**1.1 Project Management Methodologies**

1. What is Project Management?

Project Management is the way that a certain group or company can move towards a goal through pre-determined steps. It can be thought of as a ladder, where the best way to reach the top, or the end goal is by utilizing the other steps. There is not one ‘correct’ way to do project management, as every project will have a different budget, time limit, scope, and resources (Reh, 2018).

1. Why do we need Project Management?

Project Management is important because it ensures that the project aligns with the original aim and doesn’t veer off course. It ensures that the client is always aware of what is happening at every step, and they can see if the project that they wanted is different to what is being done. Project Management gives a sense of leadership to a team, which also adds accountability if something were to go wrong, or if something is done really well. Project Management ensures everyone involves always has a clear idea of what is happening, and therefore can work to make their part of the project work together with the project as a whole. It ensures that the project is realistic, and the quality of the project is maintained throughout the entire process. It also can mean the project can steer clear of risks that might damage the project (Aston, B. 2017). A good example of a project that failed as a result of poor Project Management, is Apple’s Lisa Computer, which was originally released in 1983. One of the causes for the failure was the fact that the project scope kept changing. As a result of this, engineers were forced to recreate the computer many times, before Apples founder, Steve Jobs, finally left the team a year before the release. Some say that Jobs leaving the team was the killing blow, because he was one of the main people holding the project management together for the project, and when he left the project spiralled out of control, until a product was released. The Apple Lisa only sold 100,000 machines, which is still considered one of Apples biggest failures to date, but it might have all been avoided if proper Project Management was carried out (Bianchini, R. 2018).

1. What are some good Methodologies for Software Project Management?
2. Briefly explain what each Methodology is about/How does it work?
3. What are some advantages/disadvantages of each?

**Waterfall**

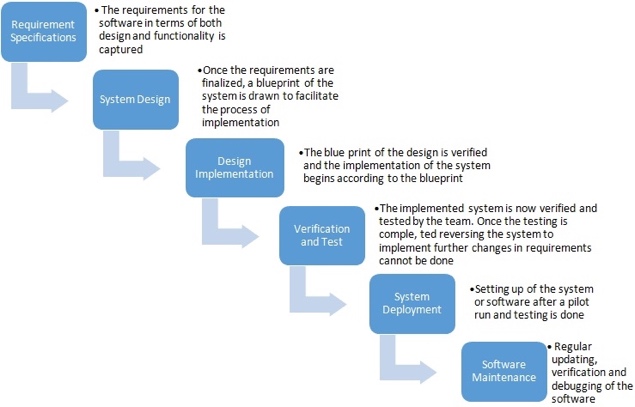
****Waterfall is one of the oldest methodologies that have been used for project management. Waterfall works as the name suggests, the workflow falls much like a waterfall, as steps are set out, usually much like the picture shown on the left, and work is done by moving down each step. To move on to the next step, you must first complete the previous one. Waterfall is good for projects that need to be completed quickly, or those that are fairly small. It also works well if the project has a fixed end goal. Because of the way that waterfall is set out, it is very simple to use, and many people that haven’t had much experience with project management will find it easy to catch on to. Because waterfalls steps are set when the plan is originally made, waterfall creates a rigid work flow, and it means once started you are immediately able to divide up the work throughout the team, and each step will be done to a higher standard. Because you can’t go back once you have done a step, it also encourages the team to document more of the development process, which in turn gives a better idea of what is going on for the client. Waterfall also comes with a higher risk, however, because if there is any error that is discovered later into the project, the entire project must be started again. It may also be a disadvantage to the client because if what gets made is not what they originally wanted, the client may need to pay more money for the entire project to get started again, as the workflow only works forwards and never looks back to reflect if this is what they intended to make (Cohen, 2017).

Image: Kukhnavets, P, 2016.

**Agile**

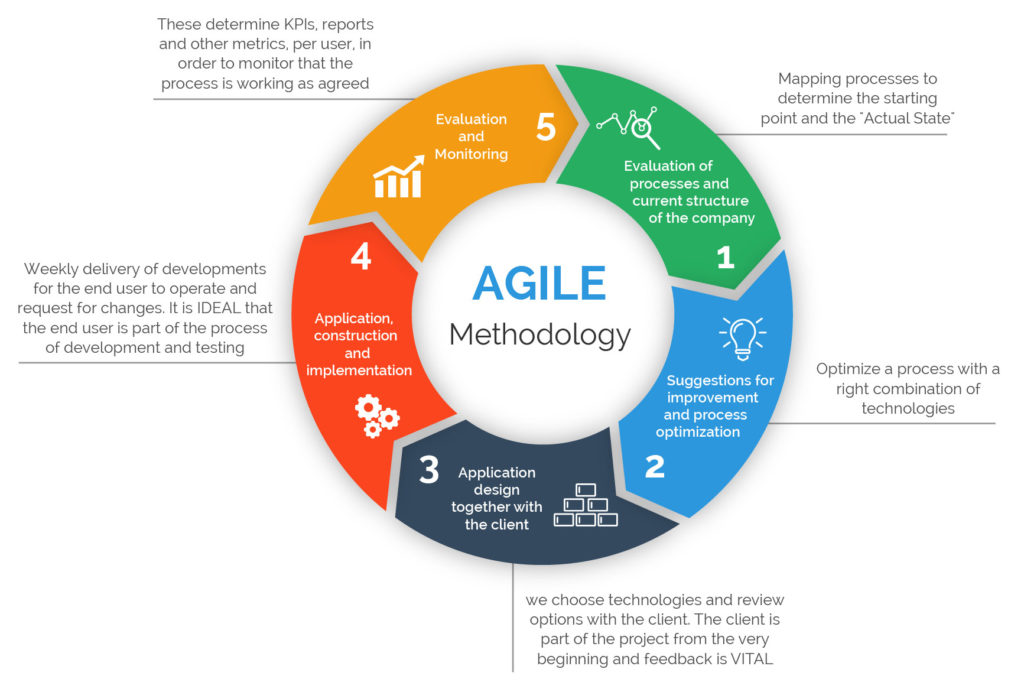
Agile is one of the most popular project management methodology for software development. Agile works by using short development cycles to prioritize improvement and development. An example of this is shown on the picture on the left. Instead of the entire plan for the project being at the beginning, like waterfall, agile gets the main idea for the plan done at the start, but fine details are modified as the project gets made. The project gets made in small blocks that are generally 2-3 weeks long, which takes the team through the process that is shown in the picture above. At the end of the 2-3 weeks, the team meets back with the client to ensure that what is being developed is in line with what the client wants, and that they are still developing what they originally intended. If it is, then the team gets to work on the next section, but if it isn’t, then they can easily redo the past 2-3 weeks, rather that the entire project. Agile is good for clients that want to be heavily involved in the development process, because they get constant updates, and can still change small areas as development continues. Agile will produce better quality products because as development is going, the developers will be constantly looking back to see if what they are doing is in line with what was originally wanted. As a result of this, agile tends to produce better customer satisfaction, and better team morale, as the product that gets produced is better. Agile creates better project control and predictability, so reduces overall risk for the project. However, agile, due to the constant reviews and checks, also takes a longer time to create the project, and therefore takes more money and resources, so isn’t really necessary for shorter or smaller scale projects (BackLinkME, 2018).

Image: BackLinkME, 2018

**Scrum**

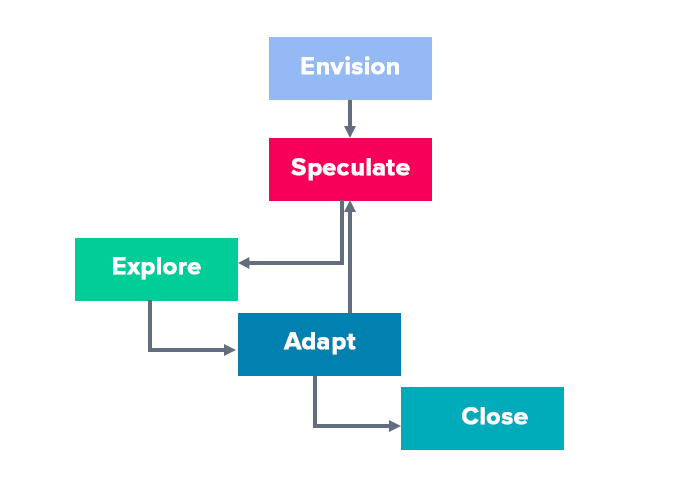
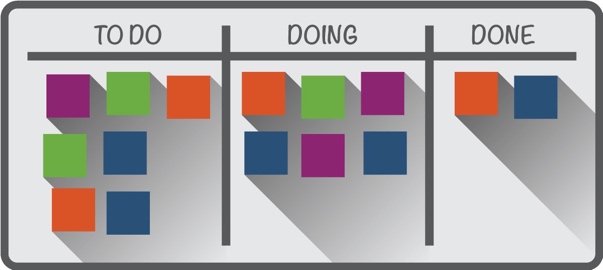
Scrum isn’t entirely a methodology, but rather a different take on agile. The biggest difference between agile and scrum is the ‘sprints’ that happen during development. Scrum works in much the same way as agile, in that development is done, and then the team reviews the plan with the client to ensure the best quality possible. There often isn’t a project manager for scrum, as generally when teams choose to use scrum, they are experienced and can usually ensure that they do what needs to be done for them. Scrum is heavily focussed on ‘sprints’, which lasts for about a month, which break down the end goal into smaller chunks with daily meetings with all in the team. This can make it easier to manage larger projects. Because of this 30-day sprint time limit, scrum heavily encourages fast development, which is good for the client. The members of the team, as a result of the daily meetings all have lots of insight into the project and can then set their own priorities as per their own knowledge and capabilities. Scrum has all of the flaws and benefits of agile, but can lead to scope creep, as there is no set finish date for the project. Scrum relies on the team to all be experienced, and wouldn’t work for large teams, but is good for large projects (Cohen, 2017).

Image: Cohen, E. 2018.

**Kanban Boards**

Kanban boards are a visual kind of project management, where all of the information is stored on a board, whether virtual or real, showing tasks organised by, for example, a to-do section, a doing section, and a done section. This is great for software project management because most of the time Kanban boards are stored online and can be accessed by everyone involved in the project, including the clients, which can then easily give a visual representation of what is happening at all times. All of the tasks or milestones that need to be done for the project are first created when planning the project, and more can be added throughout the project just by adding another board to the to-do column. When someone is doing a task, they would move the task to the doing section, which would then prevent anyone else doing this. As a result of this, Kanban boards are really good for large scale projects with lots of small tasks that need to be done. A problem with large scale projects is having to allocate which work to give each person each time work is assigned, but with Kanban boards, this is eliminated, as anyone could just go to the board and pick from one of the tasks that still need to be done. It is also good for projects that have people from all over the world, as rather than having a big meeting every week which might clash with time zones, people from other time zones can just grab whatever from the doing section without disturbing anyone else. Because any new task can be added at any time, Kanban boards are meant for projects that might change slightly over time, so isn’t great for people with a strict structure. Because it is less rigid, it can be very intimidating, and may lead to scope creep with less experienced people. Because of the time it takes to set up a Kanban board, it isn’t really practical to create one for something that might only take a week, so for smaller projects Kanban isn’t the way to go (Hubstaff, 2018).

1. Which project management methodology would you use for developing your own project and why?

I will use a mix of scrum and Kanban boards for my project, due to the ability to create a base plan to start off with, and then adding things as the program progresses. I would use scrum mainly, but due to the nature of having ‘sprints’, I would use the Kanban board for this. This is because I usually have problems with my plans where I say one thing, but then halfway through think of something else that I could add, but then means I need to redo the original plan again, but with Kanban boards I can just add things as I go along. Also, because I’m not going to be constantly working on the program, it would be helpful to see a list of things that I was working on if I get back to the project without knowing what I was doing. With Kanban boards I can also easily manage my time because I can easily see by looking at where the tasks are if I am ahead or behind of where I need to be. With scrum I can take this even further, by measuring once a week what was done, measured via the Kanban board, and whether or not it is in line with what I expected to have done by this stage, or if I had to adjust certain things for future sprints to ensure the project is completed on time, and in line with what I set out to do.

1.2 Reference List:

Question 1:

Reh, J. (2018). *Basic Project Management 101: What is Project Management?*. https://www.thebalancecareers.com/project-management-101-2275338 (Accessed 19 February 2019).

Question 2:

Aston, B. (2017). *Why is Project Management Important?*. <https://thedigitalprojectmanager.com/why-is-project-management-important/> (Accessed 19 February 2019).

Bianchini, R. (2018). *The different fate of Apple’s Lisa and Macintosh (and why design matters)*. https://www.inexhibit.com/case-studies/different-fate-apples-lisa-macintosh-design-matters/ (Accessed 19 February 2019).

Question 3:

Cohen, E. (2017). *The Definitive Guide to Project Management Methodologies*. <https://www.workamajig.com/blog/project-management-methodologies> (Accessed 20 February 2019).

Image: Kukhnavets, P. (2016). *What we should know about Traditional Waterfall Methodology*. <https://blog.ganttpro.com/en/waterfall-project-management-methodology-pros-and-cons/> (Downloaded 20 February 2019).

BackLinkME. (2018). *Agile Project Management Methodology – Manifesto, Frameworks and Process*. <https://backlinkme.net/project-management/agile-project-management-methodology/> (Accessed 20 February 2019).

Hubstaff (2018). *Kanban Project Management: Everything you need to know.* <https://blog.hubstaff.com/kanban-project-management/> (Accessed 20 February 2019).