|  |
| --- |
| **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](#_Toc151988484)

[1. Project Brief 2](#_Toc151988485)

[2. Team member 1 (Viran Pravinda) tasks 4](#_Toc151988486)

[**2.1.** **Task 1: Add project brief section to chapter 1** 4](#_Toc151988487)

[**2.2.** **Task 2: Add quick facts and organisation background to RFP** 5](#_Toc151988488)

[**2.3.** **Task 3: Add contact info and RFP process schedule to RFP** 6](#_Toc151988489)

[**2.4.** **Task 4: Add team member two tasks to chapter 1** 7](#_Toc151988490)

[**2.5.** **Task 5: Add commit history to chapter 1** 8](#_Toc151988491)

[3. Team member 2 (Sanduni Udulitha) tasks 9](#_Toc151988492)

[**3.1.** **Task 1: Fork repository** 9](#_Toc151988493)

[**3.2.** **Task 2: Add agile mindset overview to chapter 3** 9](#_Toc151988494)

[**3.3.** **Task 3: Add waterfall methodology discussion to chapter 3** 11](#_Toc151988495)

[**3.4.** **Task 4: Add scrum methodology discussion to chapter 3** 12](#_Toc151988496)

[**3.5.** **Task 5: Add guidelines for selecting the project methodology in chapter 3** 13](#_Toc151988497)

[4. Team member 3 (Dinesh Madumal) tasks 14](#_Toc151988498)

[**4.1.** **Task 1: Fork repository** 14](#_Toc151988499)

[**4.2.** **Task 2: Add decision-making criteria to RFP** 14](#_Toc151988500)

[**4.3.** **Task 3: Add project summary to RFP** 16](#_Toc151988501)

[**4.4.** **Task 4: Add the project’s target market/users to the RFP** 17](#_Toc151988502)

[5. Commit history 19](#_Toc151988503)

[**5.1.** **Commit history of team member 1 (Viran Pravinda)** 19](#_Toc151988504)

[**5.2.** **Commit history of team member 2 (Sanduni Udulitha)** 22](#_Toc151988505)

[Chapter 2 – Request for Proposal (RFP) 23](#_Toc151988506)

[1. Request for Proposal (RFP): Integrated Business System for Aussie Business Buzz (ABB) 23](#_Toc151988507)

[Chapter 3 - Software Development Methods, Processes and Techniques 28](#_Toc151988508)

[1. Agile Mindset 29](#_Toc151988509)

[2. Waterfall Methodology 32](#_Toc151988510)

[3. Scrum Methodology 35](#_Toc151988511)

[2.3 Waterfall or Scrum? 38](#_Toc151988512)

[References 40](#_Toc151988513)

# **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**

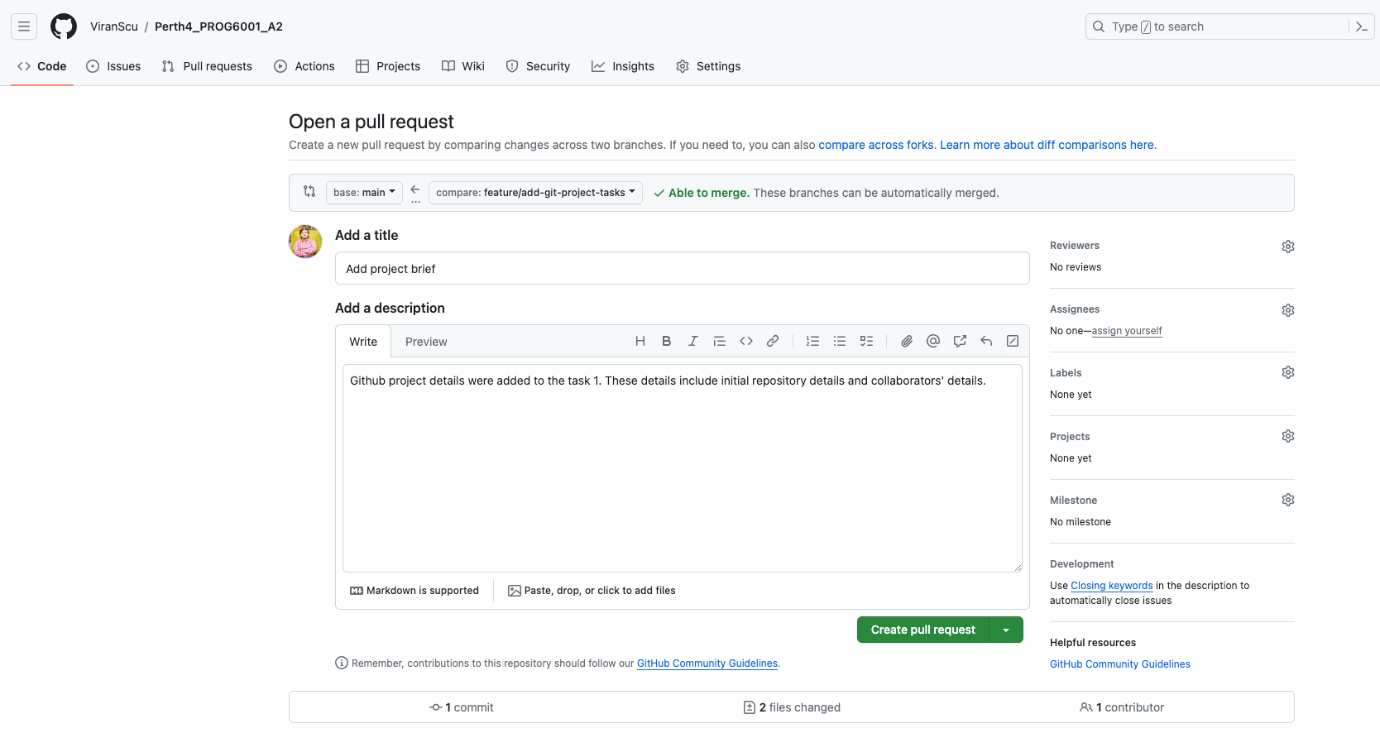
**ViranScu** is the person who created the repository, while **Dinesh** and **Udulitha** are the contributors to this project.

## **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.**

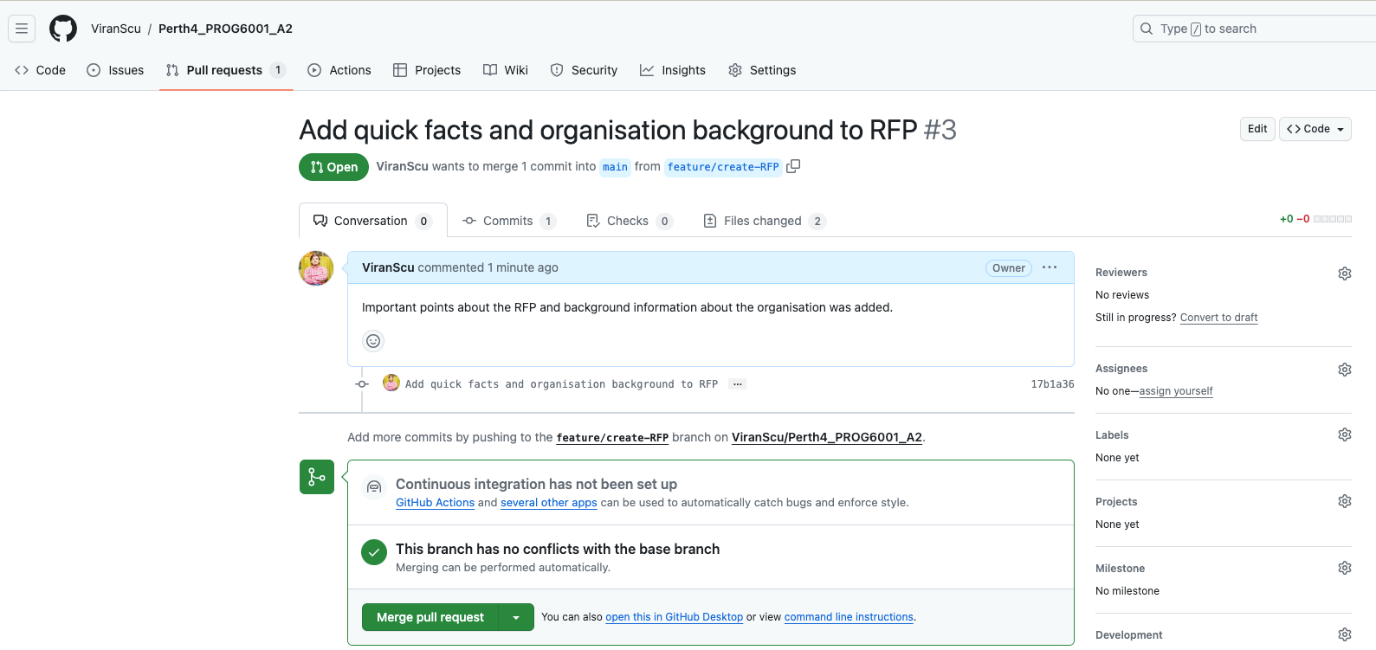
### **Task 2: Add quick facts and organisation background to RFP**

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) in chapter 2. See Figures 5 and 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 in chapter 2. Refer to Figures 7 and 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.**

### **Task 4: Add team member two tasks to chapter 1**

This task involves adding screenshots of pull requests and final merges of tasks related to team member 2. Refer to Figures 9 and 10 for the pull request and final task merge.

A screenshot of a computer

Description automatically generated

**Figure 9 - Pull request for add team member 2 tasks.**

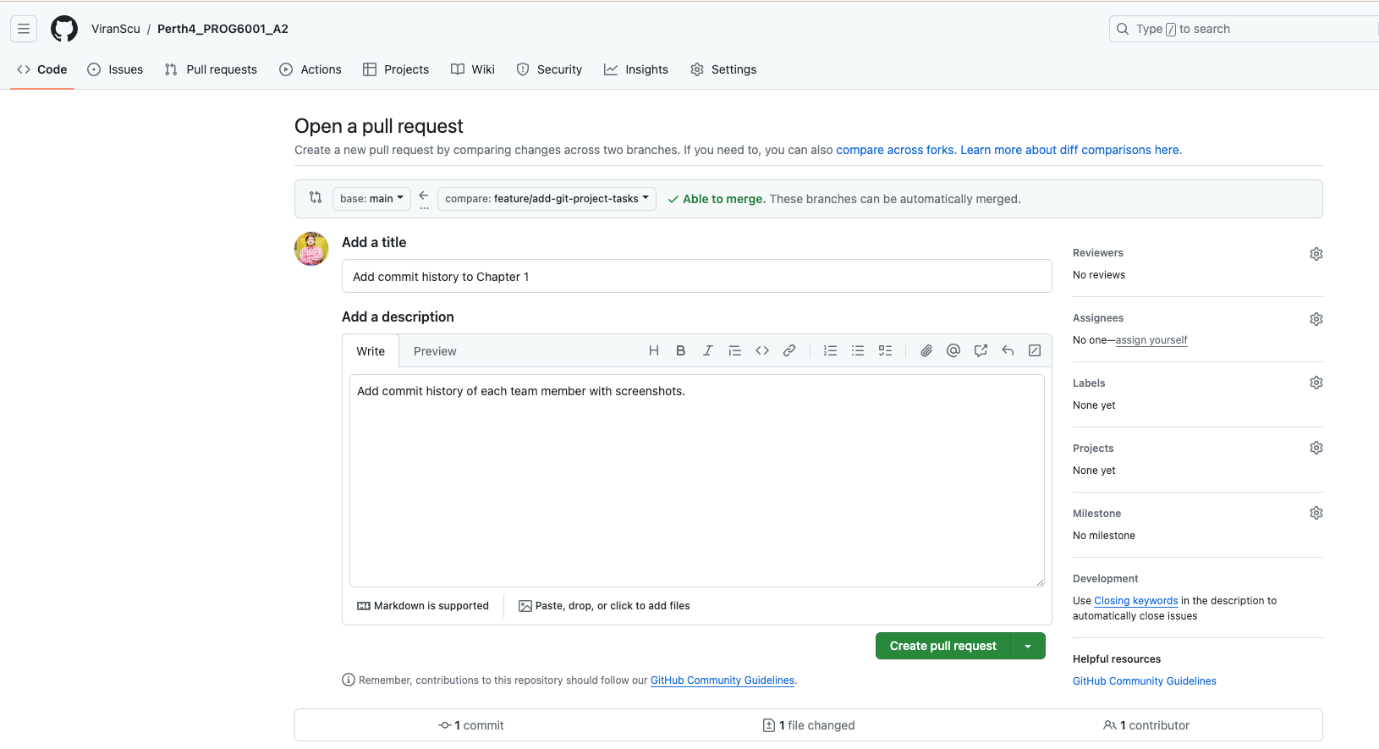
A screenshot of a computer

Description automatically generated

**Figure 10 - Final merge for add team member 2 tasks.**

### **Task 5: Add commit history to chapter 1**

This task involves adding the commit history of each team member with screenshots. See Figures 11 and 12 for the screenshots of the pull request and final task merge.



**Figure 11 - Pull request for adding commit history to chapter 1.**

A screenshot of a computer

Description automatically generated

**Figure 12 - Final merge for adding commit history to chapter 1.**

## **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 11 on creating a new fork.

**Figure 13 - 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 12 and Figure 13 for the pull request and the final merge of the task.

A screenshot of a computer

Description automatically generated

**Figure 14 – Pull request for add agile mindset overview task.**

A screenshot of a computer

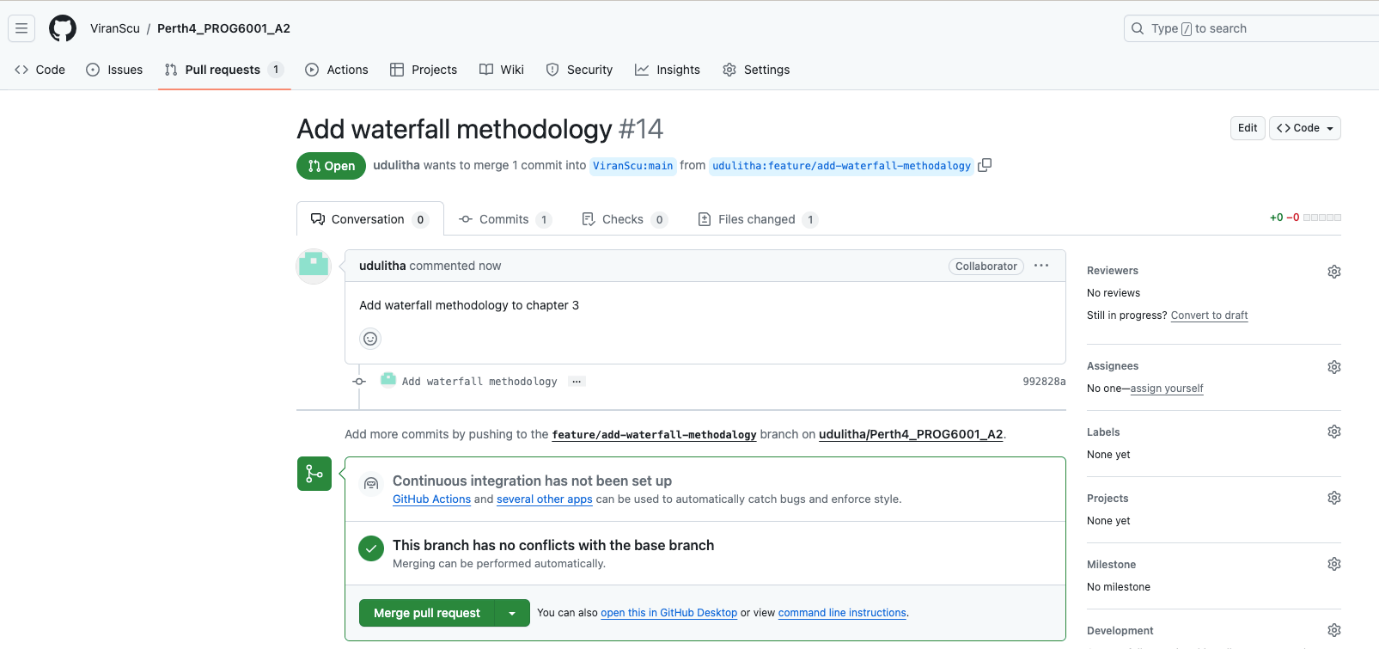
Description automatically generated

**Figure 15 – Final merge for add agile mindset overview task.**

### **Task 3: Add waterfall methodology discussion to chapter 3**

This task involves discussing the waterfall methodology in detail. See Figures 14 and 15 for the pull request and final task merge.

**Figure 16 – Pull request for add waterfall methodology discussion.**

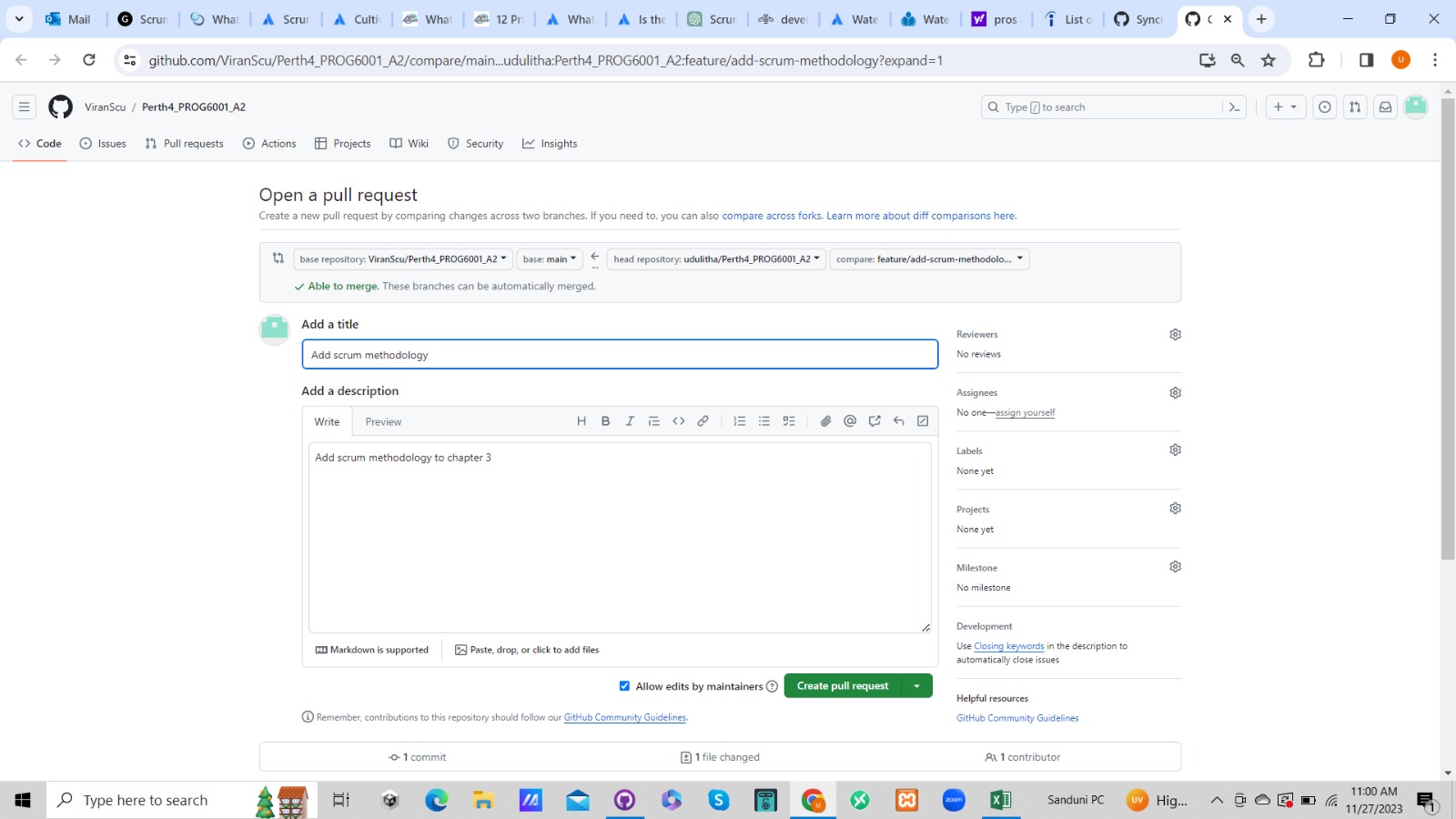
A screenshot of a computer

Description automatically generated

**Figure 17 – Final merge for adding waterfall methodology discussion.**

### **Task 4: Add scrum methodology discussion to chapter 3**

This task involves adding a discussion of the scrum methodology to chapter 3. Refer to Figures 18 and 19 for the pull request and final merge of the task.



**Figure 18 - Pull request for adding scrum methodology discussion.**

A screenshot of a computer

Description automatically generated

**Figure 19 - Final merge for add scrum methodology discussion task.**

### **Task 5: Add guidelines for selecting the project methodology in chapter 3**

This task involves the comparison between scrum and waterfall methodologies to select the best one for the given project. See Figures 20 and 21 for the pull request and final merge of the task.

A screenshot of a computer

Description automatically generated

**Figure 20 - Pull request for selecting the project methodology.**

A screenshot of a computer

Description automatically generated

**Figure 21 - Final merge for selecting the project methodology 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 22 - Create fork**

### **Task 2: Add decision-making criteria to RFP**

This task involves adding details of decision-making criteria to the Request for Proposal (RFP) in chapter 2. See Figures 17 and 18 for the pull request and the final task merge.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Figure 23 – Pull request for add decision-making criteria to RFP task.**

**Figure 24 – Final merge for add decision making criteria to RFP.**

### **Task 3: Add project summary to RFP**

This task involves adding project summary details to the Request for Proposal (RFP) in chapter 2. See Figures 19 and 20 for the pull request and the final merge of the task.

A screenshot of a computer

Description automatically generated

**Figure 25 – Pull request for adding project summary to RFP.**

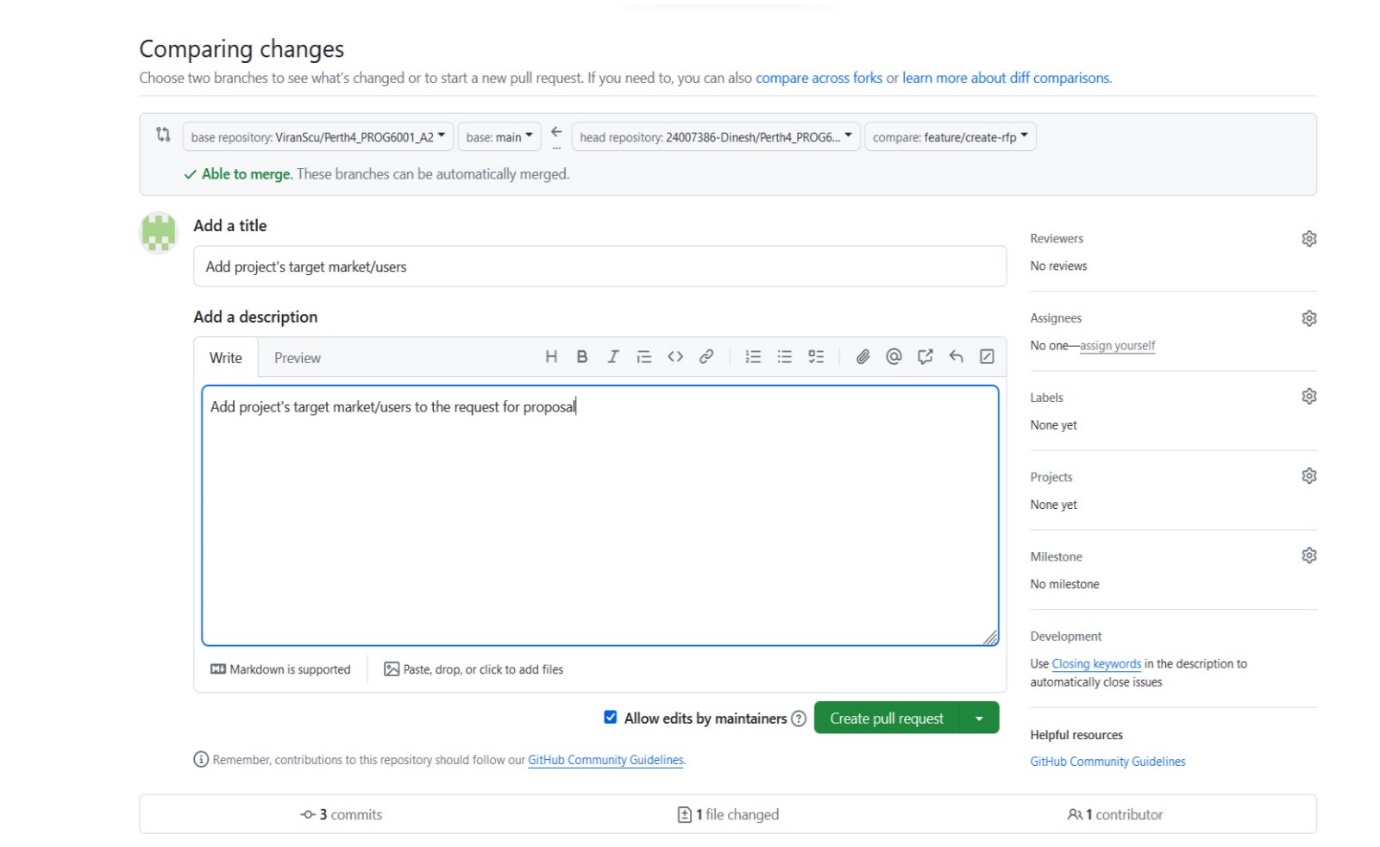
A screenshot of a web browser

Description automatically generated

**Figure 26 – Final merge for adding project summary to RFP.**

### **Task 4: Add the project’s target market/users to the RFP**

A screenshot of a computer

Description automatically generatedThis task involves adding the details of the target market/users of the Aussie Business Buzz (ABB) company’s project to the Request for Proposal (RFP) in chapter 2. See Figures 21 and 22 for the pull request and the final merge of the task.

**Figure 27 – Final merge for adding project's target market/users to RFP.**

**Figure 28 – Pull request for adding project's target market/users to RFP.**

**Please note that only some screenshots of team members’ tasks are displayed in the above sections. This demonstrates how each individual performed GitHub-related activities (fork, pull request, merge, etc.).**

## **Commit history**

This section displays the commit history of each team member who contributed to the GitHub project. The following graphs show the number of commits made by team members during the project period (see Figure 23).

A screenshot of a graph

Description automatically generated

**Figure 29 - Number of commits made by team members.**

### **Commit history of team member 1 (Viran Pravinda)**

See the below screenshots for the commit history of team member 1 (ViranScu). One commit consists of the commit title and the description. The commit title is highlighted in yellow, and the description can be seen right underneath the title.

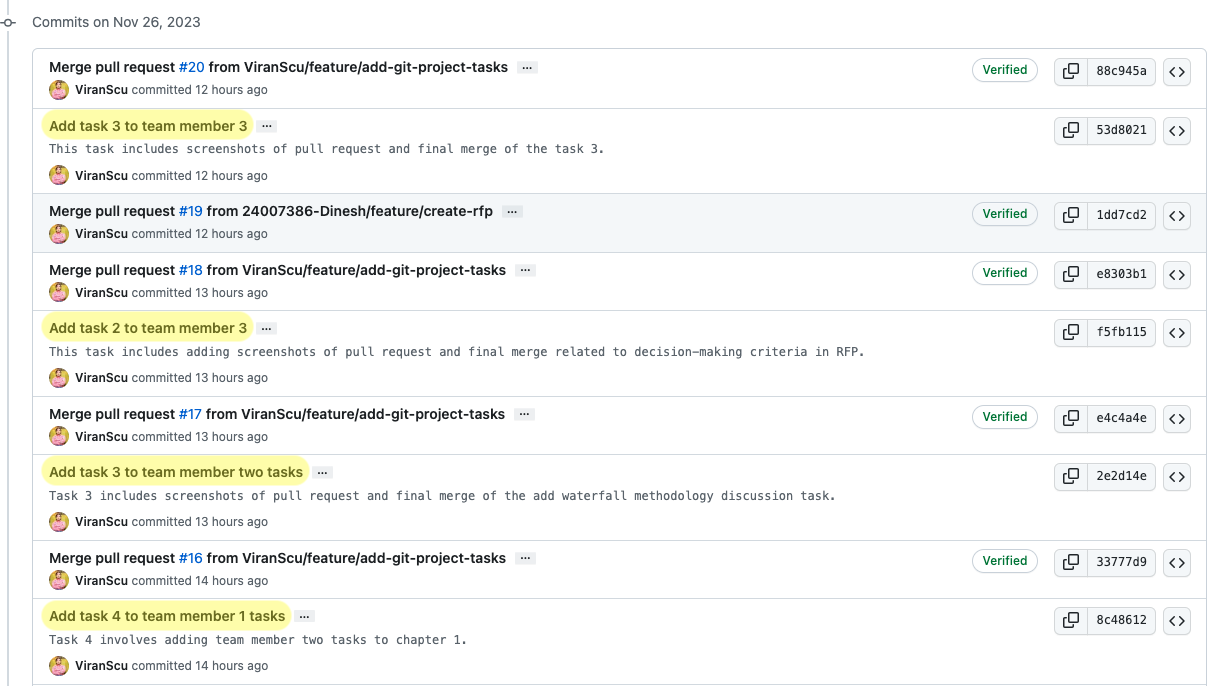


**Figure 30 - Commit history screenshot 1**

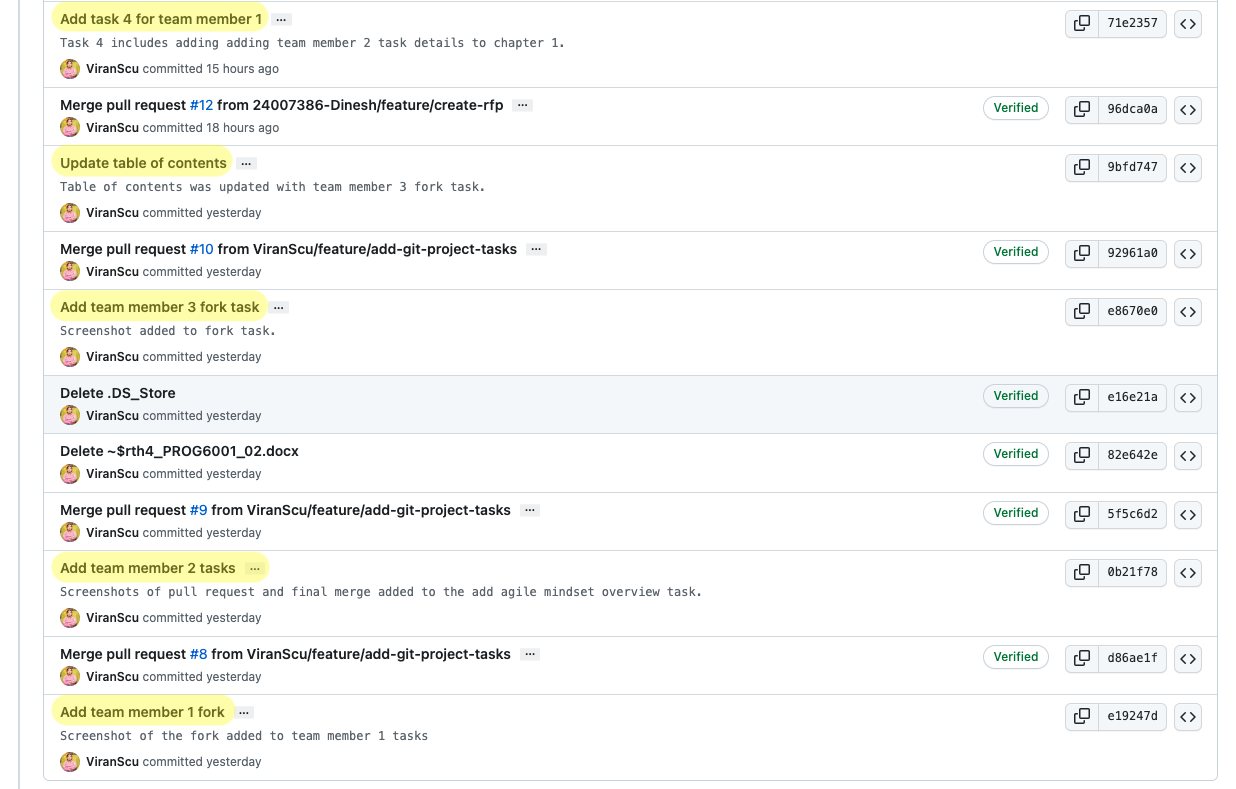
A screenshot of a computer

Description automatically generated

**Figure 31 - Commit history screenshot 2**



**Figure 32 - Commit history screenshot 3**



**Figure 33 - Commit history screenshot 4**

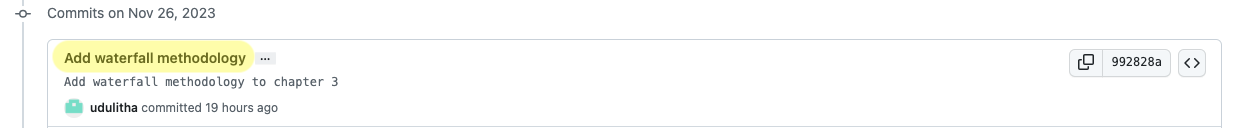
### **Commit history of team member 2 (Sanduni Udulitha)**

Screenshots related to the commit history of team member 2 (udulitha) are displayed below. Commit titles are highlighted in yellow.

A screenshot of a computer

Description automatically generated

**Figure 34 - Commit history screenshot 1**



**Figure 35 - Commit history screenshot 2**

A white background with black lines

Description automatically generated

**Figure 36 - Commit history screenshot 3**

# **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** | Our company, 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. We sell 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** | We 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** | We aim to establish an integrated business management system supporting our four branch shops, with a vision for future expansion. As a technology retail and repair business, our primary needs include,   * Customer relations database for managing product and service information, including details of devices left to repair. * Digital marketing system that includes modern marketing techniques that connect to the Customer relations database and accommodate prospective customers on our website. * Stock management system covering product and repair parts, enabling automatic ordering from wholesalers, and facilitating location-specific product searches across all our branches. * Comprehensive report for management on customer relations, marketing, and stock status to inform decision-making.   We encourage innovative solutions, allowing for bespoke software development or considering existing applications, SaaS solutions, other solutions, or any combination aligned with our evolving needs. The proposal submission deadline will be on the 22nd of December 2023, and we will announce the finalists on the 8th of January 2024. All the finalists will be selected according to the criteria mentioned above. The finalists' presentations will be on the 15th -19th of January 2024, and the final vendor will be chosen by the 24th of January 2024. The project will commence on the 5th of February 2024. |
| **Project’s Target Market/Users** | **Types of Users:**   * **Customers**: Individuals or businesses purchasing technology products, including mobile accessories, and seeking device repairs from our company * **Employees**: All our staff across our four branch shops   **Audience Targeting/Profiling:**   * **Demographics:** * **Customers:** Tech-savvy individuals or businesses * **Employees:** Various personnel with skills in customer service, marketing, etc. * **Geographics:** * Currently focus on the locations of the four branch shops and future expansion plans to cover a wider area. * **Behaviour/Psychographics:** * **Customers:** Interest in technology products, latest technology trends and seeking convenient repair services * **Employees:** Proficient with technology, flexible and capable of handling multiple responsibilities   **Size of Overall Audience/ User Base:**   * **Current:** Customers and employees in all four branches * **Projected:** As we plan to expand our business, the user base is expected to grow exponentially   **Other User Characteristics:**   * **Associated Web/ Technology:** * Customers may interact with our company through the existing website for product information and promotions. * Employees will use the integrated business management system for daily operations. * **Usage Times:** * **Customers:** Any time during the day * **Employees:** During their work shifts * **Preferences:** * **Customers:** Online interactions, including digital marketing * **Employees:** user-friendly system to manage daily tasks efficiently |

# **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 37: 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.

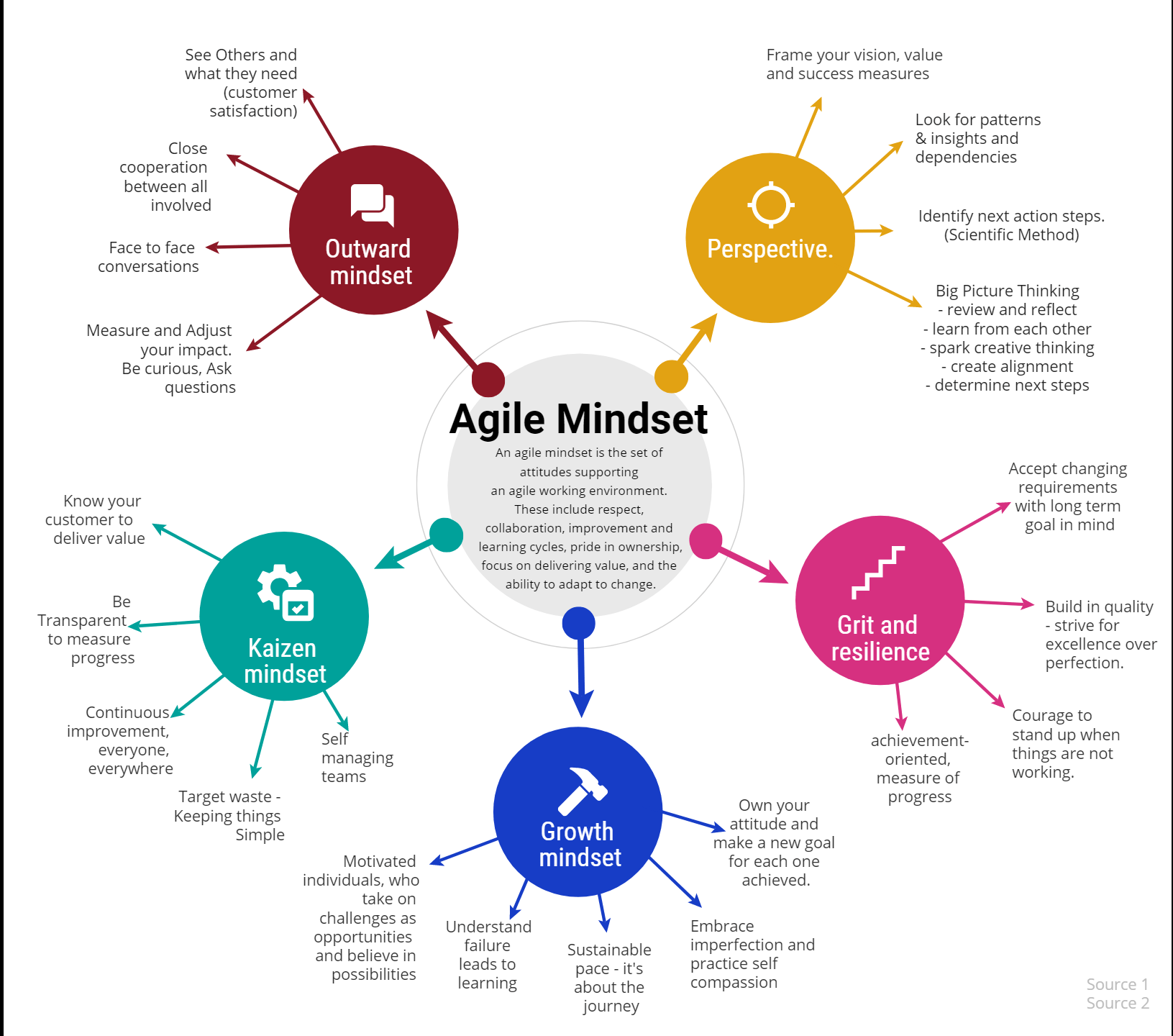


Figure 38: Agile mindset (https://mylifenotesweb.wordpress.com/2019/04/18/breaking-an-agile-mindset-down/)

**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.

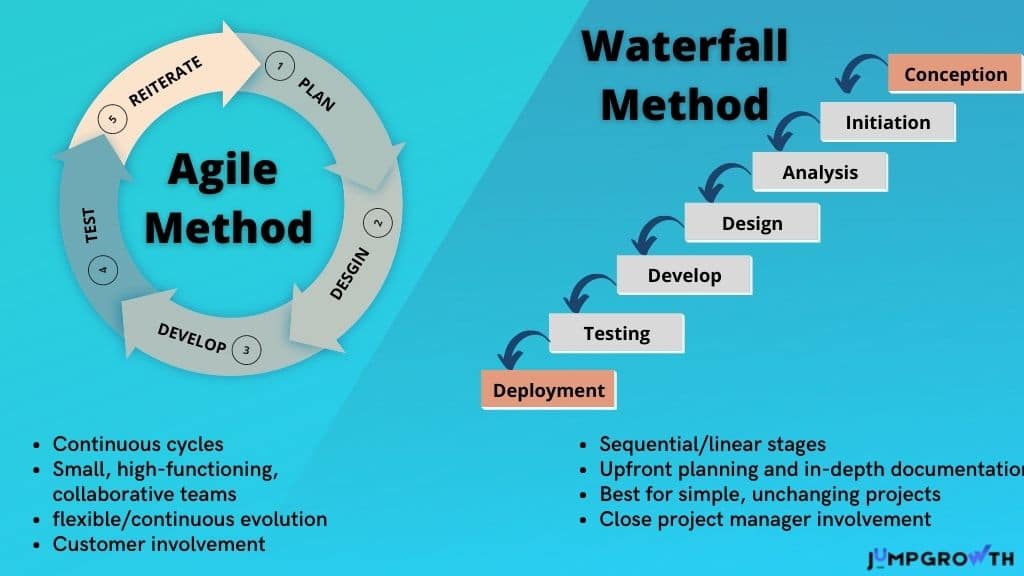
Focus on Relentless Improvement:

* 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. 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>

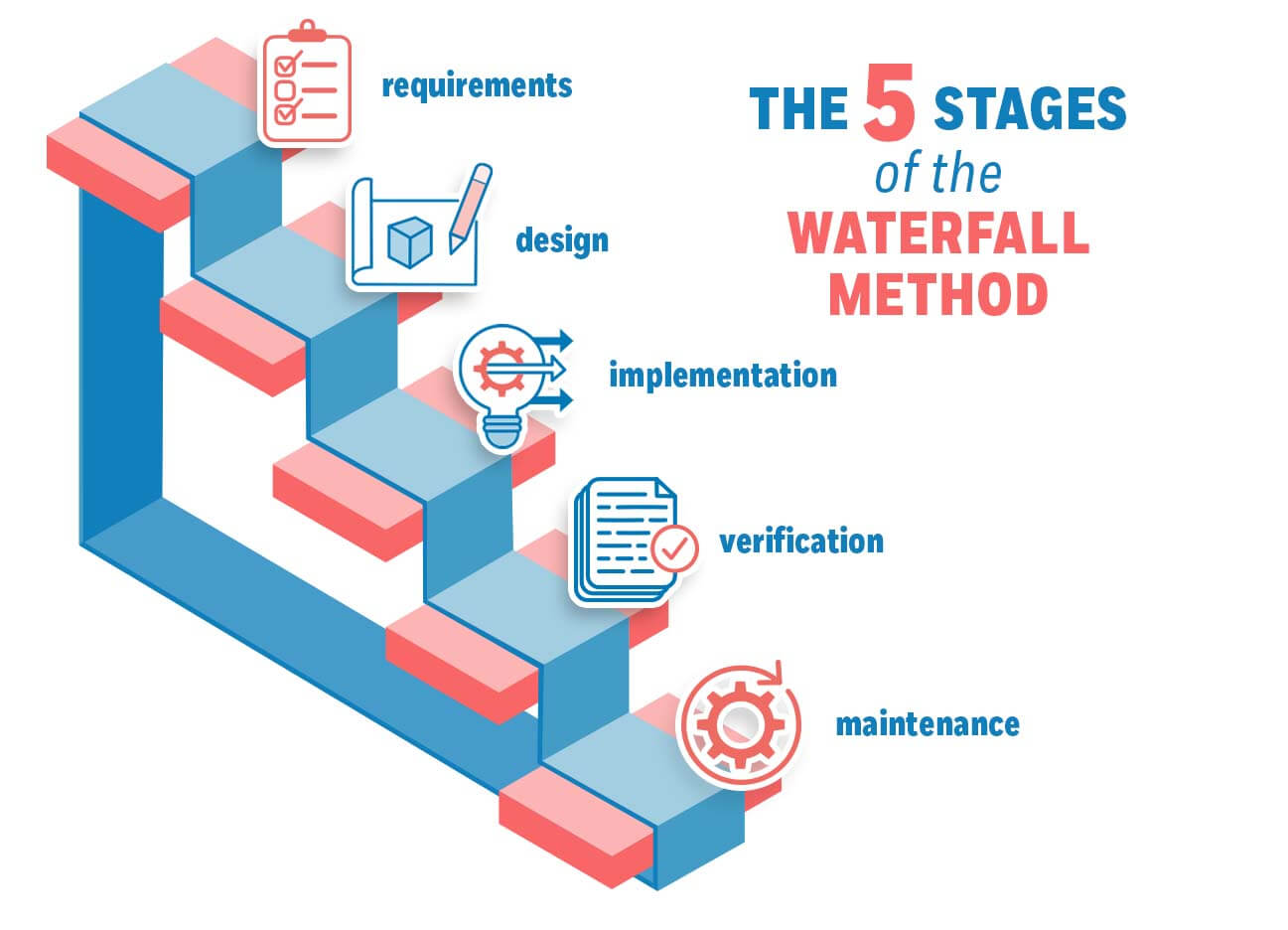


Figure 39: Stages of Waterfall Methodology (https://management.org/waterfall-methodology)

**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 and drawbacks of waterfall methodology:**

|  |  |
| --- | --- |
| **Benefits** | **Drawbacks** |
| Clear Project Structure: Everyone understands the final goal and knows what tasks to complete and when. | Longer Delivery Times: The rigid, step-by-step nature can extend the overall project duration, affecting the entire timeline. |
| Static Project Scope: Costs and timelines are determined early because the project scope remains relatively stable. | Difficulty in Time Estimation: Estimating the total duration is challenging due to organizational processes, project-specific complexities, and unforeseen delays. |
| Easier Tracking: Progress is visible quickly with simpler tasks, and tools like Gantt charts help in monitoring. | Difficulty in Welcoming New Requirements: Adapting to new requirements is challenging, often necessitating revisiting initial stages, causing potential disruptions. |
| Comprehensive Project Documentation: Waterfall provides a blueprint and historical record, offering a complete overview of the project. | Limited Client Involvement: Client involvement and feedback become limited after the requirement phase, hindering necessary improvements. |
| Improved Risk Management: Early planning reduces risks, fixing design problems before coding begins. | Lot of Change Requests: Clients may have numerous change requests, posing challenges in maintenance and causing delays. |
| Enhanced Responsibility and Accountability: Each phase has clear goals and timelines, promoting team accountability. | Deadline Creep: Issues in one phase can bring the entire process to a halt, requiring revisiting previous phases and causing delays. |
| Fewer Delays Due to Additional Requirements: Completing the design early reduces extra requests, preventing delays and keeping the cost of changes low. |  |
| Easy Adaptation to Key Member Changes: Detailed documentation describes required skills, making transitions easy. |  |

* <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>
* <https://www.atlassian.com/agile/project-management/waterfall-methodology>

## **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>

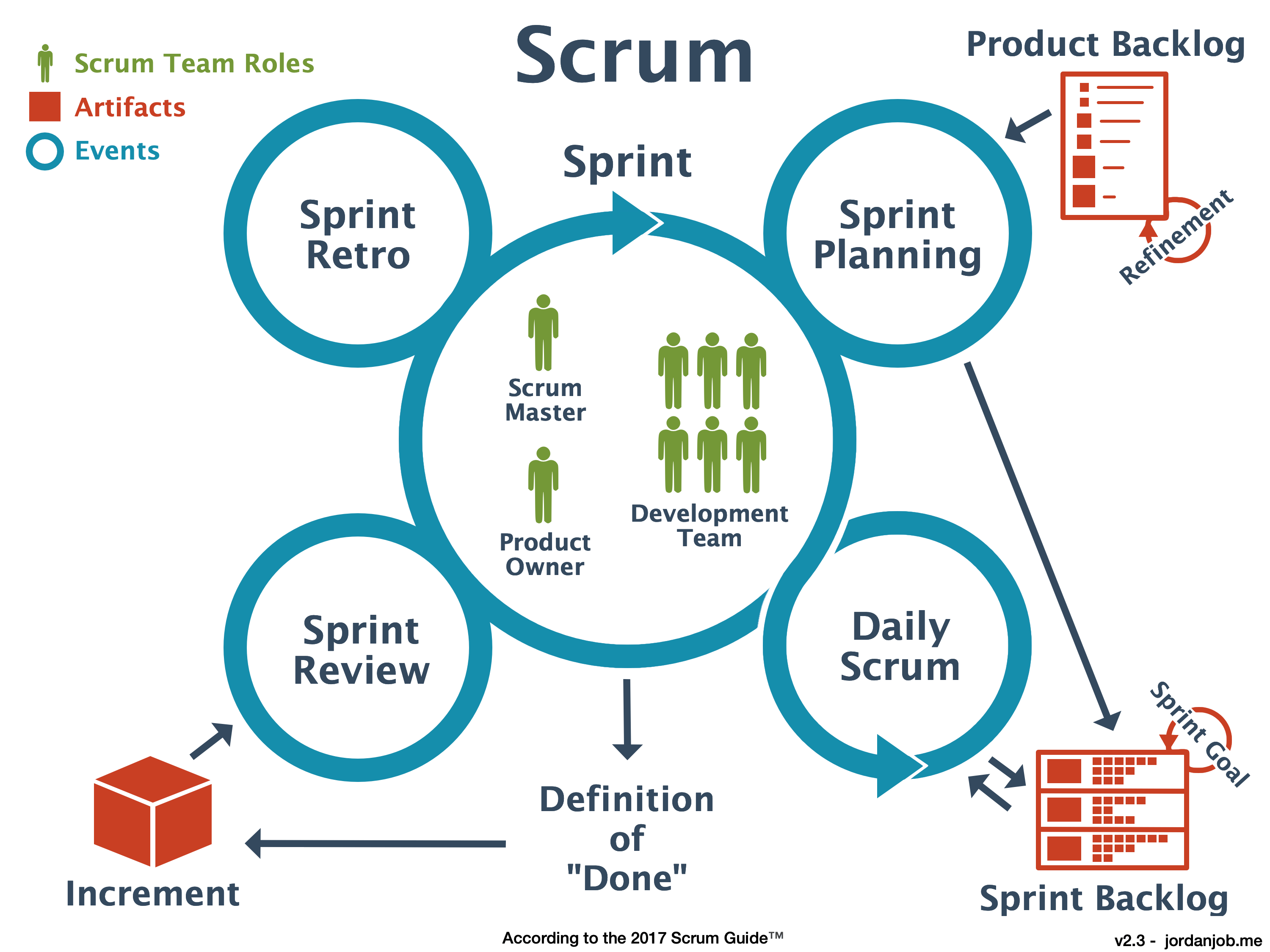


Figure 40: Overview of scrum team and process (<https://jordanjob.me/blog/scrum-diagram/>)

**Sprint**

An incremental cycle in scrum is called a sprint with a typically fixed duration, commonly ranging from two to four weeks. During a sprint, the team focuses on completing a set of prioritized tasks outlined in the sprint backlog. This iterative and time-boxed approach allows teams to make consistent progress, enhance flexibility, and respond effectively to evolving project needs. (blackboard unit)

**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.

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

There are three main Scrum artefacts: 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 gathers for a demo of the completed tasks. The development team demonstrates the finished work to stakeholders for feedback. Then team decides if it's ready to release to the client. In this meeting, the product owner updates the product backlog for the project based on what was done in the current sprint. This updated list helps plan for the next set of tasks in the upcoming sprint.

Retrospective Meeting

After the sprint release the team gathers to talk about what went well and what didn't in the sprint. They discuss things like the project, people, relationships, tools,. The goal is to make a space where the team can concentrate on the good things and figure out what can be better next time, rather than just focusing on what went wrong.

**Benefits and drawbacks of Scrum methodology:**

|  |  |
| --- | --- |
| **Benefits** | **Drawbacks** |
| **Adaptability:** Scrum is highly flexible. | **Training Challenges:** Effective Scrum needs a well-trained team. |
| **Creativity Boost:** Collaborative Scrum encourages creative approaches. | **Scaling Difficulties:** Adapting Scrum to large projects is tough. |
| **Cost-Effective:** Scrum is often cost-effective. | **Organizational Changes:** Scrum adoption may demand significant transformations. |
| **High-Quality Work:** Scrum promotes ownership and responsibility. | **Integration Issues:** Scrum may not seamlessly integrate with classic project management. |
| **Customer Satisfaction:** Constant adjustments lead to popular and satisfying products. | **Deadline Management:** Scrum involves smaller deadlines, lacking explicit support. |
| **Employee Satisfaction:** Scrum's emphasis on individual responsibility motivates and satisfies employees. | **Team Size Constraints:** Scrum works best with small teams. |
|  | **Experienced Personnel Needed:** Successful Scrum adoption relies on experienced team members. |

<https://www.indeed.com/career-advice/career-development/disadvantages-of-scrum>

## **2.3 Waterfall or Scrum?**

Selecting between Scrum and Waterfall for a software project depends on various factors. Table 1 outlines criteria that spotlight the key differences between Scrum and Waterfall methodologies. These considerations are essential when deciding on a project methodology.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Waterfall** | **Scrum** |
| **Project Type** | Ideal for projects with well-defined requirements and minimal changes | Well-suited for projects where adaptability, short delivery cycles, and continuous value delivery are crucial |
| **Project Size** | Often preferred for small projects | Often preferred for small to medium projects |
| **Flexibility and Adaptability** | Limited flexibility in accommodating changes once the project has started | Promotes adaptability, enabling teams to pivot on ideas and plans without disrupting the project flow |
| **Project Control** | Provides more control from the project's outset, with a well-defined plan | Offers less control over the final outcome initially but allows for continuous improvement |
| **Repeatable Processes** | Works well for repeatable processes with predictable outcomes | More suitable for projects with evolving requirements and outcomes |
| **Team and Responsibilities** | Follows a more rigid structure with defined roles and responsibilities for each phase | Precisely defines roles and responsibilities, focusing on process control. Promotes collaborative, cross-functional teams with frequent communication |
| **Delivery Speed** | Longer delivery times, typically waiting months or years for software release especially for complex projects | Enables faster value delivery through short sprints, making it suitable for dynamic projects |
| **Client Involvement** | Client involvement mainly occurs at the beginning and end of the project | Regular client feedback and involvement throughout the development process. |
| **Risk Management** | Risks should be identified and addressed in the planning phase | Risks can be Identified and addressed iteratively throughout the development process. |
| **Cost Management** | Costs are determined upfront and remain relatively fixed | Cost will be more less than waterfall due to adaptive planning and minimal documentation |

<https://www.float.com/resources/scrum-vs-waterfall/>

Even though waterfall and scrum are different, a "Hybrid approach" is an option if it suits the project. It lets the project team combine the advantages of waterfall with the flexibility of scrum. In this model, requirements gathering and design follow a waterfall approach, while implementation uses scrum iterations. This is helpful for organizations shifting to agile allowing flexibility for adapting to changes during projects. <https://www.float.com/resources/scrum-vs-waterfall/>

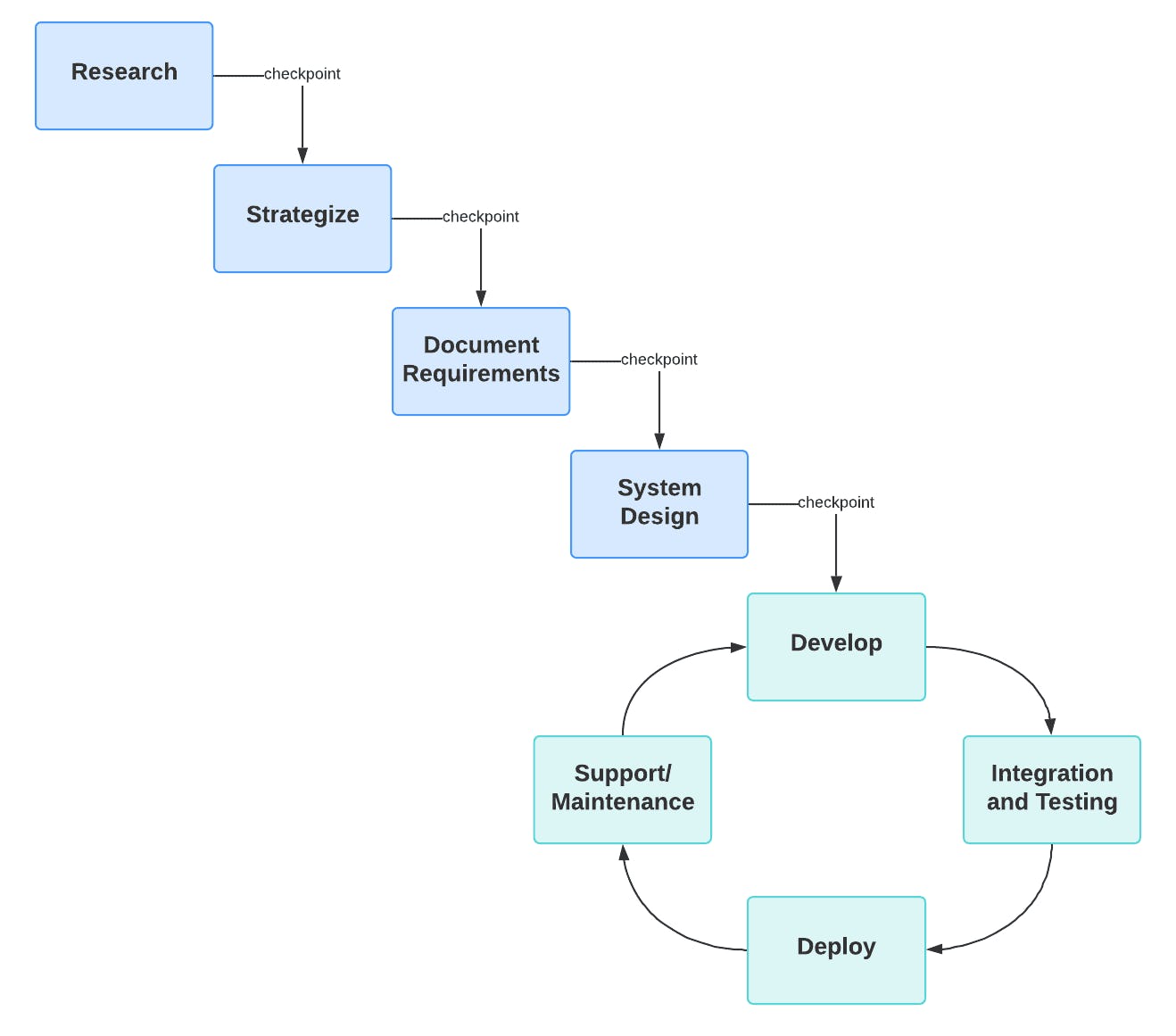


Figure : Hybrid approch (https://www.float.com/resources/scrum-vs-waterfall/)

# **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/>

<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/>

<https://jordanjob.me/blog/scrum-diagram/>

<https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint>

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

<https://www.slingshotapp.io/blog/scrum-vs-waterfall-how-to-choose-the-right-method-for-your-project>

<https://www.float.com/resources/scrum-vs-waterfall/>