**Homework: Lesson 10 – Agile Techniques**

**Question 1**

Complete definitions for Scrum related key terminology provided below.

**SCRUM CEREMONIES**

• Product backlog refinement - Product Backlog Refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised.

• Sprint planning - Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. The product owner describes the objective (or goal) of the sprint and what backlog items contribute to that goal.

• Daily scrum - the purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work. The Daily Scrum is a 15-minute event for the [Developers](https://www.scrum.org/resources/what-is-a-scrum-developer) of the Scrum Team. Their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management. Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

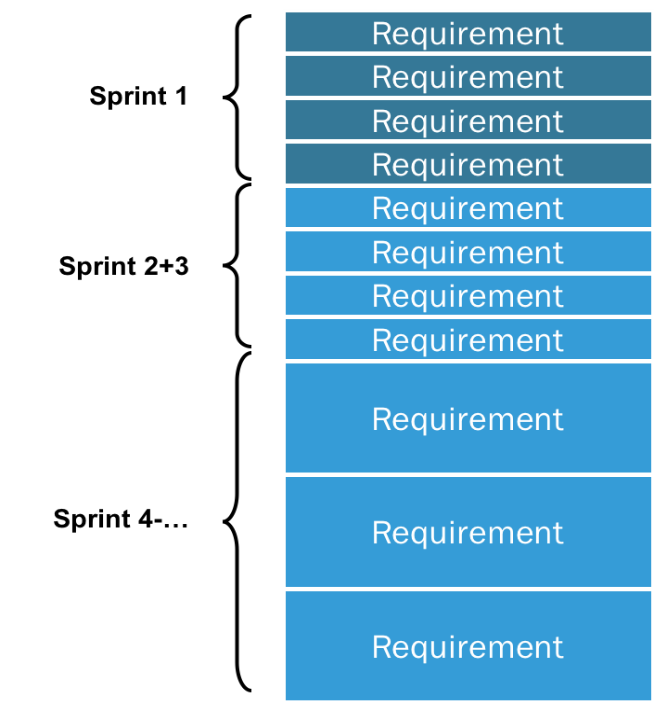
• Sprint review - the purpose of the Sprint Review is to inspect the outcome of the [Sprint](https://www.scrum.org/resources/what-is-a-sprint-in-scrum) and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed. During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next.

• Sprint retrospective - The sprint retrospective is a recurring meeting held at the end of a sprint used to discuss what went well during the previous sprint cycle and what can be improved for the next sprint. The Agile sprint retrospective is an essential part of the Scrum framework for developing, delivering, and managing complex projects.

**SCRUM ROLES**

• ScrumMaster - The scrum master is the team role responsible for ensuring the team upholds agile values and principles and follows the processes and practices that the team agreed they would use. They ensure good relationships between the team and product owner as well as others outside the team are established.

• Product Owner - a Scrum Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team.



**Product Backlog**

The Product Owner is also accountable for effective Product Backlog management, which includes:

* Developing and explicitly communicating the Product Goal
* Creating and clearly communicating Product Backlog items
* Ordering Product Backlog items
* Ensuring that the Product Backlog is transparent, visible and understood

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

• Development Team - A development team is a group of people that work together to create software. This is complex, creative work that requires adaptability as technical challenges arise and business requirements evolve. An agile development team will seek to meet these challenges by applying the principles of cross-functionality and self-organisation.

* Cross-functional - Rather than organising people into specialist teams such as analysts, coders, and testers; agile development teams have all of the skills to turn requirements into production-ready product. This does reduce pure efficiency and utilisation. There may be times that there is nothing to do but test, so coders and analysts will have to help out. The pay-offs for this compromise include:
* Reduced dependencies and co-ordination required to get a feature into production, increasing predictability.
* Faster feedback cycles: mature teams are able to get from idea to deployed feature in a matter of weeks.
* Increased control and visibility.
* Enhanced risk management for decisions made on requirements and technology.
* Self-organising - There is no project manager assigning tasks to an agile development team. Self-organisation is built on the premise that if no single person is in charge, then the whole team needs to stay vigilant and engaged in their shared goals. Individuals share whole-team accountability for their actions and decisions. If implemented well, this is likely to mean:
* Higher motivation arising from greater autonomy and trust.
* Increased innovation, resulting in new feature ideas and technical solutions.
* Reduced defects as the whole team takes shared accountability for overall outcomes.
* Efficient problem resolution: teams are able to act without escalating up the chain of command and waiting to be told what to do.

**Question 2**

You are leading a development team that was given a task to create a new yoga booking system. High level description of the system is as follows:

• It has a very simple interface to accept user input (bookings) and display classes information

• All bookings, appointments, schedule etc should be stored in a SQL database

• There is ‘backend’ system that should be written in Python to handle the logic and manage the data flow. Your team has two weeks to build a simple prototype that will be shown to the client to seek their feedback and discuss further enhancements

**TASK**

• Break this task into smaller stories (chunks of work) for the team to work on

• Assume that one person works on one task

• Mark tasks that can be worked on in parallel and perhaps those that need to be worked on in particular order

1. Meet with the clients – ask what features they want the booking system to have, make suggestions, take notes on design etc.
2. Draft / wireframe – based on instructions from clients, produce a mock-up of what the booking system will look like and what features it will have.
3. Parallel Tasks:

* UI team – Produce user interface, add some suggested features.
* Backend development team – create SQL database and write Python programs to manage dataflow.

1. Present prototype to client, get feedback.
2. UI and Backend teams to make adjustments and amendments.