**SSW 555 Agile Methods for Software Development**

**Homework 8**

**------------------------------------------------------------------------------------------------------------------------------**

**Prototyping plays a major role in DSDM. One of its main purposes is to replace documentation  describing the final product with a working version of the product. That is, DSDM emphasizes the  creation of software artifacts rather than documents in order to capture requirements and design  intentions.**

**1. Describe at least 2 risks of using prototypes rather than documents to describe  requirements and designs. Hint: not all of the actors involved in DSDM are software  developers.**

The 2 risks of using prototypes rather than documents are:

1. As only software developers know the technologies used in building the prototype, no one, other than them can make any changes in the prototype. It can become even worse if one of the developer leaves the team and others have no idea of what he used in building the prototype.

On the other hand, documentation can be easily understood by everyone. Also, anyone could clearly suggest if any changes are required. Versioning of documentation is easy than versioning of software.

1. If the design of the prototype is not as desired, it is very costly to design and build the whole prototype again from scratch. This can prove very costly, if the working project is very large.
2. Progress made in software development can be hard to judge and problems made in the prototype or missing components, if any, may be hard to find with no documentation to support the prototype. Every time the prototype is refined, it should be tested all over again to see if the earlier components are working fine and new functionality built on it is functioning as desired.

**2. For each of the risks you just identified, describe some other artifact that could be used  besides a prototype (or, in addition to a prototype) that would mitigate the risk.**

Artifacts that could be used besides/or with prototype for above risks:

1. For the prototype to be easily understood and changed easily, it makes senses to use PowerPoint presentations of the design flow of the prototype (showing screens in a flow by drawing layouts) or design a low-fidelity prototype with easy to use wireframing tools like, Balsamiq. These tools should be as simple as possible so that the prototypes can be altered by anyone. Also, for a clear understanding and agreement between the user and the developers, the prototypes should be accompanied by specification manuals.
2. It is better to just have a user requirement specification document at this early stage of the project. Changes to documentation are much more affordable than changes to entire coded prototype.
3. A specification documentation and version control system would be best to use along with the prototype. Thus, the user can get a clear idea of how the prototype got refined and what additions have been made.

**3. For each of the mitigating artifacts you just described, estimate the cost of producing the  artifact and compare it to the cost of producing the prototype.**

Costs are:

1. There are very few costs associated with creating PowerPoint presentations. Probably 1-4 developer days. Also for wireframing tools like Baslamiq are free and can be learnt by anyone very fast. Developing prototypes on those also will take like a week at most.

On the other hand, developing a fully functional prototype version will take about a month to few months.

1. Now for understanding the prototype and writing a documentation will also take very less time compared to developing a prototype.
2. Version control system like GIT or Fabricator are also easy to use and are available at very low costs. Creating documentation will take very little efforts compared to maintaining prototype.