Xinyu Jiang
CSCI5040 Final
12/11/2020

1. **In lecture, we discussed the role of architectural design in developing systems. We discussed tools like the 4+1 and C4 models for creating architectural designs. When is it appropriate to use these methods? What do you feel the benefits, issues, or best practices for such architectural design might be? Please support your statement with at least one or two solid citations.**

Architecture diagrams are a very useful technique in software development, programmers can use them to explain how you will build a software system in the pre-design period. C4 and 4+1 design models are two significant methods in architecture visualization, and the rest of the paper will analyze the benefits of those techniques and when it is appropriate to use them.

C4 model is an “abstraction first” software architecture diagram method, which is based on how programmers think and build the software(“The C4 model for visualising software architecture”, n.d.). In the C4 model, the software system is “abstracted” into four parts: system context,container, component and code. Based on the four levels of abstraction, The C4 model consists of 4 core diagrams and 3 auxiliary diagrams, which could be used in different scenarios(“The C4 model for visualising software architecture”, n.d.). Since C4 models use a small amount of chart types, it is easy to learn for programmers. In addition to that, using the C4 model to draw the architecture diagram could separate information from the frame, thus software developers could extract more metertial from the diagram and understand the content more clearly.

Similar to the C4 model, 4+1 model is a development technique that decomposes a system from top to bottom and provides a reasonable abstract description. In order to isolate and provide useful information in different aspects of design with different information oriented people, the 4+1 model decomposes the architecture diagrams into multiple concurrent views (Simon, 2019). Based on different scenarios, the 4+1 model is separated into four parts: logistic view, development view, process view and physical view. Through the four components of the 4+1 methods, software developers could form an abstract description of the system. Therefore, all members of the development team could design, implement, and verify around this abstraction, and continuously improve and modify their product.

To conclude, both the 4+1 and C4 design models are techniques that abstract the system so that programmers could extract information from the architecture diagrams. However, because of the capacity of the diagram, both the development methods could not cover all the contents of the system. For C4 diagrams, because it uses a small amount of diagrams to illustrate the design process, thus, it is difficult to provide all the information. Same for the 4+1 model, the four levels of abstraction could not reflect information about key technologies and data structures used in the system. Therefore, in software development, it is necessary for programmers to combine those techniques with other design methods to consummate the product.

Xinyu Jiang
CSCI5040 Final
12/11/2020

2. **In lecture, we discussed leadership and management of software projects. One of the references I used was an IEEE article titled "What Makes a Great Manager of Software Engineers?" (at the link https://www.microsoft.com/en-us/research/uploads/prod/2018/06/kalliamvakou-tse-2018.pdf). Please provide your observations on this article and any statements in it that you agree or disagree with, or that you have a different perspective on. Please support your statements with at least one or two solid citations.**

The essay "What Makes a Great Manager of Software Engineers?" illustrate several strategies that help leaders become successful project managers. In the essay, it expresses fifteen attributes a great manager requires with examples how those attributes are used in actual scenarios(Kalliamvakou, 2017). In my opinion, those methods are a treasure in software development, but to implement those theoretical knowledge, it is necessary to continuously practice and form an advanced style that suits you. After I read the paper and have a basic understanding about what attributes a greater manager needs to have, I will discuss my opinion about how to be a successful team manager.

There is a stereotyping thought about project leaders that they only give command to the team. However, just the opposite, managers should focus on serving employees, not just giving orders("What makes a good Software Development Manager", 2019). In order to enable all the programmers to work in their best conditions, it is necessary for a team leader to provide a relaxed and friendly communication atmosphere, so that they could share/gather information from programmers("What makes a good Software Development Manager", 2019). In the process of product development, mistakes and bugs are always unavoidable. A great team should fix the problem once they find the problems and report to the manager ahead of time. Therefore, establishing a trusting relationship between software developers and team leaders is essential in project management.

In addition to that, a solid internal quality is an important component of software management. In order to communicate effectively with developers and not give unfulfillable needs, it is necessary for programmers to have a strong technical background. Also, during software development, a manager needs to have a product architecture diagram in his/her head, and that diagram needs to be improved continuously. Furthermore, after product models are created, it is essential for project leaders to have reverse thinking ability. To strengthen product quality and improve its competitiveness, managers need to use their products, and think about what to improve in the users' perspective, so that users could have a great user experience.

To conclude, project manager is the main character in a software development team. They not only need to have solid interpersonal skills, but also necessary to include a strong technical background. In order to create a competitive product, project leaders need to aggregate power from all team members and build their product in both developer and user perspectives. Therefore, a great project manager does not just make a salary from the company, instead, they are creating value for the society.

Xinyu Jiang
CSCI5040 Final
12/11/2020

3. **Briefly describe the work you did for the sponsor project you were assigned to this semester. Identify three elements of that work that either went particularly well, were more difficult than you expected, or otherwise did not go the way you expected or planned. Finally, describe any concerns or observations you have about the sponsor project development planned for next semester. (There is no citation requirement for this answer.**

This semester I was assigned to join a graduate capstone project, the topic for my team is called "Activities of Daily Living" sponsored by Inovonics. Simply speaking, it is a project that monitors and predicts people's daily activities based on data about time and location. During the implementation of this project, our team faced a lot of challenges, but we finally figured out the way to solve the problems and finish the goal we made at the beginning of the semester.

The first challenge happened when we first started the project. At the beginning of the semester, we were assigned to the project group, our sponsor gave us a long requirement document that includes what our project is about. However, because there are too many details in the handbook, and there is no additional document, I was confused about how we were going to start the project. Fortunately, our sponsor is extremely helpful, after several weeks talking to him, I have a basic understanding of the project. Also, our team decided to separate the project by Data Generation Engine(DGE) and Artificial Intelligence Engine(AIE). For the fall semester, we are going to focus on the DGE implementation.

During the implementation of the project, we faced problems about the platform of writing the program. At the beginning of the project, we planned to write the code on our local desktop then upload our work to Github like what we did for most of the collaboration work. However, because in this project, everyone in the team is responsible for some activity functions, and we are editing the same project file. We realized if all of us are going to work on his/her part and upload the work, it will overwrite functions that the previous person did. In addition to that, even though we could create our own branch and write the program in some order, we are going to wait until the previous person finishes his/her work and it will be a waste of time. Therefore, after talking to the team, we decided to use Github only for storing our final work. All of our team members are going to use Code Colab to write the program so that everyone can write their code in the same platform. In addition to that, Code Colab solved the time wasting problem we are facing, each team member only needs to implement his/her own part, and we can keep track of the correctness of our work.

In addition to that, in the collaboration of work, I realized the importance of communication between all members of our team. During the implementation of work, we found there are some parts in the requirement document that are illogical. In that time, our sponsor is extremely helpful, he always replies to our message and checks our work once he receives feedback from us. For example, in the initial version of the requirement book, after we sum up all the timelines, we found that the simulated person sleeps around 6 to 7pm everyday. After we reported to our sponsor Eric in our weekly meeting, Eric extended each activity time immediately.

To conclude, in the fall semester, we finished our Data Generation Engine and checked with our sponsor. For the next semester, we are going to implement the Artificial Intelligence Engine. We have not split the testing and training data from the DGE we created this semester. Also, we have decided what AI technique we are going to use for the AIE part, but I am confident that we will find all the answers for those problems.

Xinyu Jiang
CSCI5040 Final
12/11/2020

References

1. Kalliamvakou, Eirini & Bird, Christian & Zimmermann, Thomas & Begel, Andrew & Deline, Robert & German, Daniel. (2017). What Makes a Great Manager of Software Engineers?. IEEE Transactions on Software Engineering. PP. 1-1. 10.1109/TSE.2017.2768368.
2. Simon, C. (2019, February 27). 4 + 1 Architectural View Model: Introduction. Retrieved December 11, 2020, from https://devcycles.io/2019/02/27/4-1-architectural-view-model-introduction/
3. The C4 model for visualising software architecture. (n.d.). Retrieved December 11, 2020, from https://c4model.com/
4. What makes a good Software Development Manager? (2019, June 13). Retrieved December 11, 2020, from https://hackernoon.com/what-makes-a-good-software-development-manager-1323464e7615