**Advantages of scrum**

* Transparency
  + An Agile approach provides a unique opportunity for clients to be involved throughout the project, from prioritizing features to iteration planning and review sessions to frequent software builds containing new features. However, this also requires clients to understand that they are seeing a work in progress in exchange for this added benefit of transparency.
* Early and Predictable Delivery
  + By using time-boxed, fixed schedule Sprints of 1-4 weeks, new features are delivered quickly and frequently, with a high level of predictability. This also provides the opportunity to release or beta test the software earlier than planned if there is sufficient business value.
* Allows for Change
  + While the team needs to stay focused on delivering an agreed-to subset of the product’s features during each iteration, there is an opportunity to constantly refine and reprioritize the overall product backlog. New or changed backlog items can be planned for the next iteration, providing the opportunity to introduce changes within a few weeks.
* Improves Quality
  + By breaking down the project into manageable units, the project team can focus on high-quality development, testing, and collaboration. Also, by producing frequent builds and conducting testing and reviews during each iteration, quality is improved by finding and fixing defects quickly and identifying expectation mismatches early.

**Disadvantages of scrum**

* Harder for new Starters to Integrate in the Team
  + Agile requirements are normally barely sufficient. Requirements are clarified just in time for development and can be documented in much less detail due to the timeliness of conversations. This can mean less information available to new starters in the team about features and how they should work.
  + It can also create potential misunderstandings if the teamwork and communication aren’t at their best, and difficulties for team members (especially testers) that are used to everything being defined up front.
* Can be Very Demanding on the Users Time
  + Active user involvement and close collaboration are required throughout the development cycle. This is very engaging, rewarding and ensures delivery of the right product. It’s the fundamental principle in agile that ensures expectations are well managed. And since the definition of failure is not meeting expectations, these are critical success factors for any project. However these principles are very demanding on the user representative’s time and require a big commitment for the duration of the project.
* Costs can Increase as Testers Required all the Time Instead of at the End
  + Testing is integrated throughout the lifecycle. This helps to ensure quality throughout the project without the need for a lengthy and unpredictable test phase at the end of the project.
  + However it does imply that testers are needed throughout the project and this effectively increases the cost of resources on the project. This does have the effect of reducing some very significant risks that can cause many projects to fail. The cost of a long and unpredictable test phase can, in my experience of waterfall, cause huge unexpected costs when a project over-runs. However there is an additional cost to the project to adopt continuous testing throughout.
* Agile can be Intense for the Team
  + Finally, common feedback is that agile development is rather intense for developers. They need to really complete each feature 100% within each sprint or iteration, and the relentlessness of iterations, can be mentally quite tiring so it’s important to find a sustainable pace for the team.

**The efficiency of scrum given the time it took to use**

Since scrum using time-boxed, fixed schedule Sprints of 1-4 weeks, new features are delivered quickly and frequently, with a high level of predictability. This also provides the opportunity to release or beta test the software earlier than planned if there is sufficient business value.

**When to use scrum**

Success is defined by responsiveness to customer requests. Requires quick turnaround (2—4 week Sprints) for high-priority requests. Tightest constraint is on schedule, to achieve quick turnaround. Scope is adjusted to fit schedule. Requirements change frequently, even from month to month. Change is the norm, and requests are re-prioritized at Sprint boundaries. Work requires constant invention, so all steps are not known in advance, and estimates are not expected to be reliable. Process is cyclic: It repeats every Sprint, and planning for next Sprint overlaps with the work on the current Sprint. No steps involve long lead times or lots of specialized resources. Incremental results (e.g., at 20% of scope) have significant value

**When not to use scrum**

Success is defined by achieving the planned scope. Tightest constraint may be on scope or schedule. Sometimes schedule is extended to achieve scope. Sometimes scope is reduced to achieve schedule. Requirements are well-understood and will not change. Change requests represent exceptions that must be handled by a change-request process. All steps are known and can be estimated with reasonable accuracy. Process is linear: Starts with requirements, leads to results, and stops. Some steps may involve long lead times, or lots of specialized resources. Incremental results (e.g., at 20% of scope) have little or no value

**How we work together as a group in the project**

First is familiarization, this is the stage when the individual members of the group get to know each other and begin to understand the task they need to undertake. Time spent at this stage discussing our individual areas of interest and skills will be invaluable in helping our group develop a sense of own identity (including its strengths and weaknesses). Make sure everyone understands what it is they will need to achieve like the product backlog, the time scale and the product scale.

Second is planning and preparation, This is the stage when our group plan exactly what needs to be done, how it needs to be done, and who should do what. Agree the different elements of the task agree the best way of achieving these tasks by dividing areas of responsibility amongst the group, making sure that roles and time commitments are as evenly balanced as possible. Make the most of our different areas of expertise by dividing tasks up according to the skills of different group members. Make an action plan of what needs to be done by when, working towards the final deadline.

Third is the implementation, while our group carries out its tasks we will need to preserve our group's sense of purpose. Effective communication is vital, particularly when our group activity extends over time. To promote good communication, we share addresses, telephone numbers and email addresses at an early stage to facilitate contact between members of the group. We even set up an email distribution list for rapid communication so that issues or problems can be flagged up as and when they arise. We also stables regular meetings of the whole group to check on progress and review action plans. Take notes at these meetings to help record complex discussions.