SWEN90016 Software Processes and Management – Assignment

Virtual Surgery Project Analysis

Name: Zhe Tang

LMS login: Ztang2

Tutorial time: 11:00 -12:00 Wednesday

1. Introduction

Virtual Surgery project is a project, which requires building a software, that can be used to simulate a temporal bone surgical and examine the bone structure, as a tool for surgeons and Diagnostic Personnel. To finish this project, Team Aura, which consists of 17 members, adopts a prototype-driven, incremental software development lifecycle model.

Based on the software requirements specification(SRS) and software project management plan(SPMP), this essay will analyse the most important risks in the project. Then it will give requirements which make the project hard. Also, comparison will be made between several models and the model chosen by Team Aura. At last, conclusion will be given.

2. Risk

Almost all software projects have risks, which will affect the completion of the project. In that case, risk management is important. In this project, SPMP (Team Aura, 2011, Page 17) Shows "Due to the inflexibility of the delivery date, the project schedule will assume a higher priority than the system requirements." Therefore, to finish the project on time, risks in this project need to be well controlled.

In my opinion, the most important three risks in this project are client, technology and team.

2.1 Client

"The clients are somewhat vague about the exact functional requirements for the system, and the kind of hardware the system needs to run on." (TEAM AURA, 2011, SPMP, Page 5) In that case, there is a possibility that users need more time to modify and examine the prototyping. Although the team uses a throwaway prototype to reduce this risk, there is still not an effective way to deal with changing requirements after finishing the prototyping. It cannot make sure that the software following the prototyping can meet the real need of users as well, even users sign off. However, the requirements sub team will disband after completing requirement analysis. Therefore, if clients want to change after prototyping, there is no team work for this. This action may lead to dissatisfaction of users.

2.2 Technical feasibility

"it is unclear which of several possible approaches would be most appropriate to give the level of performance required." (TEAM AURA, 2011, SPMP, Page 5) The project is done by students, who are assumed to be inexperienced. Because these students are unfamiliar with tools and programming language, they need to spend time to learn that. Also, when they build prototyping, they cannot make sure that which method is most appropriate. Hence, students in prototyping sub team need to learn all the methods. Even if they choose one, the approach may be proved not suitable for all functions. Also, the team chooses the incremental model. They split the requirements into several incremental, then combine the incremental and test. There is a risk that these incremental affects each other and cannot combine well, lacking integrity.

2.3 change of team

In the assumption section of SPMP (Team Aura, 2011, Page 17): "It is assumed that there will not be a significant change to the team during the project." However, this assumption is too ideal. Team members have different roles in a team. Maybe there are some team members taking a role of free rider. Besides that, students in the same team have never worked together before. In that case, some problems may occur in the process of the project. For example, the speed of learning new technologies by students in the same team is different. Also, team members do not know working habits and the ways of communication with each other, which may lead to conflict. Also, it is difficult for manager to manage the team. These possibilities will affect the completion of the project.

3. Requirements

There are many requirements given by clients in SRS, like the appearances of application, the functions of applications, documents and constraints. This project lacking measure metric at some level, so some requirements are difficult to be met in this project.

3.1 Quick, continuous and real-time

"The performance of the system is important. It would be ideal for the system to refresh quickly and provide real-time performance especially while the user is drilling the bone." (TEAM AURA, 2011, SRS, Page 16) This requirement is difficult because it requires real-time refresh and do not have suitable measure metric of performance. The interactions have limits that only when users do actions in limited drilling area, the screen will refresh. Also, the screen refresh needs to be quick and continuous. It is a high requirement to software. The requirement needs to set refresh time appropriately and some other actions to guarantee the quick and continuous refresh. Also, the refresh cannot affect the operation of users.

3.2 Extendible

"A critical requirement is for the design to be easily extendible. Hence, the design should be as modular as possible." (TEAM AURA, 2011, SRS, Page 16) This software is the first step of a sophisticated virtual surgery tool, so it need to be maintained and developed easily. But this project use incremental model. It is easy for the software to lose integrity. Also, the parts of this software have dependency. It means if users want to extend the application, software engineer may need to change other parts. Therefore, the software is difficult to be extendible.

3.3 Error handling

The error handling in this project is important and difficult. Firstly, it requires no error occurs which will affect the users' operation. Then the errors need to be displayed and be realised how to be fixed. Lastly, the system should be backed up real-time to guarantee the interaction from the point that error occurs. (TEAM AURA, 2011, SRS, Page 17) However, in the software, sometimes it is difficult to figure out all errors. Also, errors occurring relays to the actions that users take, the system and the failure of hardware. These errors are difficult to be figured out, controlled and fixed, especially in the process of surgery.

4. Alternatives

In this project, Team Aura choose a prototype-driven incremental model, a throwaway prototype and controlled decentralised organisation model as the software development life-cycle model, the prototyping model and the team structure model separately. They split the requirement into two part: core requirements, which are essential of delivery, and non-core requirements, which are chosen to be completed follow the prioritisation.

Because some non-core requirements in this project are the extension of core requirements, I think the models with iteration are more suitable than Waterfall model and the V model. In that case, I will compare the Incremental model with the Spiral model and Agile approaches.

4.1 Spiral

"The spiral model is a type of iterative model in which each iteration has distinctive sequence of activities that are designed to manage risk." (Lecture notes, chapter 3, 33) In Spiral model, prototyping is also involved. The main feature of this model is risk management. It allows completely re-planning of project direction and evaluation of risk.

4.2 Agile

The agile method are not formal software development lifecycle models. It aims to offer a way which can be used for easily changing requirements and quickly delivering software. This model emphasizes code more than formal process. The Scrum methodology is based on agile and is widely used in industry. Based on the advantage of agile, the scrum methodology contains a daily meeting for team member to know others' work and track the process of the project easily. Also, finishing each sprint must meet the constrict time limit.

4.3 Comparison

In my opinion, the incremental model is more suitable for this project.

Firstly, the time limit of this project is important. Spiral model includes risk analysis and prototyping every iteration, which make this method need more time than the time needed in this project.

Considering that almost all team members in this project are students, they are not so experienced that can meet the requirements of scrum. Scrum model requires team members to do actively work, and there is a challenge to measure the performance of team.

In terms of client, spiral and scrum models need clients to be closely involved through the whole project. It reduces the risk that clients want to have changes after prototyping in incremental model. However, communication with clients and always changing requirements will cost more efforts of team, which will increase the load of the inexperienced team members.

Incremental model has some drawbacks, e.g. lack of integrity. However, each model has its own drawback. Compared to spiral and scrum, I think incremental model is the most suitable one in this project.

5. Conclusion

Client, technology and team are the most important risks that I think in this project. Also, it is difficult for the system to be continuous, be extensible and handle errors. Though incremental has some drawbacks, it is more suitable for this project than others.

Reference:

Team Aura. (2011, April 7). Software Requirements Specification.

Team Aura. (2011, May 17). Software Project Management Plan.

SWEN90016 Lecture Notes 2017. Chapter 3 Software Development Life Cycle Models