



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

*Escola Superior
de Tecnologia e Gestão*

Engenharia de Software I

-Process of Software Development-

Authors:

Tomasz Busłowski
Tomasz Suchwałko

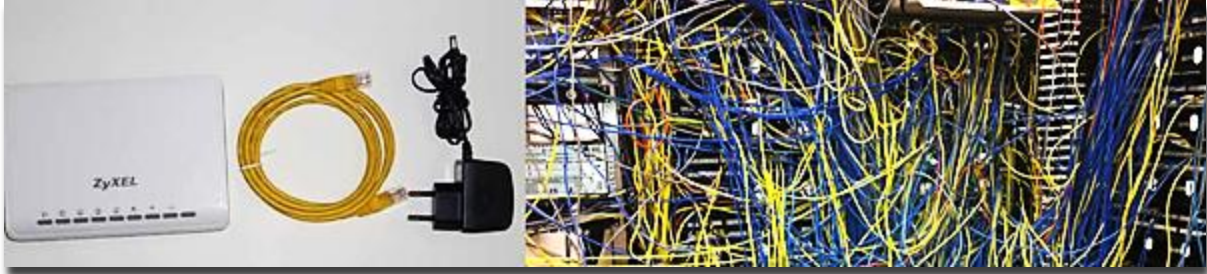
Teacher:

Professor Rui Gomes

Viana do Castelo, 10.10.2018

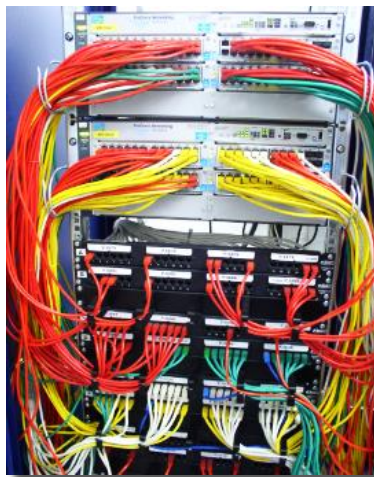
1. Introduction

Small project = small problem. Big project = big problem.



Pic. 1 small and big project

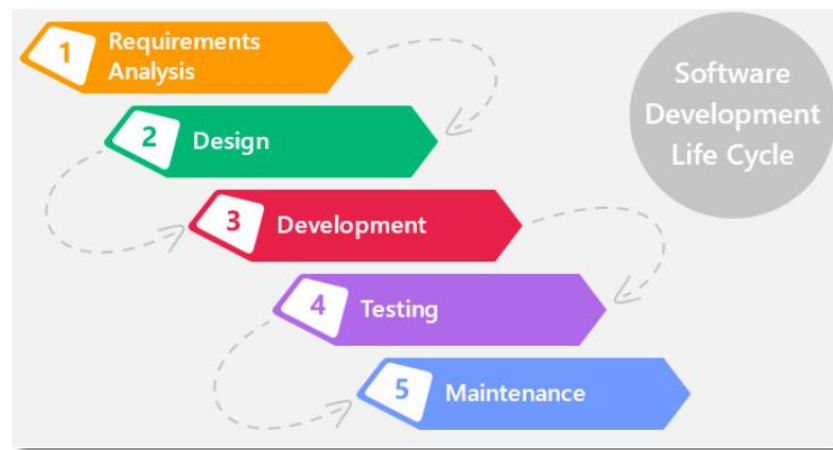
Planning and project management are very important. These elements often have great influence on success or failure of the project. If you want to do simple project probably you can make a plan in your mind start coding in few minutes and improvement your solution in real time. For a little bigger projects you need a white piece of paper to write down main goals. In practise when you work in company, projects are big or often huge. And It's impossible to work with them without any plan, strategy, documentation. Imagine yourself that you are new in project and you have to improvement some functionality or add a new feature. There's no documentation, no diagrams showing how system works. There's only thousands of files, million's lines of code and You. You in a big trouble. That's why software engineering is. It's helping for developers to understand project goals and system activities. Diagrams are very helpful for developers, project directors etc. but are good for clients too. During meeting with client, project manager can better explains system features. In next part of this work we will describe process of software development but the main goal of this concept is to avoid situation showed on a picture above with maze of wires. Using models supporting software development, keeping good project documentation should cause that our project will recall picture below where structure of the project is organised and introducing changes in system is not a mission impossible.



Pic. 2 organized wires

2. Main phases of software development.

Imagine that you are a project manager and you are responsible for creating and deploying a big project. Your client is Millennium Bank. So the first thing you need before starting work is knowledge what exactly you have to do. What features your system will have, for who it will be, how many people will be using it, how should looks main page of website etc. It is very important to get many concrete informations from client about it and confirm requirements. After that you're designing project, then developing and testing it. Your work can be done here or you could be responsible for maintenance too. So it looks like on picture below. Software development always consists of these phases. Depend of approach there are different connections between phases however. And that's why we have a few models that support the process of software development.



Pic. 3 phases of software development

Characteristics of each software development phases.

- **Requirements Analysis**- this phase is often undervalued but it's very important and when we made mistakes here we will feel the intense consequences in the next phases. So, here we make project's specification. We determine which services our system will offer, what are the limitations of project creating and working. We set budget, estimate time and costs of project realization. We studying risks.
- **Design**- in this phase we're mapping requirements determined in previous step to structures, diagrams, methods. The effect of this step should be concrete instruction for developers how to implement the system. Now they know which methods, objects are obligatory and they can start implement solutions.
- **Development**- the heart of Software Development. Something that Tigers like the most. Here our imaginations, visions of the project that we described and designed in previous steps are realising. Like after describing how our house should looks, how many windows, what kind of walls should have, how many bricks, woods, money, times and sedatives we should prepared(Requirement

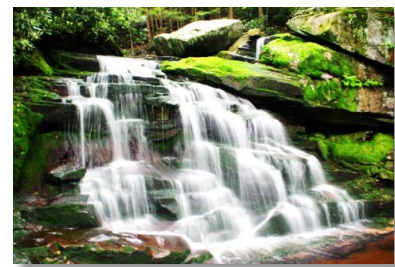
Analysis) and after making technician draws of it, making plans of construction(Design) we start building our dreamed house. The effect of this step should be a working project which meets the requirements and is ready to testing.

- **Testing**- here comes verification of developers work. Testers use application, insert data, trying to realise all possible scenarios. All possible- not only common and expected. They for try insert numbers in “Name” field. We don’t expect that people would try this but we have to be prepared for this scenario and preserve the system. So if in previous phase, developer made not sufficient solution of the task, tester should catch it and report. The point of this phase is eliminated as many bugs as it is possible at that moment so that our system was ready to be shown to the world, to the users.
- **Maintenance**- this phase relies on getting feedback from users/principal about system working and improving system. It depends of the contract but the maintenance could confines to only fixing reported bugs or improving new functionalities too.

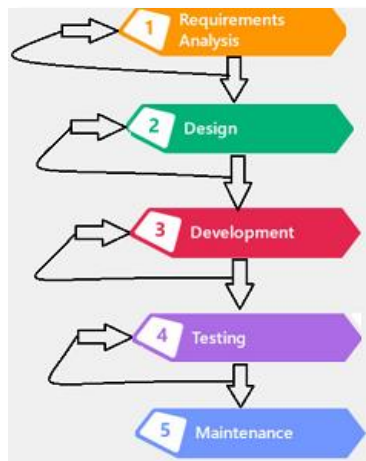
3. Models that support process of software development.

Like we said in previous chapter Software development consist of phases: Requirements Analysis, Design, Development, Testing and Maintenance. We differentiate few main kind of models that support process of software development. In each of them there are different connections between phases which determinate different ways of work. We’ll describe the most common.

- Waterfall model/cascade model- in this model after finished one phase we’re going to next and there’s no option to come back to previous stage. Like water flows from top to bottom and it cannot change direction. That makes that waterfall model is a good option when you don’t assume many changes during development phase and requirements are determinated. This model is using with big project usually.



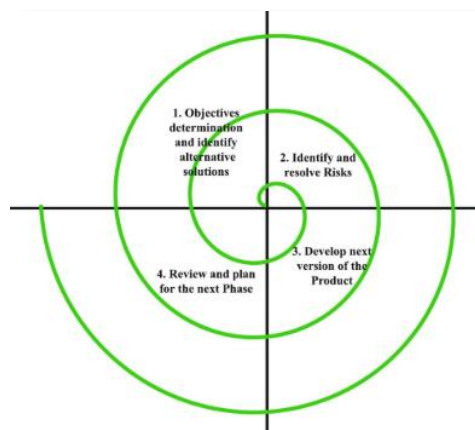
Pic. 4 waterfall



Pic. 5 iteration model

- Iteration model- using this method after you finished one phase you analyze your work, check which elements need to be improved and repeat this phase. When you and your client will claim that solution is appropriate you go to next stage of project. So it's good in small and medium projects, when specifications and requirements are not fixed and them change real-time.

- Spiral model- this interesting option is mixed waterfall and iteration model. Using this method, you're doing phases sequentially like in cascade model but after last phase you analyze solution, consult with client and (if there's need) you repeat process from first stage.

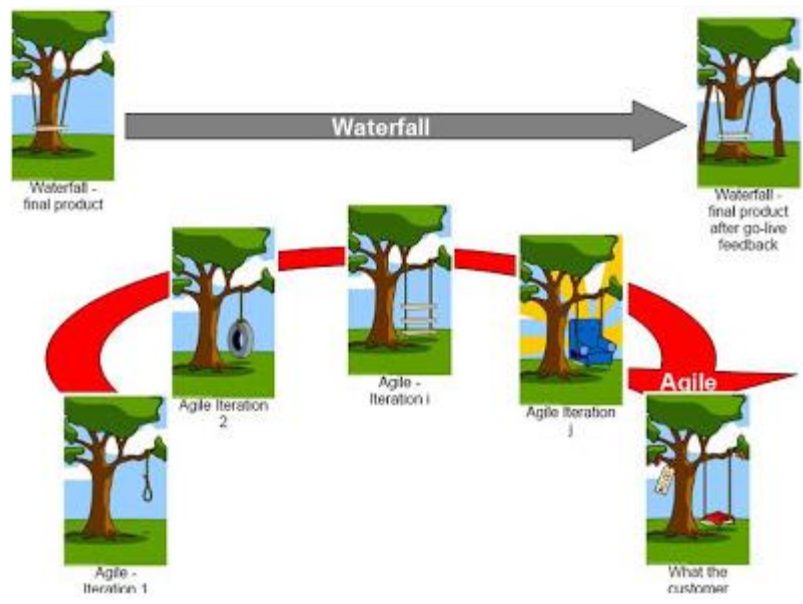


Pic. 6 spiral model

- Agile- it's not just model of software development. It's kind of methodology of it. In this approach we rejected waterfall model and using iteration instead. We don't look so far in future, we consider whole project as multiple versions and every version is a working project. How does it work? We determine requirements which should be achieved in new project version, we do it, testing and deliver working updated system version. The big advantage of this approach is the client has working version of project relatively quickly and this version is improved in next releases.

4. Models of software development- summary.

The waterfall model is utopian. It's been using there where requirements are good understood and they're not change radically during development phase. In this model client has to wait long time for working project and he has limited possibility to changing requirements. Currently technology is growing very fast, and people want speed solutions. Agile delivered working version for client relatively fast and improving it by 'sprints' making new releases. That's why this methodology is so popular now. This significant different shows picture below.



Pic. 7 waterfall vs. agile

In funny way we see possible consequences of using waterfall and agile methodology. In first way, when project requirements were bad, the whole project after long work made is bad- life verified that solution needs to be changed. Could we avoid this situation? Yes. We could do not make the whole idea of our project with full functionalities. We could use agile approach. So thinking this way, we firstly imagine that we need some line with loop dangle from the tree. This is first version of project. After that we could see solutions and claim that it would be better to replace loop with wheel. If new version wouldn't be satisfying, we could require new solution. In real-time, after life verification.

5. Bibliography

- I.J. Information Technology and Computer Science, 2013, 03, 40-48
Published Online February 2013 in MECS (<http://www.mecspress.org/>)
DOI: 10.5815/ijitcs.2013.03.05
- website: <https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc>
- website: <http://www.guerillaprojectmanagement.org/2009/01/beneficiile-agile-pentru-project.html>
- website: <http://www.agile247.pl/podejscie-iteracyjne-oraz-przyrostowe/>
- website: http://smtvsat.pl/pawelb/io_ALL.pdf