



CSE471

System Analysis and Design

Assignment : 01

Submitted By

Shoaib Ahmed Dipu

Student ID : 17101482

CSE471-2, Summer 2020

Submission Date : 15 July 2020

Assignment 1

NO. 01

For Steel Mill company, in my opinion,

Agile software development methodology will be best suited as per their situation.

There are several approaches to agile development, including Extreme Programming (XP), Scrum, and Dynamic Systems Development Method (DSDM). We will mainly take Extreme programming approach under Agile methodology for Steel Mill company.

Explanations

(i) As the mill company won't be able to provide detailed requirements because

of lack of technical person, so it will be better to use Agile Methodology as much of the modeling and ~~documented~~ documentation overhead is eliminated. An Extreme programming (XP) project begins with user stories that describe what needs to do with short details. Then, programmers code in small, simple modules and test to meet those needs. So, small company won't be affected due to this lackings of theirs.

(ii) As they need to build this system in a short time, Extreme programming is quite suitable as this approach takes an 'extreme' approach to iterative

development. New versions may be built several times per day. Increments are delivered to customers in every 2 weeks. cycles are kept short ~~one to four~~ and the development team focuses on adapting to current business environment.

(iii) As the mill company will provide manpower to collaborate with development team, Agile methodology is perfectly suited for them. Here, much emphasized is put on customer satisfaction and teamwork. Communication, simplicity, feedback and courage are core values. Developers communicate with customers and fellow programmers. Early and frequent testing provides feedback and developers are able to courageously respond to changing requirements and technology.

(iv) In Extreme programming approach,

Standards are very important to minimize confusion, so, XP teams use a common set of names, descriptions and coding practices.

XP projects deliver results sooner than

even RAD approaches and they rarely get

bogged down in gathering requirements for the system. This ^{will be} quite effective for them as they lack technical skill and have short time.

(v) And, lastly, as ~~they~~ they want to provide good budget for purchasing developing tools,

Agile methodology will be quite fruitful for them as here, the development team focuses on adapting to the current business world environment.

No.02

Differences between Prototyping and Throwaway prototyping:

prototyping	Throwaway Prototyping
Prototyping takes shorter time to deliver.	That Throwaway prototyping takes longer time to deliver as prototypes do not become final version.
Performance in complex projects is quite poor.	Performance in complex projects is quite excellent.
Performance with unfamiliar technology is quite poor.	Performance with unfamiliar technology is excellent.
Users are actively involved in case of prototyping.	Users are not actively involved in case of throwaway prototyping.

NO.03

In this case, I would use 'throwaway prototyping' methodology which is one of the RAD methodologies. Reasons behind this preference of mine are provided below:

- (i) As the retail store is a large chain, it needs to store data in each store and would exchange data with a mainframe computer. The system ~~has to~~ would be complex and needs reliability. Throwaway prototyping performs excellently than any other or most other available methodologies.

(ii) As there will be too many data transmissions and transactions, the system has to be reliable. Throwaway prototyping outperforms other methodologies for this case.

(iii) As the chain is large, ~~and it is~~ we need to build the system as quickly as possible because we will have to preserve time for system installation and also needs to maintain a mainframe computer. Because of this reason, Throwaway prototyping is ~~for~~ preferable as it is ~~the~~ both reliable, stable and quicker methodology.

(iv) Throwaway prototyping balances the benefits of well-thought-out analysis and design phases with the advantages of using prototypes to refine key issues before a system is built. It may take longer time to deliver the final system with system prototyping (because the prototypes do not become final system) but the approach usually produces more stable and reliable systems.

So, due to the above mentioned reasons, I would use 'Throwaway prototyping' for this case.

NO.04

In this case, I will refuse to switch from Waterfall development methodology to prototyping, My refutations will be,

(ii) As the project is large and complex, Waterfall development is better suited for this project, ~~as~~^{In} Waterfall development methodology, it needs to have approval from the committee and project sponsor from moving one phase to another. It is quite helpful to go through such rigourousity for a complex project.

(ii) While Waterfall development methodologies have the advantages of identifying the requirements long before programming begins and limiting changes to the requirements as the project proceeds and moves from one phase to another only if the previous phase is completed, prototyping performs analysis, design and implementation phases concurrently in order to quickly develop simplified version of the proposed systems. And so, waterfall is preferable to prototyping. One major problem with prototyping is that its quick system release challenge attempts to conduct careful methodological analysis. Often

the prototype goes through such significant changes that many initial design decisions becomes poor ones. This can cause problems in the development of complex systems because fundamental issues and problems are not recognized until well into the development process.

(iii) Even though it may seem that prototypes will perform excellently with unclear ~~etc~~ user requirements, they are much less so, since the early prototypes that are built usually only scratch the surface of the new technology. In usual cases it is only after several prototypes and several months that the developers discover weakness or problems in new technology.

iv Even though, projects can be delivered much earlier ~~than~~ ~~wa~~ ~~is~~ following prototyping methodologies, but, in case of large & complex system, customers have much or enough time and they often prioritize reliability, stability for their complex systems.

Due to the above mentioned reasons, I will refuse the Request of my manager and through these reasoning, I hope I will be able to manage permissions to go on with Waterfall methodology.