**Scrum** is an iterative and incremental [agile software development](https://en.wikipedia.org/wiki/Agile_software_development) framework for managing product development. It defines "a flexible, [holistic](https://en.wikipedia.org/wiki/Holism) product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical [co-location](https://en.wikipedia.org/wiki/Colocation_(business)) or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved.

A key principle of Scrum is the dual recognition that customers will change their minds about what they want or need and that there will be unpredictable challenges—for which a predictive or planned approach is not suited. As such, Scrum adopts an evidence-based [empirical approach](https://en.wikipedia.org/wiki/Empirical_process_(process_control_model))—accepting that the problem cannot be fully understood or defined up front, and instead focusing on how to maximize the team's ability to deliver quickly, to respond to emerging requirements, and to adapt to evolving technologies and changes in market conditions.

Tends to be the case with software development lifecycles, borrowed from elsewhere.

Scrum in relation to software development modeled after an article by Hirotaka Takeuchi and Ikujiro Nonaka (Wise Leadership) published in the Harvard Business Review in 1986.  "The New New Product Development Game" in which they emphasised speed and flexibility for [new product development](https://en.wikipedia.org/wiki/New_product_development). Aimed at manufacturing process, suggested self organizing teams, multi learning and including time for review of process with aim highlight issues and resolve them in order to continually improve the process.

As Wikipedia states,

Scrum is a feedback-driven empirical approach (processes and products are not well defined and generate unpredictable and unrepeatable outputs, relates better to software development than defined process control or rigid set of steps, as we don’t generate the same output every time given a certain input) Empirical approaches are built upon three pillars of:

* transparency (all processes that affect the outcome must be visible and known to everybody involved in the project)
* inspection (carried out frequently enough that unacceptable variances in the process can be detected)
* adaptation (inspector should adjust the process if aspects of the process are in an unacceptable range)

In relation to Scrum, means all work within the Scrum framework should be visible to all members of the team. In order to make these things visible, scrum teams need to frequently inspect the product being developed and how well the team is working. With frequent inspection, the team can spot when their work deviates outside of acceptable limits, identify impediments to the work and adapt their process or the product under development.

These three pillars require trust and openness in the team, which the following five values of Scrum enable:

1. Commitment: Team members individually commit to achieving their team goals, each and every [sprint](https://en.wikipedia.org/wiki/Sprint_(software_development)).
2. Courage: Team members know they have the courage to work through conflict and challenges together so that they can do the right thing.
3. Focus: Team members focus exclusively on their team goals and the sprint backlog; there should be no work done other than through their backlog.
4. Openness: Team members and their stakeholders agree to be transparent about their work and any challenges they face.
5. Respect: Team members respect each other to be technically capable and to work with good intent.

3 roles within scrum:

**Product owner**

Acts as customer representative, accountable for ensuring team delivers value to the business. Responsible for working with stakeholders in creating user stories or use cases, informal description of a feature, normally in format

“As a <role>, I can <action on system> so that <external benefit>”

eg: developing a shopping app, As a customer, I can view a list of special offers on today, so that I can be aware of items on offer and potentially make a purchase.

Spend majority of time liasing with the stakeholders, using user stories will instruct team on what to complete, not how. Provides feedback and guidance to shareholders.

**Scrum master**

Responsible for removing impediments to the team, ensuring they can meet the product goals and create the deliverables.

Help product owner maintain the product backlog, list of things that need to be done within the project.

Help the team determine the definition of “done” for a product that will align with stakeholder requirements.

Promote self organization within the team

Facilitate team events

**Development team**

Responsible for delivering Potentially Shippable Increments at the end of a sprint. Consists of 3 to 9 members who do actual work – analysing, designing, testing,developing, producing documenation.

Self organising and cross functional, with all skills necessary to create the product increment.

<Interaction diagram – Stakeholders deal with the product owner, who in turn deals with the master and workers, who interact with each other.>

Work flow is split into Sprints or iterations, lasting one week to month, usual is two weeks fixed advance.

Events:

**Sprint planning event recommended time 4 hours per two week sprint**

* define the sprint backlog or list of tasks that need completing for the project, taken from Product backlog, identify the work for the current sprint, estimation of commitment for the sprint goal.
* Scope for the sprint
* Whole teams spends first half to select a product backlog item that can be completed within a sprint
* Development team spends second half to prepare a sprint backlog that includes work to complete product backlog. May include splitting or returning product backlog items if they are not achievable within a sprint
* Once these task are completed, the team commits to delvering the tasks, usually by voting – sprint goals confirmed

**Daily scrum**

* Carried out daily, precisely on time even if missing people, should happen at the same time every day, should be limited to max of 15 minutes
* Anyone welcome, though only scrum team members contribute
* Each team member answers three questions:
* What did I complete yesterday that contributed to the team meeting our sprint goal?
* What do I plan to complete today to contribute to the teem meeting our sprint goal?
* Do I see any impediment that could prevent me or the team meeting our sprint goal?
* Any identified impediments (risk, issue, delay, assumption proved wrong) captured by scrum master and shared on team Scrum board. Team member is designated to working towards a resolution.
* No detailed discussions

**Sprint review and retrospective**

* Two events carried out at the end of the sprint:
* In the **review**, review the work that was completed and planned work that was not completed
* Present completed work to the stakeholders – provide a demo.
* Only completed work presented.
* Should last around 2 hours.
* In the **restrospective**, reflect on past sprint
* Identify and agree on continuos improvement actions – what went well? What could improve? What didn’t work so well?7
* Should last around 1.5 hours

**Extensions**

* Producolt backlog refinement – review backlog items possibly breaking up items into more achievable chunks, reviewing priorities or clarifying acceptance criteria
* Cancelling a sprint – carried out by Product owner, if the sprint goal is negated. Start new sprint with sprint planning event eg: Current sprint was to develop an interactor and presenter to deal with the Worldpay payment processing. This is now cancelled and using another system
* Scrum of scrums – lots of independent teams working away, they may be having similar issues or difficultires. Daily scrum appoint an ambassador to participate, the aim to resolve risks and impediments by answering four questions:
* What risks, impediments, dependencies and assumptions has your team moved on since we last met?
* What “” will the team move on before we meet again?
* Are there any new “” slowing down your team and getting in the way?
* Are you about to introduce new “” that may get in another teams way.

Credits:

By Logan Ingalls - Task board, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=19838361