

SWEN90016

Software Process and Management

Report for Assignment1

Changheng Zhou **StuID:**1290706 changhengz@student.unimelb.edu.au

March 27, 2023

1 Question1

The current IT system cannot satisfy the needs of the company due to the following reasons:

1. Long-time phone calls are needed for customers to express their demands.
2. Comparison of multiple configurations can only be expressed via long-time phone calls.
3. New technology offers can only be updated once a month and require manually contact with the customers.

Therefore, a new IT system aiming to solve these problems needs to be developed. The new IT system needs to be customer-centric and contains the following features:

1. Customers can directly configure and access their services.
2. A 'Guided Buying and Selling' can help new customers seamlessly answer non-technical questions on the user interface and present them with the most appropriate options to select for their business continuity needs.
3. A 'Business Rules Configuration' that can show appropriate configurations to customers and has a graphical user interface for adding and editing configurations.
4. The system can also compute potential optimal configurations and generate an email to inform customers about that.
5. The system can also provide cloud backup and recovery services to customers.

2 Question2

Business Rules Configuration can be very difficult since:

1. The accuracy cannot be guaranteed, and the user experience cannot be very good due to the complexity of this function.
2. The needs of different customers vary a lot from each other; the rules can be difficult to generate or too complex. The rules are not guaranteed to cover all potential scenes.
3. Some needs or scenes cannot be expressed through a few words or the 'rules', but covering them may make this system too complex for most cases.
4. The graphic interface for adding a rule should support all the fields defined in the configuration. The page should not be too long in most circumstances for convenience but need to expand to show all options. Multiple layouts may need to be developed for various situations.
5. How to deal with the special needs of each customer? For example, the video editing configuration may vary a lot from the camera they deployed.

With the changes and upgrades, to begin with, generating an email that can briefly and clearly express the potential better solution without too many words or terms is very challenging. The properties of computer hardware cannot be simplified into a few numbers and words. For example, the performance of solid state drives cannot be simplified into speed and capacity; it contains many other specifications such as writing speed with a full buffer space. Another example is that performance of CPUs and GPUs can vary a lot for different applications. Therefore, recording the performance of various hardware is quite different. Then, the algorithms for generating potential solutions could be hard to implement. How to ensure that the potential better solution is actually an optimal one instead of costing too much and being less capable? How do avoid compatibility issues? How to ensure the email can persuade the customer to make a change to their current configuration?

Implementing an effective backup and recovery cloud application is another constraint of this project. For the cloud backup and recovery implementation, we all know this process can take quite a lot of computing and network resources. It is hard to balance between the frequency and resources of cloud backup and recovery. Also, the application has to ensure that the backup and recovery really work and helps during various situations. For example, during a power cut down or system crash, the cloud backup and recovery system should still function. Nowadays, customers put their sensitive and critical information on their devices, the cloud backup and recovery should have some security methods to ensure the security of the data, which can also occupy extra computing resources.

3 Question3

Network security issues could occur during cloud recovery and backup processes. Also, during this process, software bugs and crashes could occur, for example, how to respond when a recovery loses some data of the previous status. This can interrupt the commercial process of the customer. In addition, cloud recovery and backup need to occupy some resources of the customers' hardware, such as computing resources and network bandwidth, which can further lead to dissatisfaction of the customers.

When it comes to data migration from the current IT system, the data format of the two systems can be quite different. Inaccurate, missing and even wrong data could occur in the new system. These issues

could lead to a loss in quality and progress of the customers' projects, affecting the reputation and trustworthiness of the company and forcing some customers to switch to the competitors' service.

Risks can also happen for the Business Rules Configuration system. For example, if some rules are wrong, there are too many potential rules or configurations for the customer's needs which makes the customer confused about their choices. If some rules or recommended configurations are wrong due to employees' mistakes, customers may be very unsatisfied with the results and user experience. These potential risks can also lead to the loss of potential customers. The final configuration can be too expensive or incapable, leading to customer dissatisfaction.

4 Question4:

4.1 Waterfall

Waterfall can be a possible Software Development Lifecycle model for this case study. We have a clear knowledge and understanding of the requirements and detailed possible solutions for this case study[1]. We need to design a 'Guided Buying and Selling' engine containing a 'Business Rules Configuration' sub-engine with an interface for complex configurations, a system called 'Configuration Change' that can discover new hardware solutions to customers with an auto-generated email, a system called 'Fixed Configuration Offering' offer fixed configuration for small businesses and a cloud backup and recovery system to tackle the potential risks and accidents. These requirements and implementations are quite clear and common. Therefore, we can have a clear understanding and knowledge of all the requirements at the start.[2]

The waterfall approach is a traditional and wide-known approach. Since the company was founded nearly 20 years ago, it must have quite a lot old-fashioned software engineers that are quite familiar with this SDLC choice. When the company want to hire additional employee or interneers to join in this project, it is also quite easy for them to know the structure and details of this project and then feel comfortable working with the existing team.

Due to the purely 'progressive' SDLC, we can have a clear picture of the tasks of each step due to the experience and design of the former step; this can also cut down the potential need for experienced software engineers, which means more cost and training.

Waterfall SDLC can have several drawbacks and constraints as well. We need to be careful about these setbacks[3].

First of all, either the manager of the company or the clients cannot see the outcome or results until the final stage of the waterfall. If the team have some misunderstanding or ignoring of the features at the beginning of the project, then these faults can only be discovered at the final state, which means the whole process of waterfall SDLC needs to be redone, leading to a waste of time and money. Any fault or mistake in a previous phase can affect the later stage profoundly, and the only way to fix it is to redo all the steps before.

Secondly, if the company wants to make some changes to the initial plan due to recent experience in

businesses, for example, different separate systems for different kinds of potential customers. Since all the sub-systems are some kind of connected to each other, for example, the 'Business Rules Configuration' and the 'Rules Design User Interface' are absolutely related. That leads to that redoing one means redoing both. Therefore, modification requests from the manager or customer also mean coming back to the start point, costing more time and money.

Since customers and managers can only see the results in the final stage for waterfall SDLC, that means they may lose patience and be annoyed about the delays and costs during the development phase.

4.2 Agile

Agile can also be one possible software development life cycle(SDLC) for this case study. The project can be divided into these steps. One is developing a shopping and recommendation system, which refers to Guided Buying and Selling, Business Rules Configuration, and Rules Design User Interface systems. Another two are the cloud backup and recovery system and the data migration. We firstly implement this section to get this system to work. And then implement another one and combine it with the existing one. We can add some unimplemented features such as fixed configuration after that 2 stages.[4]

The whole team can deliver an operable segment as soon as possible so that the customer and manager can see the effect and implementation in an early phase. Therefore, the customers and manager can give their feedback on the product in time. And the team can work to respond and fix as fast as they can. The cost of these modifications can be quite low since only part of the whole project is implemented. For example, the team can get the Guided Buying System done at first and listen to their customers' voices. This does offer not only customer feedback being responded to in a short time but also gives great flexibility since the product can be modified at any time.[5]

Since the initial delivery of the product is much faster compared to traditional waterfall SDLC, customers and managers can at least use some of the important functions in the early stage. With the functions being added, the customers can gradually be used to the new system. And this can also make the customers and manager more optimistic about the entire plan.

We can also begin debugging and testing at an early stage to avoid a huge debugging and testing workload in the final stage. One bug in the Business Rules Configuration can affect the Guided Buying and Selling and Rules Design User Interface. If most bugs and problems are correctly settled in the Business Rules Configuration system, then the testing and debugging in other stages can be much easier.

The roles and responsibilities of each member can switch and change as the process pushes. This can lead to the complete utilization of human resources.

The constraints of the agile SDLC can be in these fields. To begin with, all the team members need to know the structure of the whole project, the interfaces between modules, the data format of the system and the detailed implementation of other team members. This could be quite demanding for software engineers, which means it is hard for junior developers to participate in the project. This company is not a top-rated Internet company. Traditional companies like this do not have many resources in software development, especially employees that can meet the demands for fast and heavy software development. Existing members of the team can also get tired and exhausted due to long days of the heavy workload

of different kinds. Thus we need to pay a lot for experienced and full-stack developers, causing a huge increase in the cost[6].

For example, engineers specify in testing bugs and designing UI for the ‘Guided Self Buying’ system cannot meet the demands for implementation of the cloud backup and recovery system. And they have to devote into the new task after one task is finished, which is pretty tiring.

This project is quite a big one, so there could be lots of problems for members communicating inside the group since their characters can be changed from time to time.

References

- [1] S. Balaji and M. S. Murugaiyan, “Waterfall vs. v-model vs. agile: A comparative study on sdlc,” *International Journal of Information Technology and Business Management*, vol. 2, no. 1, pp. 26–30, 2012.
- [2] Y. Bassil, “A simulation model for the waterfall software development life cycle,” *arXiv preprint arXiv:1205.6904*, 2012.
- [3] V. Rastogi *et al.*, “Software development life cycle models-comparison, consequences,” *International Journal of Computer Science and Information Technologies*, vol. 6, no. 1, pp. 168–172, 2015.
- [4] J. Newkirk, “Introduction to agile processes and extreme programming,” in *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*. IEEE, 2002, pp. 695–696.
- [5] J. de Vicente Mohino, J. Bermejo Higuera, J. R. Bermejo Higuera, and J. A. Sicilia Montalvo, “The application of a new secure software development life cycle (s-sdlc) with agile methodologies,” *Electronics*, vol. 8, no. 11, p. 1218, 2019.
- [6] V. Rastogi *et al.*, “Software development life cycle models-comparison, consequences,” *International Journal of Computer Science and Information Technologies*, vol. 6, no. 1, pp. 168–172, 2015.