Greih Murray

Assignment 1.3

August 14, 2024

The history of DevOps has most recently included three primary movements or philosophies, those being the Lean Movement, the Agile Manifesto, and the Continuous Delivery Movement. Each of these methods has its unique qualities, as well as its share of similarities with the others.

The Lean Movement, first popularized in the 1980s as the Lean Manufacturing Movement, seeks to optimize efficiency and minimize waste within a software development team. The Lean Movement does this through the use of seven main principals, those being: Eliminate Waste, Build in Quality, Amplify Learning, Delay Commitment as Long as Possible, Deliver Fast, Respect People, and Optimize the Whole. Eliminate Waste is much what the name implies, if waste is detected during a development iteration, a plan is made to remove that waste so it will not happen again. The waste in question can be any of a broad range of things such as quality issues, unneeded code, having too many tasks in the task log, or even too much bureaucracy, among many others. Build in Quality is also much what the name suggests, it focuses on ensuring that not only is the product built, it is built with quality as a key component. The way to ensure quality is built in is primarily up to the team, though a couple of common methods are pair programming and test driven development. Amplify Learning focuses on the concept of sharing knowledge within the team, when any one developer learns something new, it should be shared with all other members. Delaying commitment refers specifically to delaying irreversible decisions as long as possible in order to allow for more learning and experimentation. Deliver Fast is of course focused on producing a product quickly, though it does also focus on then using feedback as a means of improvement. Respecting People is really less an aspect specific to Lean and more just a generally good idea, though the thought process for its inclusion within Lean is that respect is the basis for a productive environment through promotion of healthy conflict, proactive communication, and consistent feedback. Optimize the Whole refers to optimizing the Lean process itself more so than it refers to optimizing the product.

The Agile Manifesto, first conceived in 2001, primarily follows four main tenets, as opposed to Lean’s seven. Those four tenets are: Individuals and Interactions over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract Negotiation, and Responding to Change over Following a Plan. Agile maintains that for each of these, while the right side of the statement is important, the left side of the statement is far more important and should be the primary focus. Agile believes that Individuals and Interactions are more important than processes and tools, as Individuals and Interactions are, at the end of the day, what will decide the success of the project. Working Software is valued over Comprehensive Documentation as at the end of the day whether the software works or not is far more important than how the documentation looks, and many Agile teams actually follow the principle of the code is its own documentation, meaning the code should be written in such a way that it is easy to understand, even without documentation. Customer collaboration over contract negotiations just means that the team should value collaboration with the customer more than they value the dollars and cents they will be paid, as it will typically lead to a better product. And finally, Responding to Change over Following a Plan may be one of the most important tenets, and even a contributing factor to the name Agile. If you are Agile you are able to quickly respond to issues when and if they arise, even if it does not necessarily align with the original plan, and it goes almost hand in hand with the old quote “Everyone has a plan until they get punched in the mouth”, simply meaning that plans are well and good, but more often than not something will come along which throws the plan out the window and a team should be able to handle that.

The Continuous Delivery Movement focuses on continuous delivery as the name would suggest. This movement uses its own set of six best practices that teams should follow, those being: Make Every Change Releasable, meaning each change should include everything it needs in order to be released, Embrace Trunk Based Development, which focuses on avoiding development branches which may delay integration, and write code with the intent of merging it back into the trunk quickly, Deliver Through and Automated Pipeline, meaning a team should have a robust, automated process that can be used to deploy code quickly and consistently for both test and production environments, Automate as Much as Possible, meaning that any process which can be automated, likely should be, as it will speed up the overall development lifecycle, Aim for No Downtime, meaning the application and infrastructure should be built in such a way so as to prevent downtime during deployments and allow for deployments even during peak hours if need be, and Releasing at Granularity of the Test, which means any parts which are dependent on each other for testing, should be deployed simultaneously so as to ensure compatibility.

As can be seen, each movement has its own quirks, and they all share various similarities with at least one of the others, and many people consider the Lean Movement, and the Continuous Delivery Movement, to at some degree, be integral parts of Agile itself. The largest discrepancy of course being Agile’s tenet of People over Processes, and the Continuous Delivery Movement’s belief of automate as much as possible, though these can still be used hand in hand if done properly.