# Technical Process

## Methodology

In the selection process for a software methodology, our team considered a variety of options. The following software methodologies were considered:

* Waterfall Model: This is a linear and sequential model, implying that the next development phase cannot occur until the previous phase is completed and upon completing a phase one cannot simply iterate back. This methodology is easy to use and understand, however, it will not be used due to the following reasons:
  + No overlapping phases, implying each phase must be completed before the next iteration.
  + Testing comes late and difficult to make changes.
* V-Model: The Verification and validation model executes processes sequentially, and software development and testing take place at the same time. The advantages are that defects are detected early on, however this methodology will not be used due to the following reasons:
  + Not suitable for complex, ongoing projects and projects where there is uncertainty in the requirements.
* Incremental Model: The model divides software development processes into several increments and the same phases are followed each increment. The advantages are that the detection of errors is easy to identify, however, this methodology will not be used due to the following:
  + Complete requirements of software must be available from the start.
* RAD Model: Rapid Application model is based on prototyping and iterative model with less or no planning. Reduced development time, flexibility, suitable for small projects is a great advantage however this methodology is not suitable due to the:
  + Difficult to manage, and not suitable for complex projects with large life spans.
* Iterative Model: Software is developed by following several iterations, each iteration implies following the development process, this is repeated until the project is completed. Iterations allow for quick error and bug handling, however, this methodology is not suitable due to the following:
  + Not suitable for small projects, difficult to estimate the end date for the project, constantly changing requirements implies frequent change to software.
* Spiral Model: Each “spiral” loop is a phase in the development process. Each phase contains determining objectives and alternative solutions, identifying, and resolving risk, developing, and testing, and reviewing and planning the next phase. The advantages are that it is suitable for large and complex projects, and continuous customer feedback implies greater satisfaction. However, the model is not chosen due to the following:
  + Requires more documentation than others, success of the model depends highly on the risk analysis phase, a high-cost model.
* Prototype Model: Protypes of projects are created initially and developed upon. The advantages are that it is good when the general objective of creating software is known but not the input, processing, or output. However, the methodology is not suitable due to:
  + Compromises can be seen in the prototype version; end-users might want to small fixes on the prototype instead of a rebuilt version.
* Agile Model: Follows an iterative development method, whereby in each iteration a small and easily manageable task can be completed within a couple of weeks. This focuses on customer satisfaction and making quick changes during the middle of the development process. It has different models: scrum, crystal, feature-driven, lean and XP models. The advantages are dual programming which decreases errors, projects are completed in a short time frame, flexible to developers, realistic and managed easily. The disadvantages are lack of documentation and depend on customer representative information to build software correctly.

Considering the above advantages and disadvantages of the listed methodologies, the team reached a consensus on using Agile methodology. The model chosen was Scrum. Based on Peek (2023) the scrum model breaks the project down into sizable chunks called sprints. Agile scrum methodology is an industry norm methodology that produces projects quickly. This methodology relies on incremental development with each iteration commonly being within two to four weeks. The focus is to build the most important features first and deliver a potential product. More features are added in subsequent sprints and are adjusted based on stakeholder and customer feedback. The agile scrum has five phases are: initiation, planning and estimation, implementation, reviewing and releasing. Figure 1 displays the phases of agile scrum.

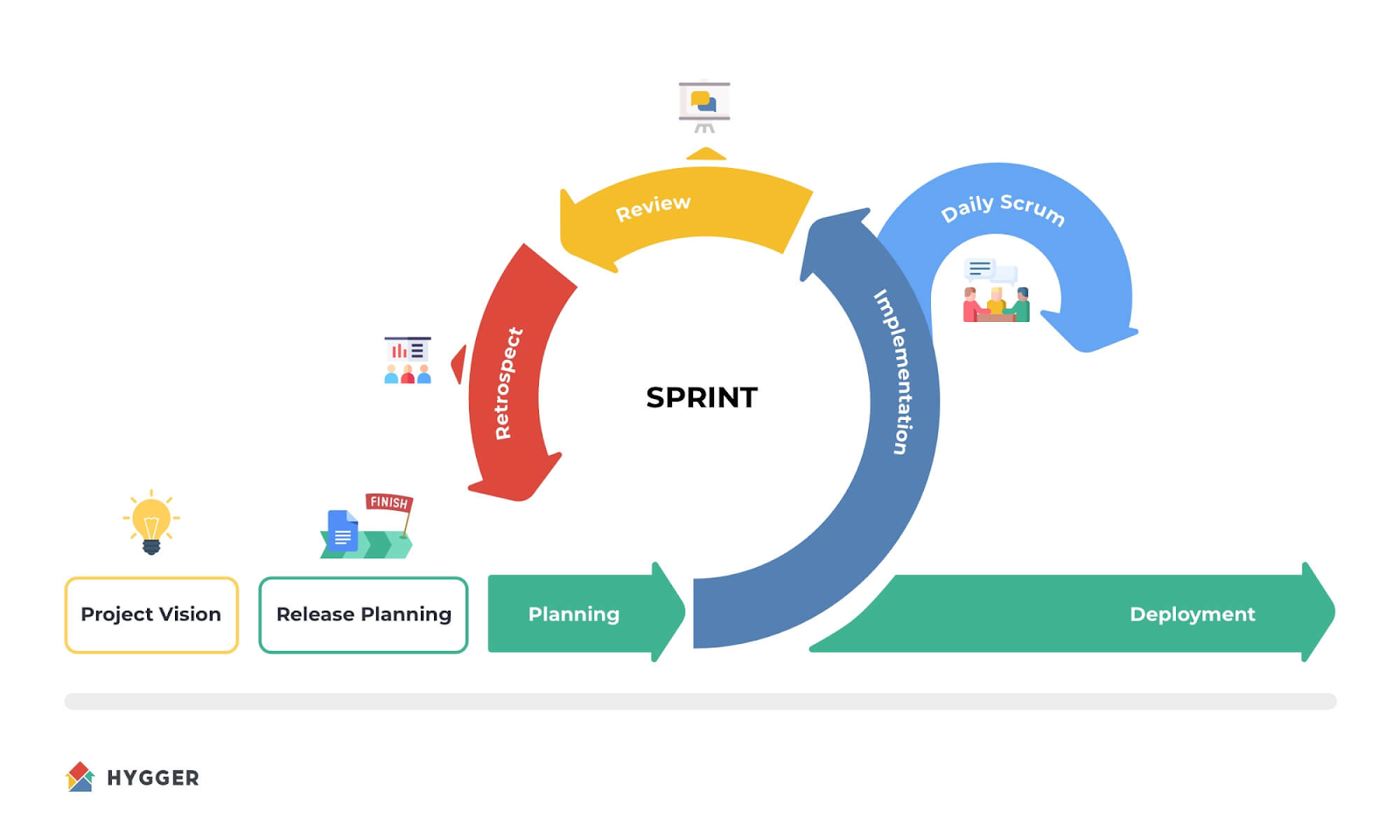


Figure 1: Agile Scrum

The benefit of agile scrum is it focuses on fast production since each goal must be completed within each sprint. It also requires frequent planning and goal setting, which increases team productivity on focus, flexibility, and adaptability where each sprint customer feedback is received and adapted in the next iteration. (Peek, 2023). The disadvantage is that it requires an expert in the company to ensure the principles are applied correctly. Agile scrum methodology involves precise execution and could result in problems if not done properly (Peek, 2023). The team has undergone agile training and is more than equipped to develop the project using this methodology.

The team will implement the Agile methodology in a two-weeklong iteration, which will provide for approximately eight iterations per semester. Given that each iteration has 5 phases, the team has agreed to a two-weeklong iteration which should provide enough time to go through each phase and receive frequent client feedback to produce a valuable project. The schedule will be evident and documented in the project schedule, which will be managed in the team Excel Sheet.

## Tools and Techniques

|  |  |
| --- | --- |
| Project Management Requirements | Tools and Techniques |
| Version control | **Git & GitHub** – The team has experience with using git & GitHub, GitHub allows for easy version control of code in a large team in a private repo. |
| Project information website | **GitHub Pages** – Allows us to freely host a project website on GitHub using HTML and CSS. |
| Scheduling software | **Excell** – The team is already using Excell as a scheduling method. The use of Excel allows for easy and fast scheduling and the use of equations can help with related tasks such as time tracking. |
| Team communication | **Teams** – The team is already using Teams as a communication method. Teams provide the following advantages: no restriction on meeting time, sharing of documents and resources, scheduling of team meetings, and recording of meetings. |
| Task Management | **GitHub Kanban** – The team is experienced with using GitHub Kanban as task management software. GitHub Kanban is easy to use and can be set to meet the needs of the team, each member can be assigned a task and can update their tasks as they progress. |
| Documentation Storage/Management | **Google Drive (App) & GitHub (code) -**  Google Drive is convenient and free to use, it promotes easy storage of applications, and documentation, and tracks history to aid in project documentation.  GitHub is an industry norm for storing code of the project in a safe and secure environment and promotes version control and branching which will allow the team to work coherently without breaking each other’s code. |
| Continuous Integration | **Jenkins** - is an open-source automation server. It aids in software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. |

## Internal Artifacts

Agile scrum artefacts are information that a scrum team and stakeholders use to detail the product being developed, the actions to produce it, and the actions performed during the project (Harris, 2024). The main artefacts are product backlog, sprint backlog and increments (Harris, 2024), they are described in detail below:

* Product backlog: a list of new features, enhancements, bug fixes, tasks, or work requirements needed to build a product. It’s compiled from input sources like customer support, competitor analysis, market demands, and general business analysis (Harris, 2024). This will be kept up to date as new information is available.
* Sprint backlog: a set of product backlog tasks that have been promoted to be developed during the next product increment. Sprint backlogs are created by the development teams to plan deliverables for future increments and detail the work required to create the increment (Harris, 2024). A Sprint backlog will be created by selecting tasks from the product backlog and breaking them into smaller actionable sprint items.
* Increments: the customer deliverables that were produced by completing product backlog tasks during a sprint. It includes the increments of all previous sprints. There is always one increment for each sprint and an increment is decided during the scrum planning phase. An increment happens whether the team decides to release to the customer. Product increments are incredibly useful and complementary to CI/CD in version tracking and, if needed, version rollback (Harris, 2024). Increments will be aligned to each backlog item, by creating a branch and build for each backlog item.

# References

GeeksforGeeks. 2023. Top 8 Software Development Life Cycle (SDLC) Models Used in Industry. <https://www.geeksforgeeks.org/top-8-software-development-models-used-in-industry/> Date of access: 8 May. 2024.

Peek, S. 2023. What is Agile Scrum Methodology? <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html> Date of access: 11 May. 2024.

Harris, C. 2024. Agile scrum artifacts. <https://www.atlassian.com/agile/scrum/artifacts#:~:text=The%20main%20agile%20scrum%20artifacts,archaeological%20ruins%20and%20ancient%20relics>. Date of access: 11 May. 2024. ()