|  |
| --- |
| **Text  Description automatically generated** |

**ASSIGNMENT COVER SHEET**

|  |  |
| --- | --- |
| Student Name & Id: | **Sanduni Dissanayake - 24009683** |
| Student Name & Id: | **Dinesh Dapana Durage - 24007386** |
| Student Name & Id: | **Hewa Iddagodage Viran Pravinda - 24007360** |
| Unit Name: | **Managing Software Development Projects** |
| Unit Code: | **PROG6001** |
| Tutor’s name: | **Mr. Tenzin Norbu** |
| Assignment No.: | **Assessment 2** |
| Assignment Title: | **GitHub and Reports** |
| Due date: | **27/11/2023** |
| Date submitted: | **27/11/2023** |

Declaration:

*I have read and understand the Rules Relating to Awards (*[*Rule 3 Section 18 – Academic Misconduct Including Plagiarism*](http://policies.scu.edu.au/view.current.php?id=00140#s18)*) as contained in the SCU Policy Library. I know the penalties that apply for plagiarism and agree to be bound by these rules. The work I am submitting electronically is entirely my own work.*

|  |  |
| --- | --- |
| Signed: | Sanduni Dissanayake |
| Signed: | Dinesh Dapana Durage |
| Signed: | Hewa Iddagodage Viran Pravinda |
| Date: | 27/11/2023 |

Table of Contents

[Chapter 1 - Collaborative Project Using Git 1](#_Toc151888176)

[1. Project Brief 2](#_Toc151888177)

[2. Team member 1 (Viran Pravinda) tasks 4](#_Toc151888178)

[**2.1.** **Task 1: Add project brief section to chapter 1** 4](#_Toc151888179)

[**2.2.** **Task 2: Add quick facts and organisation background to RFP** 5](#_Toc151888180)

[**2.3.** **Task 3: Add contact info and RFP process schedule to RFP** 6](#_Toc151888181)

[3. Team member 2 (Sanduni Udulitha) tasks 7](#_Toc151888182)

[**3.1.** **Task 1: Fork repository** 7](#_Toc151888183)

[**3.2.** **Task 2: Add agile mindset overview to chapter 3** 7](#_Toc151888184)

[4. Team member 3 (Dinesh Madumal) tasks 9](#_Toc151888185)

[**4.1.** **Task 1: Fork repository** 9](#_Toc151888186)

[Chapter 2 – Request for Proposal (RFP) 10](#_Toc151888187)

[1. Request for Proposal (RFP): Integrated Business System for Aussie Business Buzz (ABB) 11](#_Toc151888188)

[Chapter 3 - Software Development Methods, Processes and Techniques 12](#_Toc151888189)

[1. Agile Mindset 13](#_Toc151888190)

[References 16](#_Toc151888191)

# **Chapter 1 - Collaborative Project Using Git**

## **Project Brief**

This GitHub project was created to collaborate towards the assignment document of the Managing Software Development Projects unit (PROG6001). After adding the main files, the initial repository looks like the one below (see Figure 1).

A screenshot of a computer

Description automatically generated

**Figure 1 - GitHub Repository**

There are two files included in the repository, mainly as follows.

* **Perth4\_PROG6001\_02.docx**: this is the project's main file, and all the team members will collaborate in this document.
* **README.md**: this is the readme file of the project, and this file contains unit details, team members’ info, and a description of the GitHub project.

Since this is a group project, three team members are involved in this project to collaborate towards the final document of the assignment (see Figure 2).

A screenshot of a computer

Description automatically generated

**Figure 2 - Project Collaborators**

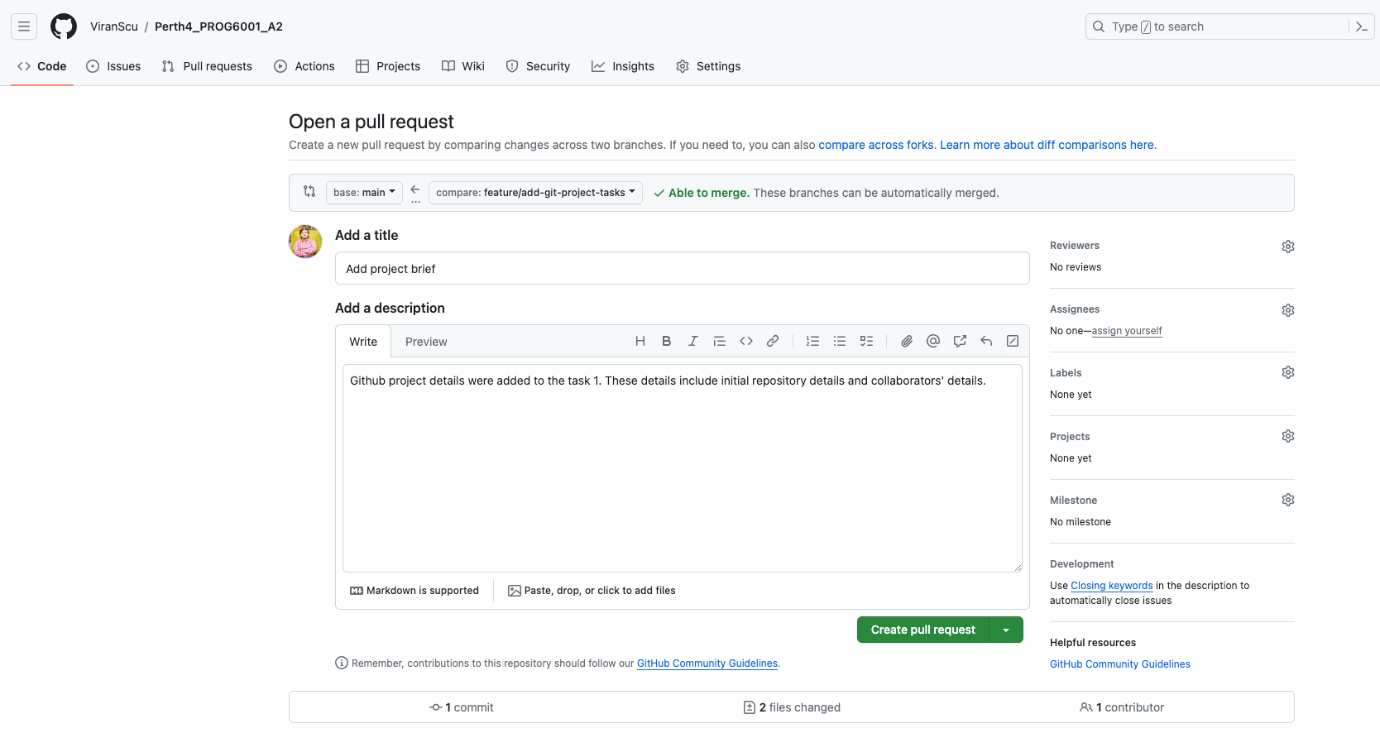
**ViranScu** is the person who created the repository, while **Dinesh** and **Udulitha** are the contributors to this project.

## **Team member 1 (Viran Pravinda) tasks**

### **Task 1: Add project brief section to chapter 1**

This task explains the files in the GitHub repository and the project collaborators. Refer to Figure 3 and Figure 4 for the pull request and final task merge.

A screenshot of a web page

Description automatically generated

**Figure 3 – Pull request for adding project brief task.**

**Figure 4 – Final merge for adding project brief task.**

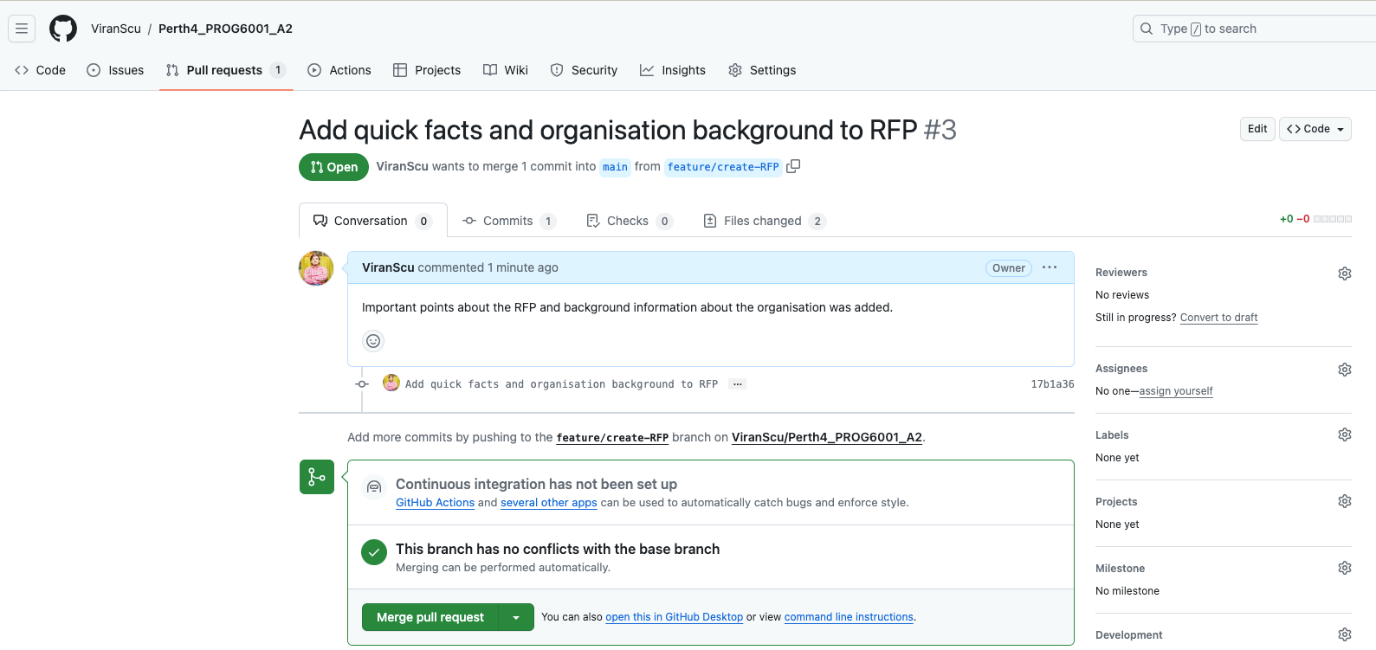
### **Task 2: Add quick facts and organisation background to RFP**

A screenshot of a computer

Description automatically generatedThis task involves adding details about the quick facts and organisation background to the Request for Proposal (RFP). See Figure 5 and Figure 6 for the pull request and the final merge of the task.

**Figure 5 - Pull request for adding quick facts and organisation background to RFP.**

**Figure 6 - Final merge for adding quick facts and organisation background to RFP task.**



### **Task 3: Add contact info and RFP process schedule to RFP**

This task involves adding contact info for the point of contact for vendors and scheduling for the proposal process. Refer to Figure 7 and Figure 8 for the pull request and final task merge.

A screenshot of a computer

Description automatically generated

**Figure 7 – Pull request for adding contact info and RFP process schedule task.**

A screenshot of a computer

Description automatically generated

**Figure 8 - Final merge for adding contact info and RFP process schedule task.**

## **Team member 2 (Sanduni Udulitha) tasks**

### **Task 1: Fork repository**

A screenshot of a computer

Description automatically generatedUse the GitHub Fork feature to create a personal copy of the main project created by another user (ViranScu). See Figure 9 on creating a new fork.

**Figure 9 - Create fork**

### **Task 2: Add agile mindset overview to chapter 3**

This task involves discussing the agile mindset overview for the “Software development methods, processes, and techniques” chapter. See Figure 10 and Figure 11 for the pull request and the final merge of the task.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Figure 10 – Final merge for add agile mindset overview task.**

**Figure 11 – Pull request for add agile mindset overview task.**

## **Team member 3 (Dinesh Madumal) tasks**

### **Task 1: Fork repository**

This task involves creating a personal copy of the main project created by another user (ViranScu). See Figure 12 for the screenshot of creating a new fork.

A screenshot of a computer

Description automatically generated

**Figure 12 - Create fork**

# **Chapter 2 – Request for Proposal (RFP)**

## **Request for Proposal (RFP): Integrated Business System for Aussie Business Buzz (ABB)**

|  |  |
| --- | --- |
| **Quick facts** | Please take note of the below information.   * **Project type**: Integrated Business Management System (IBMS) * **Level of confidentiality**: Confidential * **Response deadline**: 1-Dec-2023 5.00 pm (AWST) |
| **Organisation background** | Aussie Business Buzz (ABB) was established in 2002 in Perth, Western Australia. We have been pioneers in technology products for more than two decades now. The company sells various technology products, including PCs, laptops, phones, routers, etc. In addition to product sales, ABB offers device repairs and mobile device accessories. We have four branch shops and plan to expand our business to many other locations in Australia. |
| **Contact information** | Please get in touch with the below person for any questions.   * **Contact person**: Matt Davids * **Title:** Chief Executive Officer * **Phone**: (+61) 4 34587569 * **Email**: [matt.davids@abb.com.au](mailto:matt.davids@abb.com.au) * **Address**: 570 ABC Street, Perth, WA 6000 |
| **Schedule for the RFP process** | Please refer to the RFP schedule below.   * November 27, 2023: RFP announced/sent * December 1, 2023: Deadline to RSVP participation * December 4 – 8, 2023: Conference calls with vendors to discuss RFP * December 22, 2023: Deadline for vendors to submit proposals * January 8, 2024: Selection of finalists * January 15 - 19, 2024: Vendor finalist presentations * January 24, 2024: Final vendor selected * February 5, 2024: Targeted project start date |
| **Criteria for Decision-Making** | Aussie Business Buzz (ABB) staff will review submitted proposals and select vendors based on the following criteria.   * **Technical Suitability**: How well the proposed solution aligns with the specified requirements. * **Cost Proposal**: Clear breakdown of costs, including development, maintenance, and ongoing support * **Experience and Reputation**: History of successfully delivering similar solutions and references. * **Scalability**: Capability of the solution to accommodate future growth and evolving needs * **Support and Maintenance**: Proposed plan for continuous support and maintenance * **Innovation**: Creative ideas or features that contribute value beyond the outlined requirements |
| **Basic Summary of the Project/Initiative** |  |

# **Chapter 3 -** **Software Development Methods, Processes and Techniques**

## **Agile Mindset**

**Overview of Agile Mindset in Software Development**

Agile software development transcends mere frameworks and practices, evolving into a comprehensive mindset that emphasizes collaboration, adaptability, and continuous improvement. Unlike traditional methodologies, such as Waterfall, Agile is not confined to a specific set of rules; rather, it is an umbrella term encompassing various frameworks and practices aligned with the values and principles articulated in the Manifesto for Agile Software Development. <https://www.atlassian.com/agile>

**Agile Values and Principles:**

The foundation of Agile is laid out in the Manifesto for Agile Software Development and its accompanying 12 Principles. These values and principles guide how teams approach software development, change, and uncertainty. <https://www.atlassian.com/agile/manifesto>



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

**People-Centric Approach:**

Agile is special because it really cares about the people who work together to create something. It believes in finding solutions by having teams that organize themselves and have different skills working together. It's not just about making a product; it's about how the team works and cooperates on the way to making something great.

**Agile as a Mindset:** [**https://www.agilealliance.org/agile101/**](https://www.agilealliance.org/agile101/)

Agile is not just a set of practices; it's a mindset informed by values and principles. When faced with uncertainty, Agile encourages trying something, obtaining feedback, and adjusting accordingly. The Agile mindset guides the choice of frameworks, practices, and techniques, emphasizing the importance of context in decision-making.



Figure : Agile mindset (https://mylifenotesweb.wordpress.com/2019/04/18/breaking-an-agile-mindset-down/)

**The Agile Mindset in Practice:** [**https://www.atlassian.com/agile/advantage/agile-mindset**](https://www.atlassian.com/agile/advantage/agile-mindset)

The agile mindset isn't something you pick up once and you're done. It's more like a continuous way of thinking. It involves understanding, collaborating, learning, and staying flexible to achieve high-performing results. When teams really get why they're doing things the agile way, the way they do things falls into place and suits what the team needs.

**Four Pillars of the Agile Mindset:** [**https://www.atlassian.com/agile/advantage/agile-mindset**](https://www.atlassian.com/agile/advantage/agile-mindset)

Respect for All Team Members:

* Emphasizes the essential role of every team member.
* Fosters a culture of respect and psychological safety.
* Encourages open collaboration and contribution.

Optimized and Sustainable Flow:

* Focuses on incremental delivery, reduced batch sizes, and continuous improvement.
* Maximizes value and minimizes waste.
* Enables efficient responses to defects and avoids major delays.

Encourage Team Innovation:

* Drives innovation through collaborative feedback, new ideas, and experimentation.
* Provides freedom for team members to find improved solutions.
* Values creativity and different approaches.

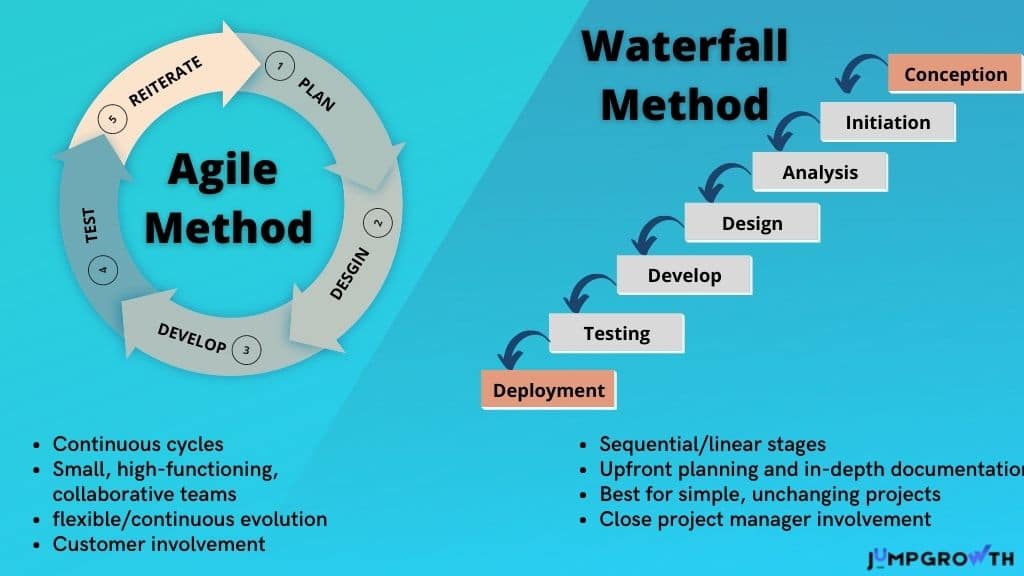
Focus on Relentless Improvement:

* Cultivates a continuous improvement mindset.
* Utilizes retrospectives for reflection and refinement.
* Creates a balance between structure and innovation.

The Agile mindset is like changing the way we think about making software. It's not just about following certain rules; it's more about working together, being flexible, and always trying to get better. To unleash the full power of Agile methodologies, teams must embrace the Agile mindset, cultivating respect, optimizing workflow, encouraging innovation, and maintaining a relentless focus on improvement. In doing so, teams can navigate the dynamic landscape of software development with agility and deliver high-value, customer-centric solutions.

## **2.2 Waterfall Methodology**

The Waterfall Model in project management operates sequentially on fixed dates, requirements, and predetermined outcomes. It follows a linear path where teams can work on their own without always having team discussions. This approach emphasizes completing one phase before moving on to the next. The model suits projects with stable and well-defined requirements but may face challenges with adaptability and frequent feedback. <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>



**how Waterfall works**

Waterfall consists of five stages – requirements, design, implementation, verification, and maintenance – progressing in a fixed order. Unlike more flexible methods like Agile, Waterfall doesn't allow jumping between phases; each step must be finished before starting the next phase. <https://www.atlassian.com/agile/project-management/waterfall-methodology>

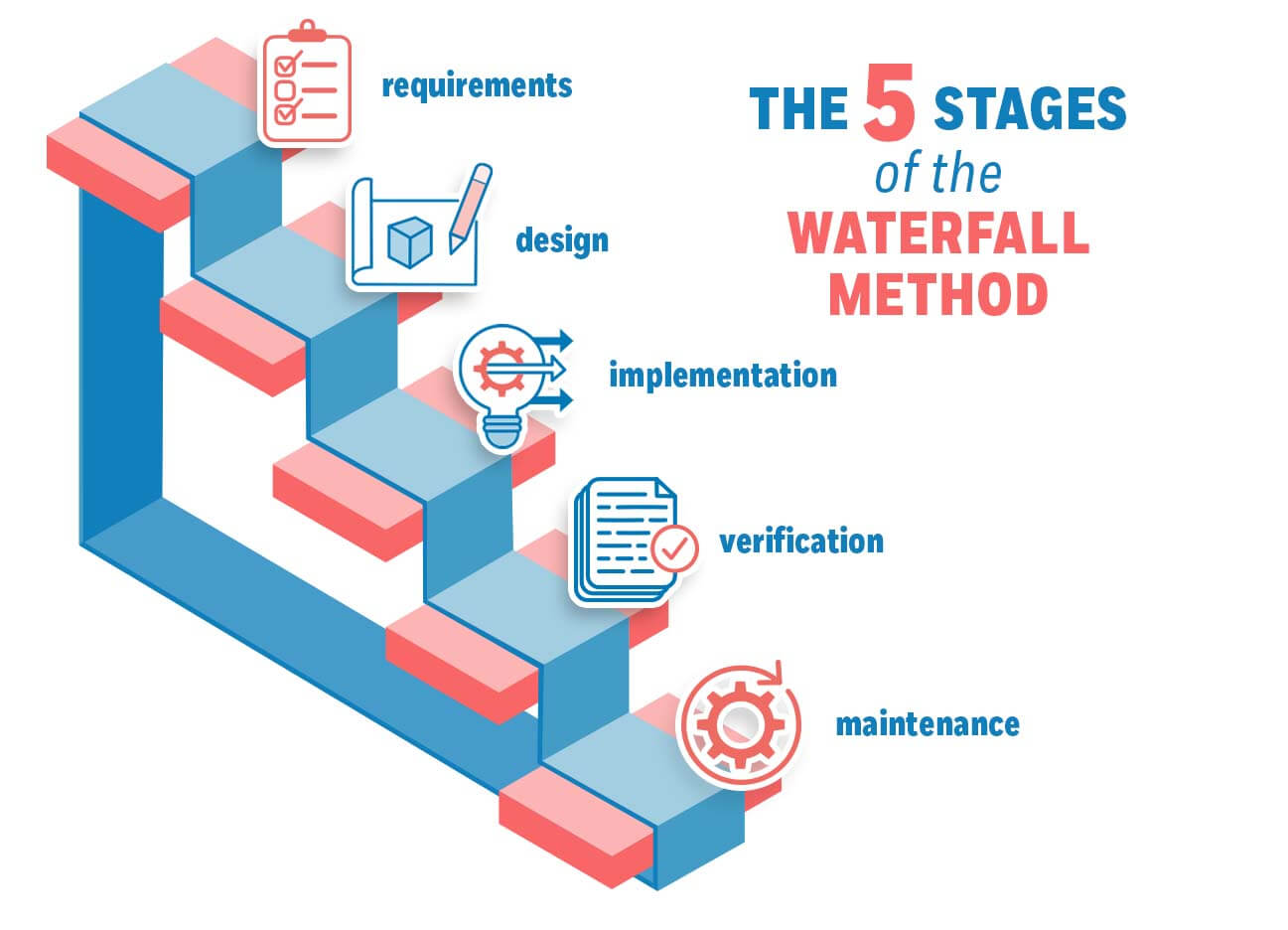


Figure 5: Stages of Waterfall Methodology (https://management.org/waterfall-methodology)

**Stages of the waterfall methodology:**

<https://www.atlassian.com/agile/project-management/waterfall-methodology>

Requirements:

* Describes what the system should do.
* Outlines the project's scope, team roles, timeline, and process details.
* Like planning the whole project from a bird's-eye view.

Design:

* Creates schedules and project plans based on requirements.
* Specifies exact details for the project’s deliverables.
* Involves designing blueprints for software or product specifications for the software project/product.

Implementation:

* Developers build the project/product based on the approved design.
* Involves creating a plan, gathering necessary data, and assigning tasks to the team.
* If there are issues, the project might need to go back to the design phase.

Verification:

* Quality assurance phase to ensure the product works well for users.
* Testing is done for all possible scenarios to avoid releasing a buggy product.
* Bugs and errors are documented and fixed.

Maintenance:

* After production release, any reported issues are addressed by the team.
* New versions of the product may be released based on customer feedback.

**Benefits of following waterfall methodology:**

* **Clear Project Structure:** Everyone understands the final goal and knows what tasks to complete and when.
* **Static Project Scope:** Costs and timelines are determined early because the project scope remains relatively stable.
* **Easier Tracking:** Progress is visible quickly with simpler tasks, and tools like Gantt charts help in monitoring.
* **Comprehensive Project Documentation:** Waterfall provides a blueprint and a historical record, offering a complete overview of the project.
* **Improved Risk Management:** Early planning reduces risks, such as fixing design problems before coding begins.
* **Enhanced Responsibility and Accountability**: Each phase has clear goals and timelines, promoting team accountability.
* **Fewer Delays Due to Additional Requirements:** Completing the design early reduces extra requests, preventing delays and keeps the cost of changes low.
* **Easy Adaptation to Key Member Changes:** Detailed documentation describes required skills, making it easy to transition when key team members leave. Even less-experienced team members can easily join in with Waterfall.
* <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>
* <https://www.atlassian.com/agile/project-management/waterfall-methodology>

**Drawbacks of Waterfall Methodology:**

* **Longer Delivery Times:** The rigid, step-by-step nature of the Waterfall approach can extend the overall project duration. Since each phase must be completed before moving on, any delay in one phase affects the entire timeline.
* **Difficulty in Time Estimation:** Estimating the project's total duration is challenging due to various factors such as organizational processes, project-specific complexities, and unforeseen delays. This uncertainty can make planning and scheduling difficult.
* **Difficulty in Welcoming New Requirements:** Adapting to new requirements in a dynamic environment is challenging within the Waterfall Model. Any change in project requirements often necessitates revisiting the initial stages of requirements and design, causing potential disruptions.
* **Limited Client Involvement:** Once the requirement phase concludes, client involvement and feedback become limited. This reduces the chances of incorporating valuable client input into the project during later stages, hindering the ability to make necessary improvements.
* **Lot of Change Requests:** Clients may have numerous change requests after the project's execution. This influx of feature requests can pose challenges in terms of maintenance and may cause delays in the final product launch.
* **Deadline Creep:** Issues encountered in one phase of the project can bring the entire process to a halt. Addressing these problems may require revisiting previous phases, leading to delays and potential disruptions to the project timeline.
* <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>
* <https://www.atlassian.com/agile/project-management/waterfall-methodology>

## **2.3 Scrum Methodology**

**Scrum Framework Overview:**

Scrum serves as a collaborative and agile framework designed to achieve incremental progress in completing tasks. The key focus lies in promoting continuous experimentation and feedback loops within the work process. By encouraging teams to iteratively experiment with their methods and gather feedback at each stage, Scrum facilitates a dynamic and adaptive approach to achieving project goals. <https://www.atlassian.com/agile/scrum>

Scrum consists of a Scrum Team, comprising a Product Owner, a Scrum Master, and Developers, each with specific roles.

The framework involves scrum ceremonies and artefacts that guide the scrum team through work. <https://www.atlassian.com/agile/scrum>

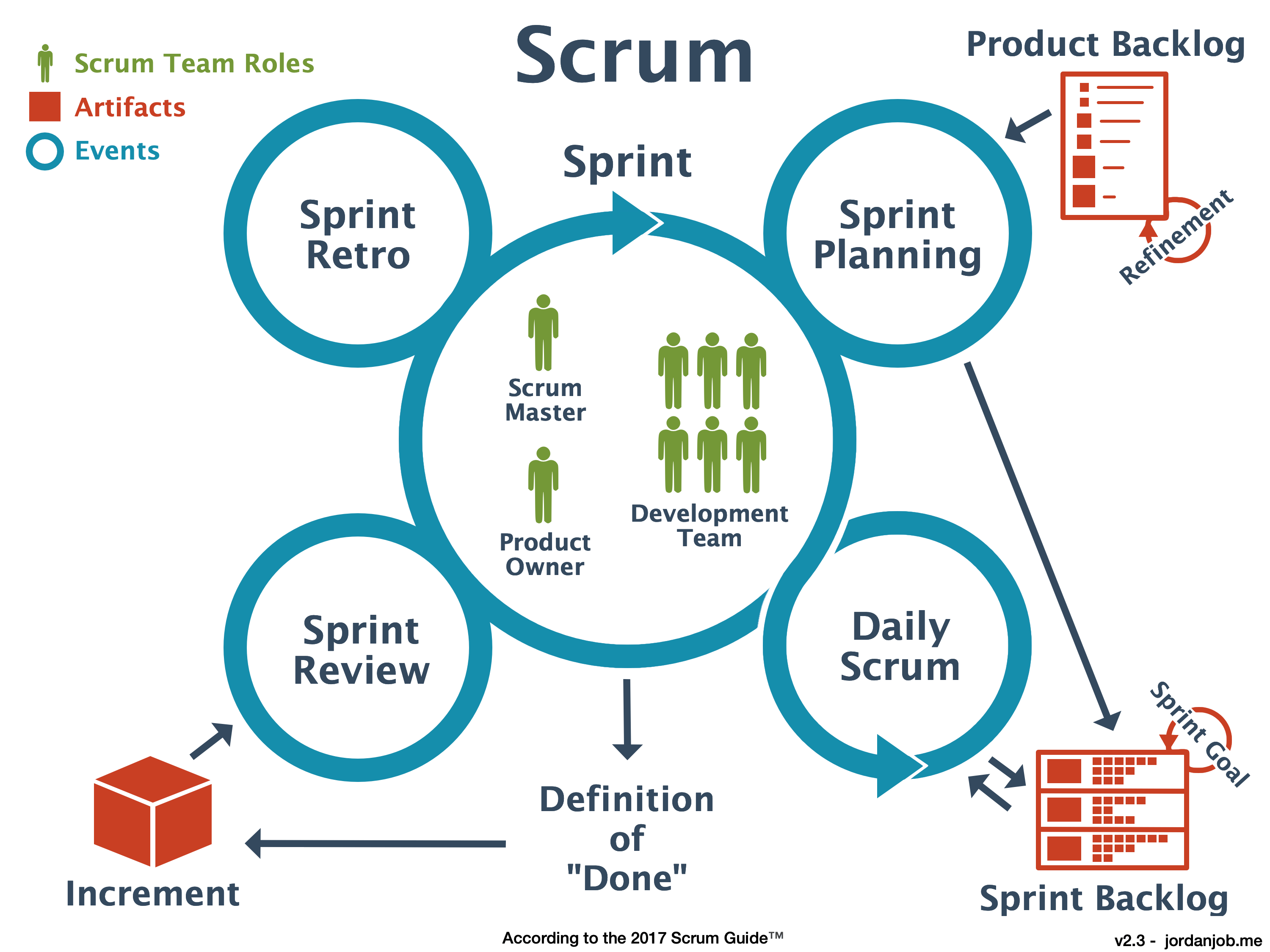


Figure 6: Overview of scrum team and process (https://jordanjob.me/blog/scrum-diagram/)

**Scrum Team:**

A Scrum team is a small group, usually 7 to 10 people, but big enough to get a lot of work done in a sprint. There are three important roles in the team: Product Owner, Scrum Master, and the Development Team.

**Scrum Roles:**

**Product Owner:**

* Responsible for understanding business requirements.
* Manages the product backlog of the project and prioritizes work for the development team.
* Work closely with the client and the team to ensure clear understanding of backlog items.
* Facilitate feature delivery and decides when to release the product.

**Scrum Master:**

* Guides teams, product owners, and the business in understanding and practicing the Scrum process.
* Understanding of the team's work deeply and making sure it's clear and delivered efficiently.
* Leads various Scrum ceremonies like sprint planning, stand-up, sprint review, and the sprint retrospective within the team.

**Development Team:**

* Usually consists of five to seven members
* The Development Team has various experts like testers, designers, UX specialists, ops engineers, and developers
* Encourages practices that ensure development is sustainable in the long term
* Team members have different skills and train each other to prevent work bottlenecks
* Collaborate with each other to complete sprints successfully.

**Scrum Artefacts:**

Scrum artifacts are essential pieces of information used by the Scrum team to define the product and plan the work.

There are three main Scrum artifacts: the product backlog(product goal), the sprint backlog(sprint goal), and the increment (definition of done).

**The Product Backlog**

It is like a big to-do list for the team, and it's managed by the Product Owner. It's a dynamic list that includes tasks, improvements, and fixes needed for the project. This list is regularly update based on priority. The Product Owner makes sure it reflects the team's current goals. As things change or new information comes in, items on the list might need to be adjusted or removed. It's a way to keep track of what needs to be done next.

**The Sprint Backlog**

It is list that includes tasks, stories and bugs. Development team works on this list during the current sprint. Before each sprint starts, the team picks what they want to do from the product backlog and creates sprint backlog. The list of things they choose can sometimes change during the sprint, but the main goal they want to achieve in that sprint stays the same and cannot be changed.

**Increment:**

Increment is the total of all the tasks finished during a sprint, along with the work done in previous sprints. It's like a usable and releasable version of the product at the end of each sprint, showing the progress made by the team. The increment helps to decide if the sprint was successful. The team shows the increment to stakeholders during the sprint review meeting. This helps everyone see the work done, get feedback, and make any needed changes.

**Scrum Ceremonies**

Scrum ceremonies are specific events within the Scrum framework that facilitate communication, collaboration, and transparency among team members.

Backlog Grooming:

Backlog grooming is a continuous task, where the Product Owner keeps updating and refining the backlog. Product owner add or remove items to backlog, add/change time estimation to each task, adjust priorities of backlog items according to client requests. This process ensures that the backlog items are clearly defined, prioritized, and ready to be included in upcoming sprints.

Sprint Planning:

The Scrum team has a meeting at the start of every sprint to plan the work for that sprint. They choose tasks (like user stories, tasks, bugs, or improvements) from the product backlog and figure out how to do them. Each task is then assigned to a specific team member.

Daily Scrum Meeting:

A quick daily meeting, often in the morning and lasting about 15 minutes. It's also called a 'daily stand-up' because it's meant to be short. The purpose is to talk about any issues or obstacles that might be stopping the team from reaching their goal for the day. This way, everyone knows what's going on, stays focused on the goal, and plans for the next 24 hours. Each team member usually answers three questions:

What did I do yesterday?

What do I plan to do today?

Are there any problems or obstacles?

Sprint review:

At the end of the sprint, the team gets together for an informal session to view a demo of, or the increment(completed sprint tasks). The development team showcases the backlog items that are now ‘Done’ to stakeholders and teammates for feedback. The product owner can decide whether or not to release the increment, although in most cases the increment is released.

This review meeting is also when the product owner reworks the product backlog based on the current sprint, which can feed into the next sprint planning session. For a one-month sprint, consider time-boxing your sprint review to a maximum of four hours.

# **References**

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

<https://www.atlassian.com/agile/advantage/agile-mindset>

<https://www.agilealliance.org/agile101/>

<https://mylifenotesweb.wordpress.com/2019/04/18/breaking-an-agile-mindset-down/>