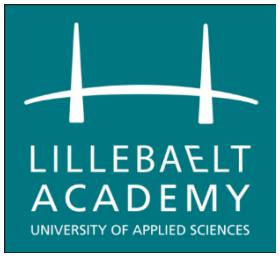
IRL Project

In cooperation with:

QuickOrder ApS



Made by:
DONATO R. DATTOLI, BOGDAN LULIAN, ATMAR KOHISTANY, PEDRO ORTEGA

Table of Content

Introduction

We are a group of four passionate international students from the Computer Science of Lillebaelt Academy that currently attend their 3rd semester and we are trying to use our knowledge to help our company improve the way they are managing a project from beginning to deployment.

The purpose of this report is to present an analyse of how the company is handling the development process from the point of receiving the deal to deployment and present them suggestions on how they can improve the quality of their work where it's necessary.

Our goal is to exercise our analytic skills and our knowledge on a concrete situation, with a real company that produce and deliver reliable software to it's clients.
The suggestions that we will present throughout the document have the foundation on different popular software development techniques that we are familiar with and they should be seen as an opportunity to improve the quality of the work.

REQUIREMENTS GATHERING

In any organization, defining and structure a plan is the first step to organize a project. Not having it my can cause many problems and can harmed the main goal of the project. The plan should contain the main objective, the propose of the project, the role of the project members and give an overview of the direction of the project.

The first steps to build a plan (in software projects) consist to defining what are the desires of the customer and create a requirements document. Having this will be easy to see what the customers will be getting from the project and what the project members will be building. During the project, both customers and team members can check the requirements to make sure that the projects is heading to the right path. If some suggestions come up you can check if that was included in the requirements. If not, you can agree to include it if at any case, will be useful for the main goal of the project and also if will not interfering with the current schedule.

Customers needs

According to our meeting and the feedback we kindly heard from Lucas regarding the way the company is organized, we found that they write down the customer needs, but without following a specific pattern and with no formal documentation. We also realise that both Business and product documentations are in some way, limited and they are prepared by the manager/salesman. This is something that doesn't be considerate a bad practice, especially if the company is a "start-up" when in most of the cases, the most

important is to get the final result as fast as possible. Anyway, we would recommend that in future projects will be important to have written down the customer's needs in detail.

Changes/suggestions

In other hand, Quick Order ApS has a very good approach regarding the suggestions or possible alternatives that they may present to the customer as normally, customers don't know quite well if it's possible to do. Regarding analysing a specific company and the documentation for it there's no necessity for such approach once the company is used to work with this particular market (restaurants). The main approach is trying to improve the current platform, understanding what is the best existing solution among restaurants by comparison existing systems of other companies.

Communication

The communication between the company and the customers are not formal, which can allow some flexibility. Main methods are phone or e-mail.

The next step

After establishing and understand what are the customers' demands and wishes, the salesmen get feedback from the customer, meet with the developers team, do a brainstorming session and defining the features to give priorities to. After this, they split the test in smaller pieces on utrack putting one piece per day and making sure that all the tasks and duties are finished according to the deadlines.

Rules to follow

Regarding the developers and the group members, they are free to use the tools that they prefer except for utrack.

The have external deadline for the customer (about 1 month) and internal deadline that is half of the time. This internal deadline is a very good approach one it can provide a margin of error in case of any delay or anything that was not predictable.

Communication inside the team

Regarding the communication inside the team they use Stack. Is a specific solution very similar to messenger, but is applied regarding business. They are used to it and it's the main communication tool inside the teams.

Code sharing and information.

They often can use the information regarding previous projects to modify or used as a base for the upcoming projects. That information is placed in *utrack*. About the code, in order to keep everything organized they used *Github* as the main tool to be storing repositories.

High Level Design

Hight-Level Design Definition

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. We use high level design to understand what should be in the system, not how the system works.

Analysis

We have analysed the company from the very first part of the High-Level Design: the hardware.

By hardware we mean the hardware choice by the company in term of target/platform. The company has been trying to work on the same platform, IOS, over years, where the primary device is the ipad.

They have chosen this platform since the very first developers had experience in using Swift, but also because they think IOS is reportedly, on average, safer than other platforms. Furthermore the Ipad as main device is really cheap to buy, meaning the possibility to reach as many customers as possible. It's also worth mentioning how the company is open to new platforms like android.

Security

Another important part of the high level is the security, which is currently not a priority for the company. It believes in being able to handle the security issues without outsourcing them. For the security they primarily rely on the best practices and experience in building the system. In particular they are used to create backup copies and rely on the amazon server that allows an automatic switch to other servers, in case the previous one was having any issue. It's useful in order to avoid the inactivity time, which could damage the customer business. Beside, they keep resetting the database in order to not to keep in memory useless data and, at the same time, protect the informations.

Usability

The true priority is about functionality and usability. Even if there isn't any formal manual to ease the customer's everyday application usage the company set up a training system that start from the very first time at the workplace. The first step is a training session simulating a daily use of the application, so that the customer can train by doing with a real time support.

The basic understands and knowledge to manipulate the platform is offered by the company to the customer in order to be able to use the system right away. After the first training the company is available on daily basis to receive contacts from customers such as calls or emails, regarding whatever issue the customer reports. Any bug or eventual failure can be immediately fixed.

Why not ?

The company is indeed not following any type of plan over the long period: how can the company knows, at any moment, about the current and future choices ?

There isn't any formal documentation, therefore the developers can't follow a specific set of values the company want to create for the customer. For example a documentation that specifies the "minimum requirements" in term of functionality, usability, reliability, performance and security. If the developer and any other member of the organization knows in advance the aforementioned values then it would be easier to work staying oriented toward the customer needs, at any moment.

As mentioned the security is not a main priority for the company, but every issue now and then could have a serious damage overtime. They should indeed analyze the possible risks, external and internal, the causes of these risks and the damage when the negative event occurs. If the company knows about the possible damage then it's easier to plan a possible "plan B" or even C in order to lower the impact as much as possible with ease. Imagine if, for whatever reason, the customer is not able to use internet for one day, what could the impact be on the business ?

Harware

Another important point is about the Hardware. The company has demonstrated being capable to work on a specific platform but is it enough flexible ?

They should think of the future. There isn't any documentation to help the future developers to understand about the current project, how the logic works and any explanation. Meaning that if the company want to switch to other platforms not only would take time to train new developers but it would not have any documentation to train the developers about the current platform.

Last but not least: aside from the personal experience and learning by doing, does the company have any documentation in order to analyze the customer business(for example strenghts and weaknesses), better understand the problem, gather the informations about the customer needs ? A such documentation is very important because could allow the company to make better high level design choices, closer to the real customer needs.

LOW LEVEL DESIGN

Our weekly meeting with the company about Low Level Design. Before that let me explain what is LLD (low level design) and why it is impertinent?

Definition of Low level design

Low Level Design is like detailing the high-level design in a much comprehensive way. After we are done with the high-level design for our project, we need a graphical representation of the structure of the software system.

This way we go deeper in the processes by writing the low-level design documentation.

A document that defines high-level details of each module in your system, including a module's interface (like input and output data types) , notes on possible algorithms or data structures the module can use to meet its responsibilities and a list of any non-functional requirements that might impact the module .

For example: are the events referenced in a sequence diagram defined as methods on the objects shown in the sequence diagram?

Why low-level document?

Low level design document provide traceability to the main idea and structure of the product that should be developed by developers. LLD defines the actual logic for each and every component of the system that we consider in the high-level design. For instance, Class diagrams with all the methods, Properties, variables and relationship between classes are part of the low level- design.

Since LLD is very technical, the aim for LLD is for developers use than the system users. Low-level design not only help the developer to start writing the code, it also helps them to work in groups better if they divide the task in small parts.

This is an overview of our meeting about low level design and how well the company is going to use LLD in their project at the moment.

In the meeting, we find out that the company doesn't use low level design documentation for their project. The reason because the company is small and they have only two developers, so instead of writing a comprehensive LLD documentation the two developer discusses about the project and it's needs.

Although the company is not using LLD in their project it doesn't mean that the company is not doing will. They are successful and growing fast.

at this stage, the company is in not facing any big issue during the projects. But as I mentioned the company is growing fast, in this case they need more developers in the future,

Long term benefits of low level design.

Why it is good for the company to use LLD in future and how they benefit from it?

1. low-level design documentation is like a blueprint for developer while they start coding and if something not working as it should be. (LLD) will help them to see if they are missing something.

2. A good Low-Level Design Document developed will make the program very easy to be developed by developers because if proper analysis is made, and the Low-Level Design Document is prepared, then the code can be developed by developers directly from LLD with minimal effort of debugging and testing. This helps the development team to give quality product.

Development and testing

After all the information is gathered and the design is well defined, the development part takes action meaning that the programmers can now go to work on the actual code. In the same time, the programmers should test the code in order to ensure that it's accomplishing it's goals and to remove any bugs may appear. Usually, after the a piece of code is tested by the programmers, it is passed to specialised testers for more thorough testing before integrating it into the project

.

Hardware

An important part of the process is using the right environment with fast and reliable hardware that will allow the programmers to execute any request as fast as possible. QuickOrder is giving the developers the freedom to use their own computers as well as deciding upon the hardware they want to use and also they can ask anytime for new hardware if it's necessary.

Network

Sometimes even the best developers on the market can have problems in reminding everything and even a small and trivial problem can hold up the entire process. The developers inside the QuickOrder are allowed to use the internet freely to search for any help they need.

Working environment

Using the right environment is crucial for the developers because, at the very least, it is translating their code into something that the computer can execute. There is a huge variety of IDE's capable of satisfying all the needs of the developers. At QuickOrder, developers are using the PHPStorm IDE from JetBrains along with Phalcon framework plus a variety of extensions and tools such as CodeCoverage, CodeClimate and Travis CI. Also, PHPStorm does an excelent job on helping the developers with formatting the code using its build-in plugg-ins.

Source Control

Source control is essential for the code in the way that even a small change can harm the program. As a programmer, you may need to go back to old versions of the program if the current version stopped working. For this, QuickOrder is taking advantage of GitHub where the developers commit changes where they must use a clear description on what has been changed.

Refactoring

Usually the developers are spending the most amount of time on refactoring the code in order to make it easier to understand and more maintainable. This is also the case of QuickOrder developers who that take advantage of their IDE that helps them in the process. They are spending a considerable amount of time on refactoring in order to ensure it's quality mostly on the application itself. Database it's not requiring so much work but when it does, they are using column mapping to ensure the backwards compatibility.

Testing

Testing is an extremely important part of the development process because it allows you to solve the majority of the bugs that the program may have before it's shipped to the user. The developers at QuickOrder are spending significant time on testing the code using different techniques and tools such as PHPUnit for unit testing, TravisCI for integration testing and AWS CodePipeline for automated testing.

Conclusion

Regarding our cooperation with Quick Order, we must say that was a pleasure for us have this possibility to get in touch directly with an IT company, specially when is a start up with young and promising people, who decided to go forward with an idea and they are successful with a great future. During the entire project, all members of our group felt that this opportunity was challenging and demanding, but we had full support from Quick Order, from Lucas Sørensen and Andreas Aagaard. They were very kind and they have answered our questions without problem. We came up with some suggestions that we have presented in this report, in each section, hopefully those suggestions can be implemented in the future, in some way, in the company.