

# **Thesis Project Portfolio**

## **How to Choose Your Agile Framework**

(Technical Report)

## **Manage Your Expectations: How has Agile Lived up to its Promise to Decentralize Power?**

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

**Jacob Rice**

Spring, 2023

Department of Computer Science

## **Contents of Portfolio**

Executive Summary

How to Choose Your Agile Framework

(Technical Report)

Manage Your Expectations: How has Agile Lived up to its Promise to Decentralize Power?

(STS Research Paper)

Prospectus

## **Executive Summary**

When building a software product, the processes by which it is built are nearly as important as the technical details, as the processes define how work is done and by whom the work is done. In the modern day, many of these development processes fall into the category of “Agile” development processes. In fact, 94% of software development companies use Agile processes to develop their software. However, it is difficult to implement Agile successfully. Many companies have tried and failed to implement Agile for a number of reasons, which range from lack of organizational support to simply choosing the wrong process for their product. Thus, understanding how Agile can be implemented effectively is of great importance to the successful delivery of software. As such, this work aims to develop a deeper understanding of how Agile can be successfully implemented for a given project as well as how the implementation of Agile can affect the benefits it provides.

Given the large number of different Agile methodologies, it can be difficult to select the appropriate one for your project. In fact, the use of inappropriate Agile methodologies is often cited for failure to implement Agile successfully. Thus, the technical portion of this Capstone aims to develop a set of guidelines for selecting an Agile framework from the three most common Agile frameworks: Scrum, Kanban, and Xtreme Programming (XP). This is done by reviewing documentation surrounding the three methodologies to develop a set of guidelines on when to use each. These synthesized guidelines emphasize the nature of the work as well as the organizational culture in which the software is developed. Kanban is useful when organizational hierarchy must be maintained and there is a need to limit concurrent tasks or monitor the work done. Scrum requires collaboration, transparency, communication, and flatter hierarchies within a team and is useful when work must be done quickly with less emphasis on code quality. XP is

useful in many organizations when Scrum might also be a good match but places a higher emphasis on code quality rather than fast delivery. Thus, each framework has its own niche which it fills. However, this set of guidelines is not exhaustive and does not consider many other Agile frameworks.

Further, this work aims to understand how the implementation of Agile can affect the benefits that Agile is supposed to provide. One such benefit is the flattening of organizational hierarchy that allows developers more power over the development process. Agile demands a flattened hierarchy, though whether this flattening occurs may depend on the type of framework being used. Kanban, for example, as mentioned previously, allows for the maintenance of organizational hierarchy, though some flattened hierarchy on individual teams is still important. It has been shown that sometimes, despite this, hierarchies may form, either explicitly or implicitly. To investigate the extent to which hierarchies flatten in Agile environments, in particular, within governmental organizations and contractors, I conduct interviews of people who have worked in an environment that is transitioning from plan-driven development to Agile development. Using an actor-network theory approach, I synthesize hierarchies based on the interviewees responses to determine the extent to which hierarchies were flattened by the introduction of Agile. From this analysis, I conclude that, in general, Agile does flatten hierarchies within these organizations, though the extent to which it does may vary. Further, I posit that the potentially limited flattening of hierarchies may be due to the inherently bureaucratic nature of many of these organizations, which necessitates a high level of organizational hierarchy.

Together, these projects help to give a more thorough understanding of how Agile can be implemented effectively by understanding how to select an appropriate framework for use on a

software development project as well as understanding the role that Agile can play in flattening hierarchies. I believe that these projects were successful in what they set out to accomplish, though they both have their own limitations. The work on developing guidelines for selecting between a set of frameworks is limited to selecting between three Agile frameworks. However, many more such frameworks exist, and warrant their own guidelines. Further, many implementations of Agile development are not one framework or the other. Rather, they combine aspects of many different frameworks to create an even more custom-tailored Agile process. Thus, future research could potentially address either the addition of more frameworks to the guidelines or creation of guidelines for the composition of elements of frameworks. The work on how Agile flattens hierarchies is also limited. Its dependence on interviews necessarily limits the amount of information that can be captured. To address this concern, future iterations of this research could instead follow an observational approach, wherein a transitional environment would be observed, and the researcher could ask questions as they arise and note interactions between participants in the system. Despite these limitations, however, this work still creates a deeper understanding of how Agile processes can be effectively implemented and how best to reap the benefits of Agile.