

Ming Lin

Dr. Richard Ens

SSW-555

September 07 2020

Homework 01

The objective of the client, Driverless Cars, is to develop a custom software solution that enables a self-driving feature in their cars. This system's most valuable proposition will allow cars to drive themselves without human intervention, including driving and parking while avoiding accidents with pedestrians and other vehicles. Based on the MVP, the proposed system will have mission-critical features, if the system's self-driving feature were to fail there would be a likely scenario with fatal consequences. Keeping in mind, the risks of the system's primary feature lets explore the possible implementation and management approaches - waterfall method, rational unified process, and extreme programming.

The waterfall method is a traditional, tried, and true engineering management approach focused on plan-driven development. Its benefits and drawback are based on the rigid and sequential development process. The waterfall method provides the lowest risk development approach as it focuses on the requirements and pre-development process. As well as, intense testing and quality control at each step in the process. As a result, this method actively reframes from performing steps in parallel, and a majority of time is allocated to discussion during the pre-development period. If executed correctly, the goal is to cultivate a deliverable product that has undergone thorough verification and validation without any iteration during the development process. This could be beneficial, perhaps crucial while developing a mission-critical feature such as self-driving.

However, the focus and structure of the waterfall method do come with drawbacks. This strict step-by-step process will leave little to no opportunities for flexible communication and iteration during the development process. The success of this method is based on the clear communication of requirements and the scope of the project. Failure to do so could result in

revisions during development. Based on the fundamentals of the waterfall method any requirement changes will result in a re-evaluation of the final product requirements. Relative to the magnitude of the change or iteration the timeline of the project could increase greater than it already is. It is important to point out that of the three software development life cycles being discussed, the waterfall method would likely have the longest timeline.

The extreme programming approach focuses on constant iterations based on the client's feedback and communication throughout the entire process. This can be perceived as the direct opposite of the waterfall method. This process involves minimal pre-development planning besides an agreed-upon basic abstract vision of the final product. The focus of this process would be on developing the core feature of the product and releasing a working version to market as fast as possible. During the development process, there would be a great focus on continuous integration, reusable components, and modern coding standards. In this case, this could be beneficial as the core feature of this project is self-driving. As a client, we expect to have frequent engagement with you to discuss iterations after we showcase small components of the feature throughout the process. If possible, it would be beneficial to both parties to have an on-site team member from the client's organization that understands the client's vision of the product.

The downsides of the extreme programming approach would typically be a result of the fast-paced nature. Due to a minimal planning phase, the development team would usually start writing code almost immediately. This priority on coding could often lead to the design being overlooked, which could lead to a final product that is not robust or scalable. Also, this approach could result in a lack of clear documentation due to constant iterating. Lastly, much of the stress falls on the shoulders of the developer as they are expected to constantly implement changes to the codebase relative to each iteration. In short, throughout this fast-paced process, it is the responsibility of both parties to remain diligent, so a crucial component is not missed.

The last approach, rational unified process, is an iterative approach focused on high-risk features. This approach would work well in this project because the high-risk feature is the core self-driving feature. This approach utilizes best practices focused on providing every team

member the same aggregated resource base. This allows team members to be synced up at all times on the status of the project. Another focal point of this approach is maintaining a single source of truth in the codebase by building versatile components and creating abstract models of concepts within the codebase. Although this approach does require some planning, it is very effective at allowing teams to work in parallel.

RUP is generally a more complex process relative to the waterfall method and XP because of its emphasis on maintaining a single source of truth. This focus on internal integration could lead to issues during testing due to conflicting integrations. This incredibly integrated knowledge could be easily skewed by a team member if they do not keep up to date. This emphasizes the need to maintain sources of truth on an administrative level and the codebase.

Now, when implementing each of these approaches they would require a minimum amount of human resources to be dedicated in order to operate. RUP would require at least 6-12 team members to operate 9 core workflows. They would all typically be working in parallel, however, the engineering team would begin approximately one phase after. The waterfall method would require a larger team of at least 15 people and due to the strict sequential nature of this approach team members are not interchangeable. Rather each team member would focus on their own specialized domain, but this could lead to idle team members. The extreme programming approach could operate with a smaller team of 5-12 members. Due to the complexity and fast-paced nature of this approach all members are expected to work closely together in parallel. Usually, RUP and the extreme programming approach would require a representative from the client's organization to be an onsite team member.

Ultimately, this the core feature of this project is high risk. As a result, there should be an emphasis on planning and producing a reliably safe solution.