the optimistic or free mood, because then he or she could be influenced by the *cognitive ease*, this is dangerous, because it make us think that we understand far more than we actually do and this can leads to underestimation or overestimation of the project. Also, there is opposite of this and its called *cognitive strain*. When you fell cognitive strain, you are more likely to be more vigilant and suspicious, invest more effort in what you are doing, feel less comfortable and make few errors, but you also are less intuitive and less creative than usual.

5.2 Research results

I was asking 13 test analyst, leaders or managers questions about estimation effort in software testing world. The exact number you can see in the figure 5.1. They answered on survey which you can find in the appendix A.

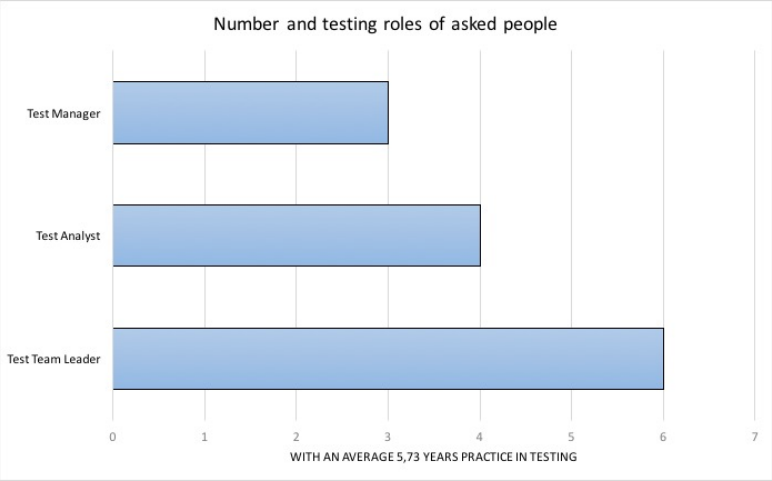


Figure 5.1. Number of the participants.[1])

[1]) Source: Author.

5.2.1 The usage of techniques

The first question was what estimation techniques do respondents know and which of them using. The most used techniques and also the best known techniques are Expert Estimation (section 3.2) and Intuition (section 3.1). This result is quite predictable because these techniques usually takes the lowest time.

Intuition can be done by every one, but accuracy of the estimation will be influenced by experience of the one who make this estimate. And not only experience in sense of how long is he doing testing but also how big is knowledge of tested application. In contrast, expert estimation cannot be done by everyone. As the title says we need this 'expert' and sometimes there is a lack of quality experts and this raises their price. So yes, this technique is fast but also can be expensive.

Of course, both of these techniques are perfect for combining with other techniques to increase the accuracy of estimation effort, which must be spent on testing application.

On the other hand, there is a Test Point Analysis (section 3.9). This technique is at the end probably because it can be use only for estimating system or acceptance tests but also for complexity of this technique. Test Point Analysis has including a lot of factors which have to be set. There is a quite a big chance to make a mistake or forget on some part and this may give us inaccurate estimation. Moreover the technique is very time consuming. Detailed overview of all techniques is in figure 5.2.

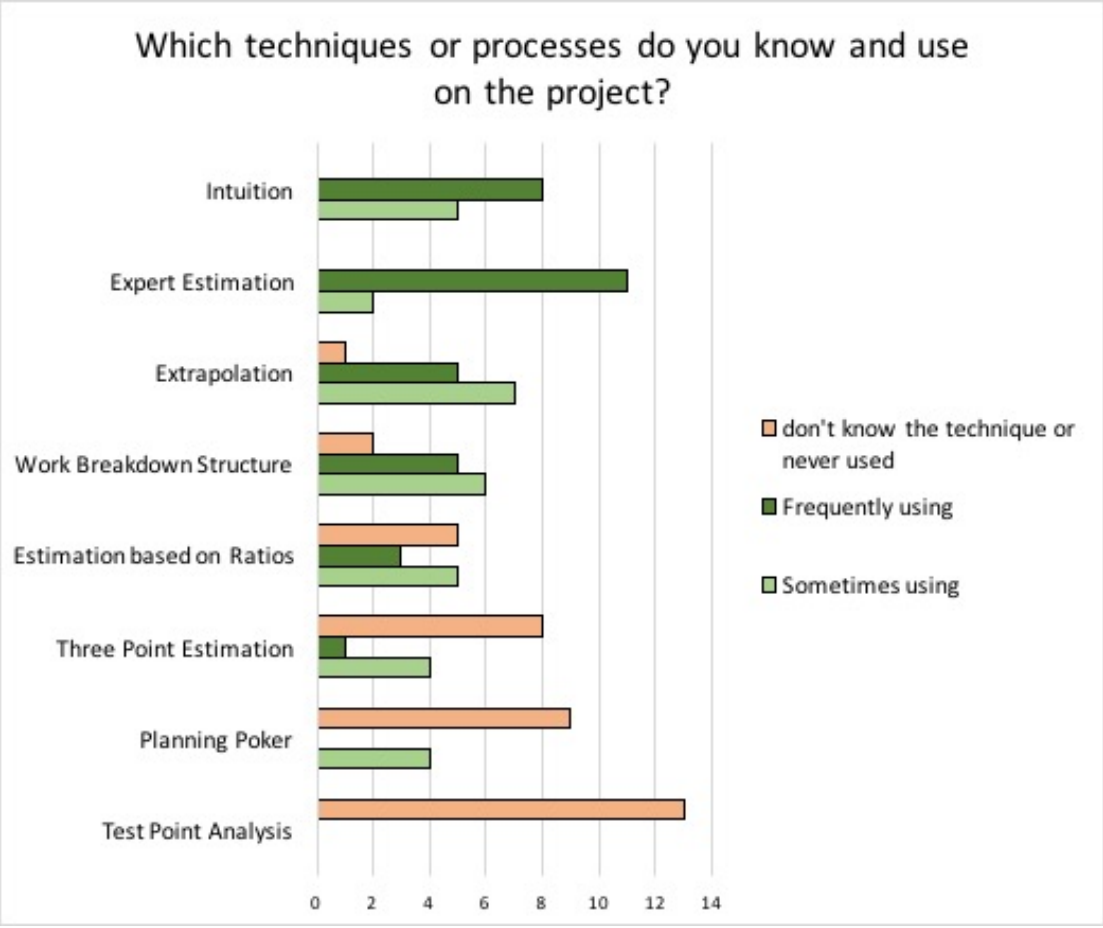


Figure 5.2. Table of the techniques usage.[1])

[1]) Source: Author.

Which technique to use for estimation is sometimes written in the methodology provided by company for current project. If not, the choice of an estimation technique is up to estimator. There is recommendation to use more than one. But how to make the decision of which technique to use?

The most common ways how we are choosing an estimation technique is shown in the figure5.3. In most of the cases it is done by the personal judgment, because we usually know the estimated project or it is similar to one which we estimated earlier. With personal judgement is related choosing the technique by previous data. For example, if we do not know the project, but they provide us some earlier data then we can make decision which technique to choose. Also we have to be very careful about cognitive bias called Anchoring effect 5.1.1 . So it is suitable to confirm the provided data.

Choosing an estimation technique by size of the project can be tricky. After the first quick familiarization with the project we should decide which estimation technique to use for estimation of the testing effort. This is also related with our experience. For project which is not too complex and an estimation can be done high level we perhaps use Intuition estimation technique with combination with some other technique to make the accuracy of the estimation as good as it is possible. In contrast, for complex project which need detailed estimation we will use WBS with combination, for example, expert estimation for estimating the small tasks in the WBS.

Interesting is that no one wants to make decision according to article on the internet. More details are searchable in the figure 5.3

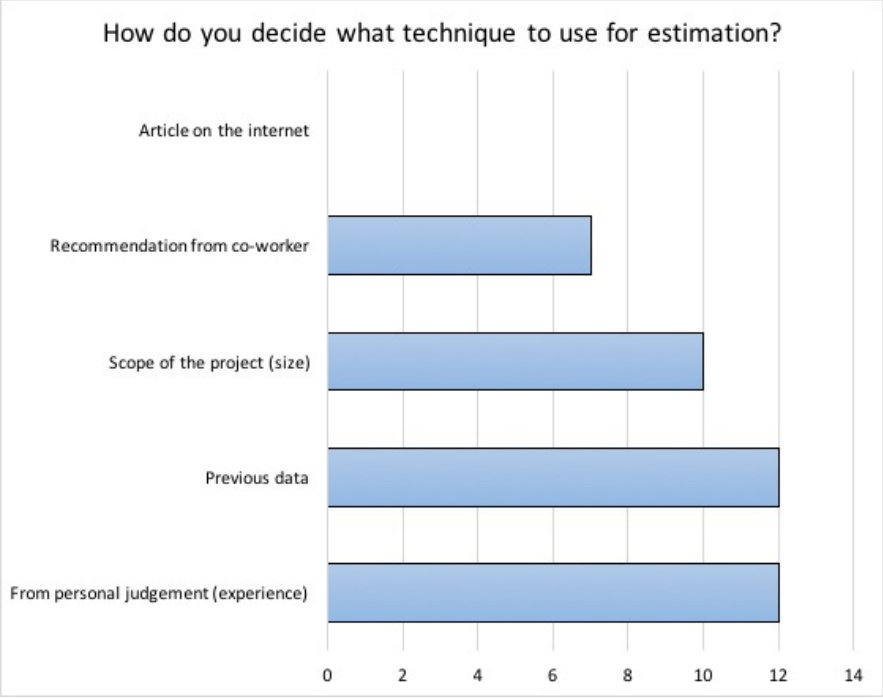


Figure 5.3. Overview of the options how to choose the estimation technique.[1])

Experiences with collecting the data for estimation of the testing effort are quite similar among respondents. Usually there is lack of time for collecting quality data and making assumptions about testing effort size spent on the tests. Due to lack of the data, an estimation has tendency to be underestimated or overestimated, especially for executing the tests.

[1]) Source: Author.

5.2.2 Inaccuracies in estimations

Inaccuracies in the estimation techniques are mainly influenced by two groups of factors. One group is inaccuracy from material causes and second group is inaccuracy from psychological causes.

In the first group there are some factual things which can influencing the estimations. There are included:

- *Functionality of environment*: If environment is ready for executing the tests in time.
- *Testability of the application*: If developers delivery not testable product.
- *Late delivery*: If developers or analytics or third party sides which are involved into developing the application are making late delivery.
- *Big amount of bugs*: If new bug occur when a solution is found for fixed bug.
- *No or missing documentation*
- *Changing the scope of project*: If there are some change requirements which was not planned.
- *Wrong estimation technique*

From the Figure 5.4 we can see that he most influencing material factor which makes testing effort estimation inaccurate is nonfunctional environment together with changing the scope of project. These two are followed up by wrong estimation technique and big amounts of bugs. The other material causes are almost similar.

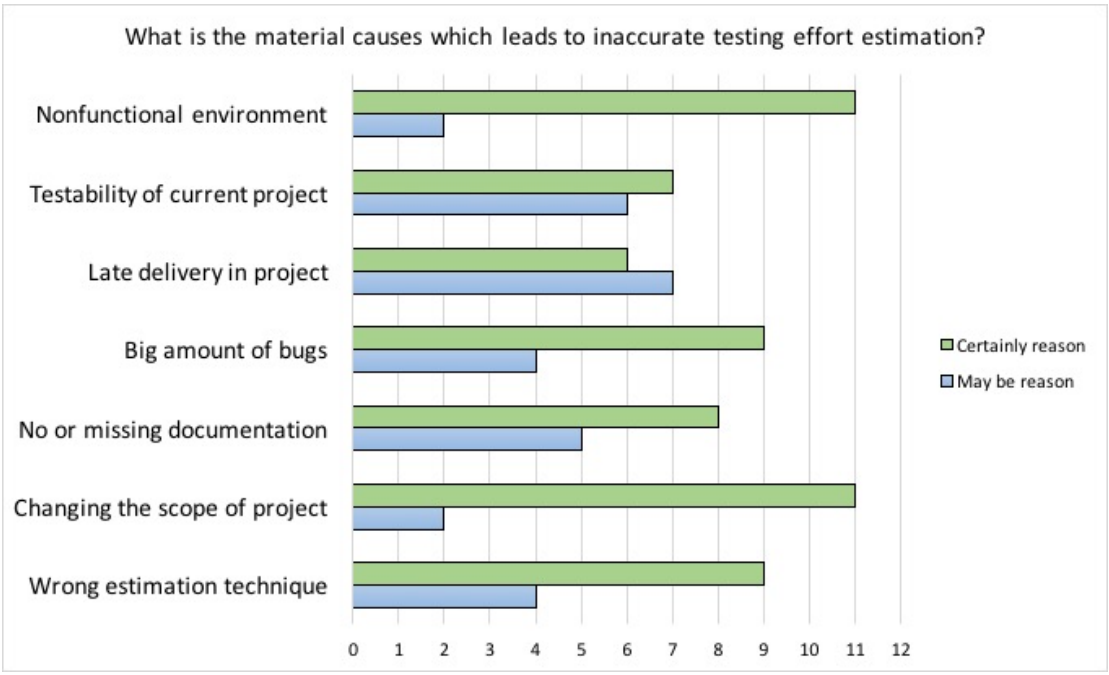


Figure 5.4. Overview of material causes.[1])

[1]) Source: Author.

The second group contains psychological causes which can leads to inaccurate testing effort estimation. There are:

- *Try to hit assumption*: To comply to owner's assumption.
- *Estimation details*: Tendency to put martial estimate.
- *High or low level estimate*: To make the high level estimate in thinking of low level estimate.
- *Problem with calibration*: Set the right amount of testing effort.
- *Bad communication in team*
- *Flaming personal judgement*

From the Figure 5.5 we can see, that the most psychological causes which involving the testing effort estimate is trying to hit an assumptions of the owners. It is followed up by estimation details, it means to put the martial estimate, for example, in one test suite, but the testing effort for single test in test suite can be different. But overall there is no big differs between causes except trying to hit the assumption.

In psychological causes which leads to inaccurate estimation, cognitive biases play big role. This is confirmed by our survey, because the main psychological cause, trying to hit the assumptions of owners, is one of the biases which are described in section 5.1. There are more details about biases.

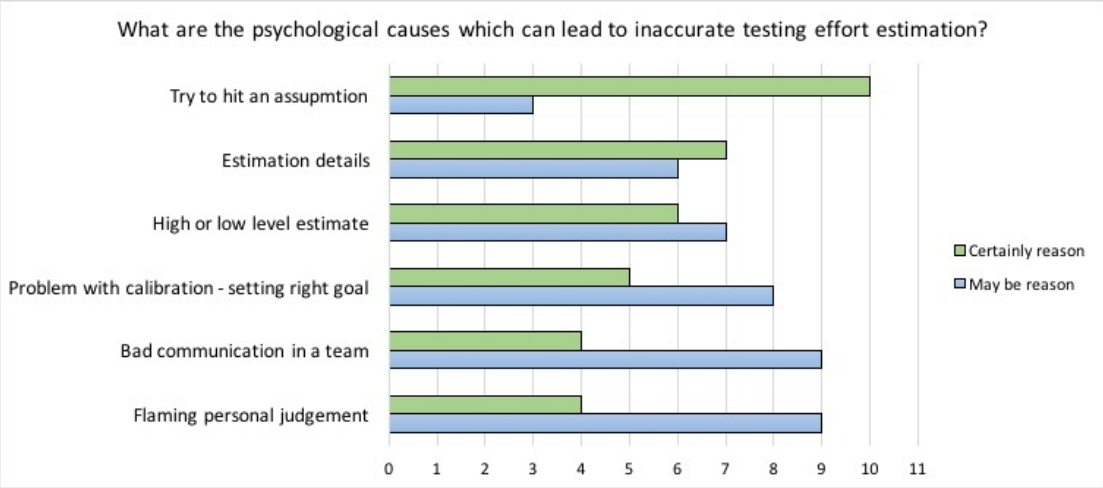


Figure 5.5. Overview of psychological causes.[1])

5.2.3 Successes and failures in testing

In this section was the main purpose to find out which parts of the estimations is possible to estimate better and worse.

The most successful part seems to be the time for preparing test cases followed by preparation data for tests. This part are quite successful to estimate because there is no third party source. It is just about people in team and test manager and leader usually knows the team well and know how people work.

State of the Application after tests, Test management and time for executing tests are in hook. This is because in these parts begin to engage other members of the team and it's not about individuals but about cooperation and preparedness environments.

Preparation environment for tests is almost at the end because I asked mostly people from in banks and there are quite strict rules and sometimes can be the time extended by bureaucracy.

[1]) Source: Author.

Last two, State of the Application after delivering to particular environment and time for bugs testing are parts with the most failures. This mainly because of the variables which are hard to forecast. These variables can be bad migrations, bad codebooks and much more. These little things can extend the time for testing more then we expected. More details are searchable in the figure 5.6.

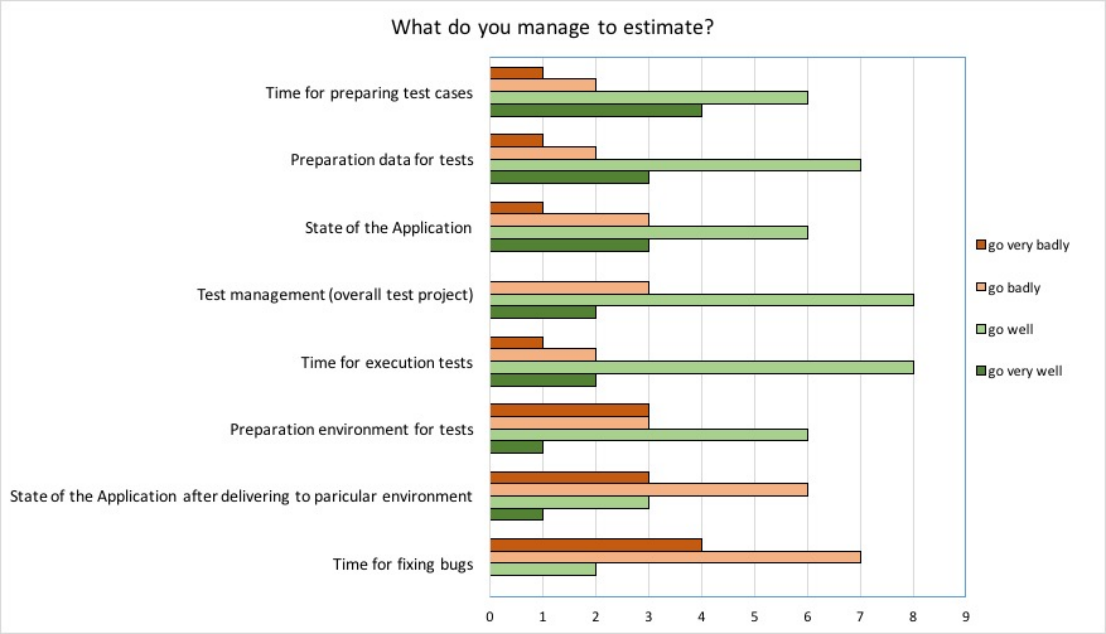


Figure 5.6. Overview of success and failures in testing.[1])

Deviations are related with the success or failure of the estimate. Respondents answered on the questions "With what accuracy are you able to make the final estimate? What is the percentage of the deviation?". The results are similar to the figure 5.6. The time for fixing bugs is estimated with only 13% deviation with means 87% success in this area. Opposite of this is the time for fixing bugs, there is the deviation 30% which means 70% success. More details are searchable in the figure 5.7.

[1]) Source: Author.

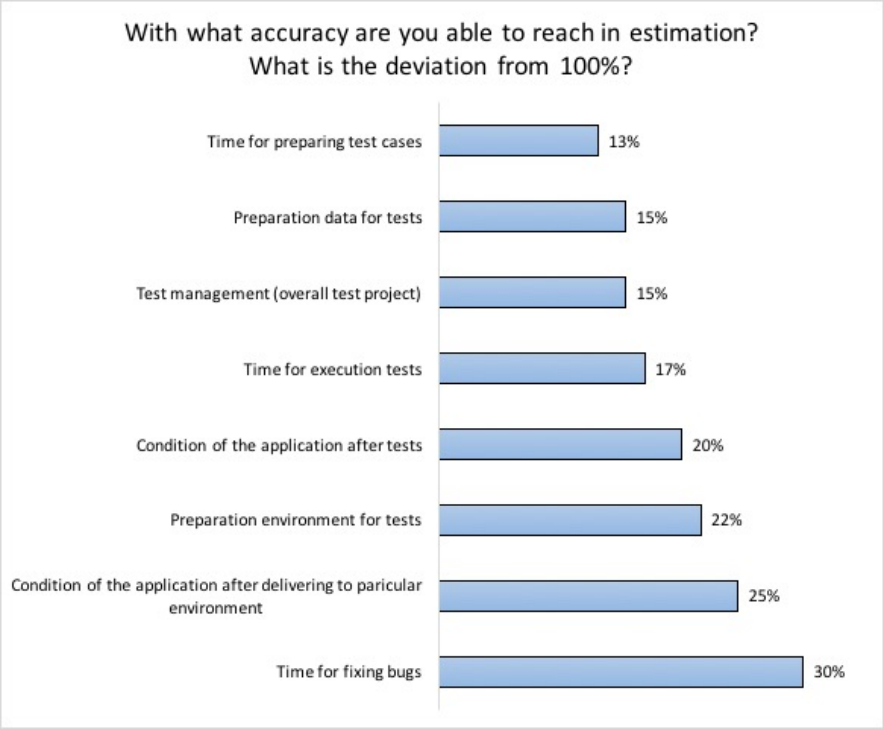


Figure 5.7. Deviations in the estimate of concrete part.[1])

5.3 Summary of result from survey

The survey found that the most using techniques for the testing effort estimation are Intuition and Expert Estimation. The least using technique is TPA. TPA is either known and never used or unknown. This could be because the technique can be used only for estimating system tests or acceptance tests or because it is too lengthy for making the estimate.

Decision what technique to use is the mostly done by personal judgement or according to size of the project scope. Sometimes the techniques are listed in the methodology of the project.

From a substantive point of view, it was found that the estimation is mostly influenced by changing of the scope, nonfunctional scope, big amount of bugs and bad choice of the estimation technique. This is mainly because there is a lot of variables which are hard to forecast.

Further has been found that the most successful estimations are in the time for preparing test cases (87%), preparing data for tests (85%) and overall test management (85%). On the contrary the least successful estimation is in the time for test execution (70%). Again there is a big influence of the variables, including the time for bug fixing, which is hard to forecast.

The most tolerant technique which works naturally with inaccuracies is Three Point Estimation together with WBS. But both of these techniques are individual time consuming.

Definitely the fastest technique is Intuition and Expert Estimation, however, Expert Estimation is difficult on human resources and experience of the expert. Intuition is

[1]) Source: Author.

simple and it can be done by every one, but it is also dependent on experiences how good will be the estimate. Intuition is used mainly within one organization.

Every technique is quite ease to learn except TPA. TPA is difficult in sense of numbers of steps to reach the estimation and very time consuming. Small mistake in the beginning can very influence the final estimate.

The most resistant to cognitive biases seems to be techniques which belong under techniques "called team session techniques". For example, Planning Poker or Wideband Delphi. Within these techniques the problem is usually discuss in the team and this give us more points of view at the testing effort. This can provide us very good estimation but it is also very time consuming and difficult on human resources.

5.4 Conclusion

My goal in the research was to make analysis of the current practice in estimating software testing efforts in the Czech Republic. The survey, which is in the appendix A helped me to get needed data from test managers, test team leaders and test analyst who making the testing effort estimates. To this research contributes 13 respondents.

The research showed that we definitely should use more than one technique to making estimations in software testing effort. The more techniques we use the more accurate will be our final estimate.

We should also verifying our estimations through the whole project and adjusting should be done as soon as possible to keep everybody in touch with actual state. After project ends, it is good idea to archive data from the estimation. Then with helping these data, we can make more accurate estimation in software testing efforts in the future.

In addition to the value that results from the testing efforts estimation, we should work with a degree of inaccuracy of the technique. This degree will be different, for example, for estimation using Intuition and Three point estimation.

We have to avoid cognitive biases which influence making estimations. Especially the outcome bias, when we trying to hit the assumptions of the owners. Our estimates will be more accurate if we can reduce the cognitive bias effect.

Chapter 6
Conclusion

The topic of the bachelor thesis was to analyze current practice in estimating software testing efforts. The basis for successful work was to study problematics about an estimations. Another aspect was to create a structured survey on the theme estimation software testing efforts, which is in the appendix A.

In the first chapter are introduction, aims and assumptions of the bachelor thesis and target groups for whom the bachelor thesis is intended.

In the second chapter there is a description of what is an estimation. Further there is a description of the test effort estimation and best practices for creating good estimation effort in the software project.

In the third chapter is selection of estimation techniques. These techniques are redesigned to effort estimation.

In the fourth chapter is description of costs which have a lot of common with estimated test effort.

In the fifth chapter I am trying to reproduce results from the surveys. For this were asked 13 respondents from testing environment detaining more than 5 years of experience in the domain. The result is a structured output where are the most widely used techniques, what is the accuracy of the estimates achieved in various fields of tests, tests which areas go well and which go bad to estimate, which factors affect the estimates the most and based on what is chosen the technique for estimating. Also there are key factors of mentioned techniques and selected cognitive biases which affect the estimate.

This bachelor thesis is providing the overview of the current practice in estimating software testing efforts according to the specification. Also it is providing the summary of the most using software testing effort estimation techniques.

References

[1] Capers. Jones, and Olivier. Bonsignour. *The economics of software quality*. Upper Saddle River, NJ: Addison-Wesley, c2012. ISBN 0132582201.

[2] Murali Chemuturi. *Software estimation best practices, tools & techniques*. Fort Lauderdale, Fla.: J. Ross Pub., c2009. ISBN 1604270241.

[3] Tim Koomen ... [et al.]. *TMap Next*. 's-Hertogenbosch: UTN Publishers, 2006. ISBN 9072194802.

[4] A. Abran. *Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers*. Los Alamitos, CA: Wiley, 2015. ISBN 9781118954089.

[5] Iris Pinkster. *Successful test management*. 2010 edition. Berlin: Springer, c2004. ISBN 3540228225.

[6] Rex Black, and Jamie L. Mitchell. *Advanced software testing*. 1st ed. edition. Santa Barbara, CA: Rocky Nook, 2009. ISBN 9781933952390.

[7] *ISTQB*. 2012.
`www.istqb.org`.

Appendix A
Survey

Questions in survey:

1) Which techniques for test estimations do you know?

- Three Point Estimation
- Work Breakdown Structure
- Test Point Analysis
- Planning poker
- Extrapolation
- Expert estimation
- Estimation based on ratios
- Others:

2) Did you use any of the technique on the project? Please choose one or more options.

- Three Point Estimation
  - Frequently
  - Sometimes
  - Never
- Work Breakdown Structure
  - Frequently
  - Sometimes
  - Never
- Test Point Analysis
  - Frequently
  - Sometimes
  - Never
- Planning poker
  - Frequently
  - Sometimes
  - Never
- Extrapolation
  - Frequently
  - Sometimes
  - Never
- Expert estimation
  - Frequently
  - Sometimes

- Never

- Estimation based on ratios
  - Frequently
  - Sometimes
  - Never
- Others:

3) Describe experience with learning and getting data in the sense of the estimation. (Please write two or three sentences/points).
4) What are the main factual causes of inaccuracies in the estimation?

- Bad estimation technique
  - Certainly reason
  - May be reason
- Change scope of the project
  - Certainly reason
  - May be reason
- No or missing documentation
  - Certainly reason
  - May be reason
- Big amount of bugs
  - Certainly reason
  - May be reason
- Late deliveres in project
  - Certainly reason
  - May be reason
- Testability of current application
  - Certainly reason
  - May be reason
- Nonfunctional environment
  - Certainly reason
  - May be reason
- Others:

5) What are the main psychological causes of inaccuracies in the estimation?

- Bad communication in team
  - Certainly reason
  - May be reason
- The problem of setting overall goals
  - Certainly reason
  - May be reason

- High – level estimation
  - Certainly reason
  - May be reason
- Try to hit the assumptions
  - Certainly reason
  - May be reason
- Flaming personal judgement
  - Certainly reason
  - May be reason
- Others:

6) Do you have experience with acceptation of technique from other project?

- Yes
  - Technique did not need modification
  - Technique had to be modified
  - Technique was wrong choice
- No

7) What are you able to estimate well and what badly? Please choose one answer in every section.

- Time for preparation test cases
  - Very well
  - Well
  - Badly
  - Very badly
- Time for test execution
  - Very well
  - Well
  - Badly
  - Very badly
- Time for fixing bugs
  - Very well
  - Well
  - Badly
  - Very badly
- State of the application after coming to the new environment
  - Very well
  - Well
  - Badly
  - Very badly
- State of the application after end of the tests
  - Very well

- Well
- Badly
- Very badly

- Test management (overall)
  - Very well
  - Well
  - Badly
  - Very badly
- Data preparation
  - Very well
  - Well
  - Badly
  - Very badly
- Environment preparation
  - Very well
  - Well
  - Badly
  - Very badly

8) With what accuracy are you able to make the estimation? What is the percentage of deviation in the estimate? Please fill deviation in percentage to the brackets.

- Time for preparation test cases (....)
- Time for test execution (....)
- Time for fixing bugs (....)
- State of the application after coming to the new environment (....)
- Condition of the application after end of the tests (....)
- Test management (overall) (....)
- Data preparation (....)
- Environment preparation (....)

9) How do you decide what estimation technique to use for estimation?

- Previous data
- Article on the internet
- Personal judgement (experience)
- Recommendation from co-worker
- Scope of the project (size)
- Other:

Appendix B
Glossary

MD	■ Man-days, it is 8 hours working time.
MTP	■ Mater Test Plan
Test Analyst	■ The test analyst is responsible for developing the test analysis. He works with the specification of the project and determine the test cases and define the test conditions.
Test Manager	■ The test manager manages the team. He is responsible for the budgeting, planning and organisation of all test activities.
Test Team Leader	■ The test leader manages the test team. He is responsible for the budgeting, planning and organisation of all test activities in the team.
TPA	■ Test Point Analysis
WBS	■ Work breakdown structure

Appendix C
Content of attached CD

The content of attached CD is:

- Bachelor thesis in PDF format.
- Source TEX files of my bachelor thesis.
- Survey in pdf format.