**Week 1 Discussion 2: Software Process Modeling**

Shaun Hoadley

CST499: Capstone for Computer Software Technology

Professor Amr Elchouemi

January 13, 2022

**Software Process Modeling**

Over the years, software development has shown an increasing decline in traditional design methodologies in favor of agile design approaches. There are a number of reasons for this change in the development dichotomy. In the following sections, this paper will discuss the limitations of traditional methods and touch on how agile process methods can better address the current challenges in software development.

**Discuss the limitations of a traditional methodology.**

There are numerous limitations in traditional software development design methods. This paper will address five of the more significant limitations. The five limitations being covered here are high costs and wasted budgets, productivity impedances, costly and difficulty making changes, management bottlenecks, and transparency issues.

First, due to the nature of traditional design processes, up to sixty-five percent of the budget might be used to develop features that may not be needed or required by the client. Such high amounts of wasted budget on those features are most certainly concerning from a business standpoint. Second, software engineers will usually work on the structure or framework for the software before even considering implementing system features. This can result in a disconnect in communication with the client causing delays in the development process. Next, traditional design processes are very rigid, particularly the waterfall method, in that each step in the process must be completed before the next phase can begin. Due to that rigidity, any changes in the design require going back, sometimes to the very beginning, in order to implement a change, resulting in further delays and costs of having to rework everything. Up next, management bottlenecks are delays caused in traditional processes from the rigorous approvals required. Every time a project needs another authorization to continue is time that is wasted while the request sits there waiting for a response. Lastly, a lack of transparency is present. Despite being inundated by status reports, those reports concern working features and functionality but have little to do with business productivity.

**Describe how agile process methodologies can better address current challenges for software development**.

While agile process methodologies are not a silver bullet for everything in programming, it does have it’s own challenges; in most situations today, agile development vastly improves software development in many ways. Agile development relies on short development iterations to release working core product features first, followed by other features and fixes in future release iterations based on priority. Agile development allows for easy and fast adaptation to change due to the compartmentalizing of product components. Agile software development is far more transparent than traditional methods because of client collaboration. With agile development, testing is done as each module is completed. As more modules are completed and tested, they can be implemented and tested within the previous working release to ensure they will not break the functional release. This testing reduces risk by allowing problems to be discovered more rapidly and addressed immediately. All of the previous statements combined mean significantly higher product quality is released. Because of the short cycles of development and feature/function priorities, deliverables can be predicted more accurately (that does not exclude the chance of unforeseen delays, though it significantly reduces their chance). There is user-focused testing and overall better client satisfaction, along with the client collaboration mentioned before (meaning increased stakeholder engagement). Above all else, agile development methods allow greater project control.

**Week 1 Concept Map**

