Software Methodology review paper

Abstract:

Throughout the semester we have developed software called “prescription” which offer very useful tool for effectively organize and manage their patient’s medication and information. The physician searches through the database of patient’s record by using patient-specific information such as MCP. Once the correct patient file has been populated, the physician reviews the current medication information and writes new prescription and updates it to the database. Throughout our project ”compRx” we have adopted waterfall and agile methodologies. As a result we strictly followed the four fundamental phases such as requirement phase, architecture design phase, implementation phase and deployment phase. These methodologies helped us to structure, plan and control the process of developing the software project. However if we would combined some other methodologies such as iterative with agile then the project would have been successful with great client satisfaction.

1. Introduction:

compRx project followed the waterfall model which describes a development methods that is rigid and linear. This model maintains distinct goals for each phase that we can only move to next phase if the previous phase is completed reviewed and verified. When the next phase is started we cannot turn back to the previous one [1]. This model spend time in the early phase to make sure requirement and design are correct so that we can save much time and effort later. In compRx project we had invested about 40 % time for first two phases (requirement and architecture design) and 40% time is for coding or implementation and rest dedicated to integration and deployment. In the implementation phase we have adopted agile methodology which is 40 % of the total project time. As agile methodology describe as iterative and incremental so every aspect of implementation is continually revised. In the implementation

phases we created stub method for gui layer, business login layer, dao layer and Db module layer then test suit for stub method. Then next iteration we have implemented actual method. We did testing concurrently with coding and also test job start in the early days of the iteration. Also iteration in agile development drove us to add method, remove method and add/remove testing in implementation phase.

2.1 Requirement Phase:

The first step of water fall model is requirement analysis. In compRx project all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification section. We gathered functional requirement by interviewing client. We used "use case" elicitation technique to create a description of each requirement. Each individual came up with their own use case on the basis of client's first interview and finally we combined all and choose the best one. But the biggest problem was the requirement we gathered in the first interview was not 100 percent accurate as client did not know the exactly what the requirement is. As a result the client constantly changed requirement. In this section the programmer have no control over it. As waterfall model is not iterative we finalized our requirement and moved to next phase with insufficient requirement. Here we should have adopted a methodology where we could find the requirement issues at the early stage of the development which would enable us to take corrective measure in a limited time and budget. In my opinion we could have used iterative process. "Iterative process model begins with an initial implementation of a partial amount of the software requirements and iteratively developed the evolving versions until the full system is up and running and completed. At each iteration, requirement and design modifications are made and new methods are added according to new requirement and design" [5].

2.2 Architecture Design

The next phase of waterfall model is architecture design based on requirement. The architecture design describes the core components and the interaction of those components, but the design does not describe the structure of each component [6]. First we had created the basic design. The architecture of "compRx" project had four layers: Gui layer, Business logic layer, Dao layer and DBmodule layer. Then we had created UML class diagram for each layer to describe technical design. As testing phase comes late in waterfall model there we had inconsistency in each layer. When we tried to combine each layer into unified class diagram we

found inconsistency between all layers. For example inconsistency between business layer and Dao layer where in BL layer we defined getuser(): this method should have a parameter to get user information. Adduser(username,password): we need to have more information in the parameter to add user in the system such as: contact number, name, address , role. Initially different groups work in different layers some class diagram methods in one layer overlaps with other layers. The lack of communications between groups lead to confused the whole class diagram. Gui layer included some method which should have belong to BL or Dao layer. BL layer included some method which should have belonged to Dao Layer. Different groups had different opinion as a result inconsistency occurs when we combined different class diagram. For creating class diagram for each layer we should have assigned this task to one group instead of creating three different groups. Also another way we could have avoided this inconsistency if we would have first created unified class diagram. Another drawback of waterfall model is that architecture group sometime does not anticipate the future implementation difficulties. In compRx project we had included isCheckedOut() method and pharmaceuticlAnalysis() method for pharmacist. Later we find out isCheckedOut() method could create dead lock and pharmaceuticalAnalysis() method would be extraordinarily difficult to implement. If we would have revised the design and iterate few times we would have address the issue before implementation phase.

2.3 Implementation or coding

In this phase we began to follow agile methodology. Agile is iterative and incremental and testing of software starts from the initial to different stage during software life cycle unlike waterfall testing began in the final stage. “compRx” project’s implementation phase starts with creating test cases for each layer. We followed coding convention written by technical lead. We then created stub method for each layer and unit test for stub method. We worked as a pair of a group. We have reviewed our code. We have tested every piece of code before move to next. In compRx project as because the four layer class diagram architecture was inconsistence, initially we had to create more stub method for each layer. After next iteration when we get unified class diagram then stub method reduce drastically in every layer. For example in DBmodule layer we had initially created 32 stub methods but finally it had reduced to 22 stub methods as well as changing in unit testing. As we were adopting agile methodology we had welcomed the changing in architecture requirement and fix the issue accordingly. In my opinion we had managed to finished implementation phase just on time by delivering the product rapidly.

2.4 Integration and deployment

The last phase of waterfall methodology is deployment. Four layers needed to integrate together before deployment. After successfully tested all four layers individually we have integrated the system. Unfortunately each interface cannot communicate properly due to missing some methods in different layers. We were able to make UI layer fully functional but all the section was not working completely due to other layers missing methods or having issue with bugs. For example Prescription screen was partially working due to problem with Doa layer. Particularly persist prescription was not working due to bug in DBmodule layer. This is just one example. At the end we had to open 43 bug issues need to be resolve in order to system up and running properly. At the last week of our project these small issues causes a lot of problem. As we have followed waterfall model until we finished the final product the client did not have any idea what we have been designed or produced is exactly what he had asked for. When we presented the final product the client was not satisfied with the product as we did not capture his imagination. The drawback of waterfall method is that "it does not take into account a client's evolving needs"[10]. In my opinion we should have adopted Iterative process.

3. Conclusion:

Throughout the project we have dealt with non IT client. So waterfall model does not work well with non IT client. We should have adopted iterative model design instead of waterfall model.

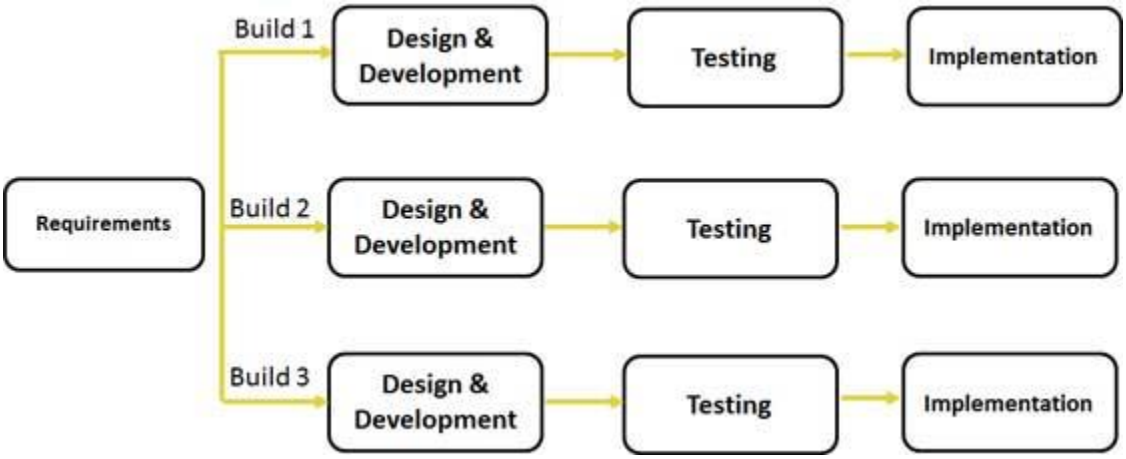

Figure iterative model [5]

The advantage of this model is within first iteration we can have a working model which is will give the developer flexibility to find bugs and design flaw. Finding issue and bugs at the early stage will save time and budget. This model also takes into account of client evolving needs [5].

References:

1. http://www.itinfo.am/eng/software-development-methodologies/
2. http://en.wikipedia.org/wiki/Waterfall_model
3. http://agilemethodology.org/
4. http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
5. http://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm
6. http://zone.ni.com/reference/en-XX/help/371361J-01/lvdevconcepts/lifecycle_models/
7. http://www.waterfall-model.com/
8. http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t
9. http://www.buzzle.com/articles/waterfall-model-advantages-and-disadvantages.html
10. http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/