Software Development Unit 33, 1.3

Anna Legaspi

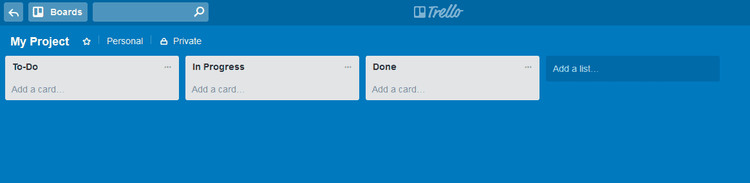
# I can understand a range of tools and techniques and relate these to projects.

There are a variety of tools available that support project management, including software development projects. These include using the SCRUM method under Agile, using Kanban to monitor task status, and also other software tools that are used to create content for the project.

## Trello for Kanban

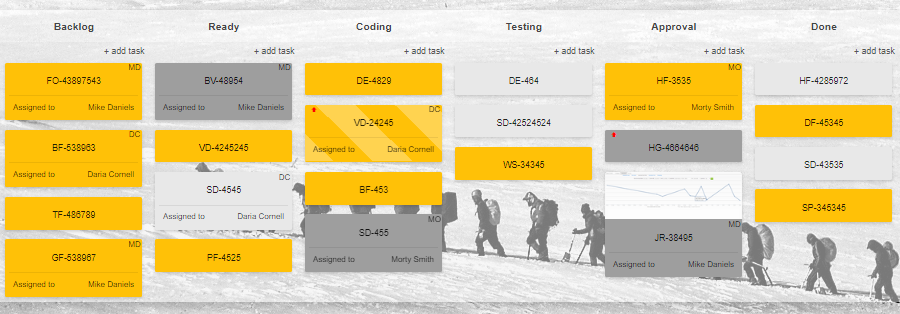
Kanban is a technique of keeping track of work that needs to be completed and was developed in Japan for Toyota. The idea is to have an open board for project members to see. It is normally broken down into three basic categories:

* To do – This will have
* WIP (Work in Progress)
* Done



Small jobs are written on sticky post-it notes and are initially attached on the ‘To-do’ section of the board. Once the work has been started, the Sticky note is then moved to ‘In Progress’. When the job is completed, it is moved to ‘Done’. The breakdown of these sections is not strict, and it can change depending on the project. From a software development point of view, more sections can be added as needed, such as below. It’s important to note that Kanban’s categories are flexible, depending on the project.

* Backlog
* Ready
* Coding
* Testing
* Approval
* Done



The benefit of using Kanban/Trello, is the visibility of where the project is at for everyone to see. It then becomes easy to prioritise what needs to be done as the team can see in one look what tasks are outstanding. In addition to knowing the status of each task, some projects might apply a rule where no more than a certain number of WIP should be outstanding at any given time. This means that manpower is focused on completing tasks if possible, without leaving too many un-done tasks, which is also aligned with one of Agile’s 12 principles.

## SCRUM Technique

Scrum is a methodology that follows the principles and values of Agile as I’ve explained in previous sections of Unit 33.

In Scrum, the following members are normally involved. There should be continuous collaboration and communication amongst these members during the whole project.

* Product owner
* SCRUM master
* Testers and Developers

SCRUM normally works by running several **sprints** until the project is completed. Each sprint lasts approximately 3 weeks and is composed of the following:

* Planning – Everyone discusses the user stories to estimate the work required
* Building – Developing the product
* Testing – Testing the developed product
* Reviewing – Review the results, bugs and further improvements

The outcome of the first sprint is then discussed along with further improvements in the next sprint, and so on and so forth. The goal is to have a viable working product with each sprint tackling bugs and improvements until the project is completed. During this whole time, the product owner is collaborating with the team so any feedback received from the product owner is also implemented in the product. The main goal in SCRUM is to deliver a potential working product at the end of each sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| **SCRUM** | | | |
| **Sprint 1** | **Sprint 2** | **Sprint 3** | **Sprint 4** |
| Plan | Plan | Plan | Plan |
| Build | Build | Build | Build |
| Test | Test | Test | Test |
| Review | Review | Review | Review |

## Knowledge of HTML, CSS and other programming languages

To enable developers to create the software, they need to have knowledge of coding or writing software. This can be as basic as HTML and CSS, or it even be more complex programming languages like Python, C#, etc. The developers would need to make sure that they have knowledge and skills on the relevant programming language to be successful.

## Other tools

In addition to principles, techniques and methodologies, other tools are also used in the project to help create the software such as integrated development environments and hardware.

* Graphics – someone in team must know how to create or the graphics to create the interface of the software. This can be created with the use of GIMP, Photoshop, MS Paint, etc. The correct or most appropriate filetype should be used if it means it will improve the performance of the software. i.e. JPG, SVG, etc.
* Browsers – If the software is going to be available online such as a website, then it will need to be tested in several browsers. i.e. Google Chrome, MS Edge, Apple Safari, etc
* Sounds – If the software needs to have audio, the audio files (mp3, ogg, etc) needs be created in the right software too. i.e. Audacity, Abelton, etc.
* Video – If videos need to be created, this can be created in MS Video editor, Filmora, etc.
* Hardware – the hardware used by the developers and testers must also adhere to the minimum requirements of the development environment and any integrated software within it.