Linnéuniversitetet
Kalmar Växjö

2DV608 – Assignment 0
Software Design

Author: LEONARDO PEDRO
Semester: Spring 2020
Email: lr222qp@student.lnu.se

TASK 1 – *"No Silver Bullet"*

"No Silver Bullet - Essence and Accident in Software Engineering" written by F.P Brooks is based on the authors opinion and talks about the "Silver Bullet" and how no piece of software created in the past is error-free or completely reliable and that means that there is a need in the SE market for a solution that produces a simple and a more reliable piece of software.

Over decades, we see no silver bullet and no single development that has improved in Productivity, Reliability and Simplicity thus the author talks about essential difficulties and how these problems make it so hard to find the "Silver Bullet". Since the software is not being developed to target conformity, changeability, invisibility and complexity these essential difficulties remain in our days and that just shows how important software engineering is becoming and how much harder it is to keep up with the advancement of hardware technology to the extent that software engineers do not bother to build software that can match up the capabilities of the current hardware capabilities and that's why Brooks says "We have no silver bullet that makes software drop as rapidly as computer hardware costs do".

All software construction involves Essential and accidental difficulties, and at its core these difficulties are Complexity, Conformity, Changeability and Invisibility.

Three core problems are Complexity, Changeability and Invisibility.

Time-sharing is one of the ways of mitigating Complexity, it preserves immediacy and hence enables us to maintain an overview of complexity but we can also use Unified programming environments to mitigate Complexity since they attack accidental difficulties of using programs together by providing integrated libraries, unified file formats, pipes and filter make it more simple and reliable.

As to everything in the world, software has a lifetime and so it reaches a time where it has to change and evolve to something better fulfilling all the needs exiting, and the power of a software in charge of changing and adapt during the program lifetime is called Changeability and can be mitigated using Expert Systems where the changeable material parts are encoded in the rule base in a uniform fashion and tools are provided for developing, changing, testing and documenting the rule base.

Lastly, Invisibility is one of the core difficulties since software is abstract and sometimes hard to express, we can use some strategies to mitigate this issue, Object-oriented programming is one of them since we allow the designer to express the essence of the design without having to express large amounts of syntactic material that adds nothing to the project.

TASK 2 - The Design Question

In this book Brooks talks about how the process of design and how design can improve rapidly by helping the client understanding and knowing what they want, providing both personal and academic evidence in support of this.

One of the problems addressed in the book was the fact of not knowing the end goal, since clients can become a bit annoying and indecisive, always changing their minds about their product.

For a fact, last year s project in Computer Science I was part of a team, of 8 elements, and we were in charge of making a website to sell products online. At that point, I just had passed java2 so I wasn't that comfortable with coding and we had to learn 3 other new technologies to develop the website so I was feeling overwhelmed and I had no clue how to start, just the feeling that you don't know what to do or how to do it is frustrating and sometimes can make projects just follow some brutal paths where which element of the team works in the project with their way of viewing the website. Luckily some of my teammates already in masters had way more experience than the majority of the team so they kinda took control and leadership in helping the more in need explaining and manoeuvring the meetings we had.

Now I understand that you always need someone that knows what it needs to be done and that guides the whole team making sure everybody is getting the thing and that everybody is gonna work on the same track. So we had this meetings to build up the skeleton of our project where we set what we wanted to achieve and how we wanted to approach the website as a whole and was this design that they presented to us that makes us know what requirements we need to nail, what resources are available to us, whats the deadlines for every iteration, how many iterations we gonna have and so many other things that needed to be discussed and organized before everybody start to work individually.

As the book explains to us, planning the whole design process before starting coding or formal drawings avoids many troubles and much-wasted effort and now I can understand why.

Another problem I had was relating the design tree since we don't know it beforehand, we discover it as we go and I encounter with this problem last year doing a hangman game where I didn't knew what I wanted in my game, what features to include or not, how to structure the whole program and make sure I got all the requirements right so I just ended up doing bit by bit, starting with the main topics I wanted to hit in my program and whenever the next step arrived I decided then and I would construct my tree until the end of it.