Week 1 Deliverables

**IMPORTANT LINKS**

Here is a link to BitBucket:

[Bitbucket | Git solution for teams using Jira](https://bitbucket.org/)

Tutorial on how to use BitBucket:

[Learn Git with Bitbucket Cloud | Atlassian Git Tutorial](https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud)

BitBucket Command Page:

[Preparing for an SVN to Git Migration | Atlassian Git Tutorial](https://www.atlassian.com/git/tutorials/svn-to-git-prepping-your-team-migration)

Git Cheat Sheet:

<https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

Scrum Link: <https://www.atlassian.com/agile/scrum>

Agile Link: <https://www.atlassian.com/agile>

Agile Manifesto: <https://www.atlassian.com/agile/manifesto>

Kanban Link: [Kanban - A brief introduction | Atlassian](https://www.atlassian.com/agile/kanban)

# BitBucket (Resource) – Version Control

Git is a free and open-source version control system created in 2005. Before that, it was SVN and CVS. Git is a distributed open-source version control system, as everyone has their own individual copy of their repository locally, making operations such as commit, blame, diff, merge and log faster.

A modern software for development, Git’s pull requests are useful allowing for collaboration so people can see eachothers code. It has an excellent system for merging and rewriting local repository history.

How Git Works:

1. Create a repository with a Git Hosting Tool (BitBucket)
2. Copy/Clone the Repository to your local system
3. Add the file to your local Repo
4. Push your file to main branch
5. Make a change to you file using the Git hosting tool and commit
6. Pull these changes into your local machine
7. Create a branch (version), make a change and commit this change
8. Open a “pull” request (proposing changes to the main branch)
9. “Merge” your branch to the main branch

Learning Git with BitBucket Cloud, using a space-themed tutorial, learning the topes of Git completing the tutorial and tracking down all the team’s space stations.

* Git clone, git config, git add, git status, git commit, git push, git pull, git branch, git checkout, git merge are Key Functions

# Scrum

Scrum is an Agile Framework Methodology, which helps a team self-improve and work on their losses through continuous reviewal. Like a Rugby Scrum, Scrum Methodology reflects on its wins and losses to continuously improve, while experiencing and self-organizing to improve. Scrum describes a set of meetings, tools, and roles.

**What is Scrum + Why is it important?**

Deliver, Develop and Sustain products, Agile is a set of principles whereas Scrum is a framework for getting work done (Sprint) frameworks. Practice bringing Agile Principles into your work. Software Development + Engineering. Multiple Sprints as part of a Scrum framework.

**What are Sprints?**

Sprints are a short, time-boxed period when a scrum team works to complete a set amount of work, before it is reviews. A product is built in a series of iterations called Sprints, breaking down big, complex projects into bite-sized pieces.

**Agile Methodology**

Agile is a way of thinking, and Scrum is a framework for getting that implemented. The Agile Team are like individuals, they take some time to grow. It focuses on smaller, fixed-length iterations, and once the time-period for a sprint is finalised, the stories or product backlog entries can be implemented during the sprint cycle. Incremental improvement through small and frequent releases – cannot go Agile because it takes dedication.

The Agile Manifesto contains 4 values:

* Individuals and Interactions over processes and tools.
* Using Comprehensive Documentation to produce Working Software.
* Customer Collaboration over a Contract Negotiation.
* Responding to change over following a plan.

Definition of Scrum is based on empiricism and lean thinking; saying knowledge comes from experience and that decisions are made based on what is observed. Scrum Framework is Heuristic; based on continuous learning and adjustment to fluctuating factors.

A diagram of a scrum cycle

Description automatically generated

Scrum Framework: 4 parts: Sprint Retro, Sprint Planning, Sprint Review and Daily Scrum. These 4 are all used within the Sprint Cycle to help the product develop.

**Members of a Scrum Team:**

**Product Owner-** Champions for their product – Bring the product vision to life and they understand business and the markets current niches to prioritise the work to be done by the development team.

**Scrum Master-** Coach businesses, product owners and developers on the scrum process to finetune their practice.

**Development Team :** Testers, Designers, UX Specialists, Ops Engineers, Developers – Do the day-in and day-out work to conclude how much work should be done in each Sprint.

Sprint Plan – Daily Scrum – Sprint Review – Sprint Retrospectives

# Kanban

**Kanban** is another popular framework used to implement Agile and DevOps software development, incorporating real-time communication of capacity, and a full transparency of work. Work items are represented on a **kanban board.**

Improve Work-Management, Kanban helps you visualise your work, making it less in tangible.

Kanban – Japanese for Visual Signal 🡪 main purpose is to maximise flow.

Helps keep everyone on the same page via a **Kanban Board.**

Build a **Kanban Board**, Fill with **Kanban Cards**, **WIP Limit (Work in Progress).** Kanban starts with what you are doing now; it is a methodology used to make Kanban work for you.

Trello + Jira is used on the Kanban Principle, to regulate workflow.

Bandwidth – aim to move kanban cards from left to right. One Kanban card is used as one work item – place them in the appropriate stage for the workflow. These cards need to be small enough in terms of their time-frame, so one card can be moved to the other card very quickly.

Focus on Kanban Lead-Time needs to be controlled by the customer, and is used to help set deadlines for project, corresponding to Kanban Card Task Size and how many kanban cards are made for each project.

**Kanban** is a popular software development methodology and workflow management used by software and DevOps teams who want to work in Agile. It has several advantages including:

* Planning Flexibility – A team completes a work item, then pluck the next work item off the top of the backlog; **Product Owner** is free to reprioritise work in the backlog without disrupting the team, as any changes outside work items don’t impact the team.
* Shortened Time Cycles – Cycle Time is a key metric for time it takes of work to travel through the team’s workflow – from the moment work starts to the moment it ships, allowing the team to confidently forecast the delivery of future work.
  + Overlapping Skills reduces the number of bottlenecks and shortens the time cycle. If there is only one person with a specific skillset, if he/she gets stuck then some other team members can take on heterogenous work, optimising cycle time. The point is for the whole team to complete the cycle and the whole team to address a single bottleneck.
* Fewer Bottlenecks – Multi-tasking kills efficiency – the more work items in flight at any given time; the more context switching, hindering their path to completion. This is why a key component of Kanban is to reduce the WIP by highlighting bottlenecks and backups in the team’s process due to lack of focus, people/ skill sets.

WorkFlow Progress:

* To Do, In Progress, Code Review, and Done : An example of a methodology which can be used.

# Comparison and Analysis of Agile, Scrum and Kanban

**Agile:** A structured and iterative approach to project management and product development; there is a recognization of the volatility in product development, for self-organising teams to respond to change. Agile is not a competitive advantage anymore; everyone uses it.

**Kanban:** Apps like Trello and Jira use the Kanban Project Management Framework to maximise efficiency/ flow. Kanban focuses on reducing the time a project takes from start to finish, using a Kanban Board.

**Scrum:** Scrum Teams commit to completing an increment of work; which is potentially shippable through a series of set intervals called Sprints; these Sprints main aims are to learn loops quickly gathering and integrating customer feedback. These scrum teams offer specific roles, create special artifacts, and also hold regular ceremonies to keep things moving forward.

A screenshot of a computer screen

Description automatically generated

**Scrum** – There are scrum ceremonies in between sprints with sprint planning, daily scrum, sprint review and sprint retrospective (reflecting on the past to improve the future) meetings. Scrum Master ensures everyone is grounded by the sprints in the scrum framework, the Product Owner advocates for the customer, managing a product backlog, helping prioritise the work done by the development team. Scrum Metrics are a data points scrum teams can use to improve efficiency and effectiveness; informing decision making and helping teams become more efficient in planning and execution. **Miro, Zoho Sprints**

**Kanban** – Backlog, to Prioritised, to Outlines Ready, to Writing, to Designing, Technical Review, and Shipped. This is more in terms of a collective responsibility of the entire team to collaborate and deliver tasks on board. **Trello, Jira**

# Agile Quiz

A screenshot of a quiz

Description automatically generated

# Model Evaluation Metrics

* Precision-Recall Curves (PR)
* KS-Statistic (Kolomagarov – Smirnov Statistic)
* Precision, Recall, Accuracy, F1 Statistic (F2 also allowed potentially)
* ROC-AUC Curves
* Lift or Gain