Scrum
an agile development process methodology

- Abhijit Mahajan
- Neelam Agrawal
Scrum is an agile software development methodology.

It is an iterative and incremental methodology for software projects and product- or application-development.

Projects progress via a series of iterations called sprints, which are usually 2-4 weeks long.

A typical scrum team has between five and nine people, but Scrum projects can easily scale into the hundreds.
1993
First Scrum team formed by Jeff Sutherland at Easel Corp.

1996

1998
"Scrum, a pattern language for hyperproductive software development" published by Ken, Jeff, et al.

2001
"Agile Software Development with Scrum" published by Ken Schwaber and Mike Beedle.

"I think the root of both Scrum and Lean is complex adaptive systems theory.
....
I think Scrum and Lean are complementary implementations of ways to deal with physical reality where things are often not linear, not simple, and not predictable."
History

- 1993-Jeff Sutherland, John Scumniotales and Jeff McKenna, came up with an approach at Easel Corporation first to refer it using the single word *Scrum*.

- In 1996, Sutherland and Schwaber jointly presented a paper describing the *Scrum method* at the Business Object Design and Implementation Workshop held as part of OOPSLA ’95 in Austin, Texas.

- 1998- Ken, Jeff, et al came up with “Scrum a pattern language for hyperproductive software development”

- In 2001, Schwaber worked with Mike Beedle to describe the method in the book *Agile with Scrum*
SCRUM Methodology

Planning & System Architecture

Develop

Sprints

Adjust

Review

Wrap

Closure
Scrum & Rugby

The SCRUM methodology shares many characteristics with the sport of Rugby:

- The context is set by playing field (environment) and rugby rules (controls).
- The primary cycle is moving the ball forward.
- Rugby evolved from breaking soccer rules - adapting to the environment.
- The game does not end until environment dictates (business need, competition, functionality, timetable).
Scrum phase list

- **Pregame**
  - Planning
  - System Architecture/High Level Design

- **Game**
  - Sprints (Concurrent Engineering)
  - Develop (Analysis, Design, Develop)
  - Wrap
  - Review
  - Adjust

- **Postgame**
  - Closure
**SCRUN Phases (1)**

- **Pregame**
  - **Planning:**
    - Definition of a new release based on currently known backlog, along with an estimate of its schedule and cost.
    - If a new system is being developed, this phase consists of both conceptualization and analysis.
    - If an existing system is being enhanced, this phase consists of limited analysis.

- **Architecture:**
  - Design how the backlog items will be implemented.
  - This phase includes system architecture modification and high level design.
Game
- Development Sprints:
  - Development of new release functionality, with constant respect to the variables of time, requirements, quality, cost, and competition.
  - Interaction with these variables defines the end of this phase.
  - There are multiple, iterative development sprints, or cycles, that are used to evolve the system.

Postgame
- Closure:
  - Preparation for release, including final documentation, pre-release staged testing, and release.
Development of a comprehensive backlog list.

Definition of the delivery date and functionality of one or more releases.

Selection of the release most appropriate for immediate development.

Mapping of product packets (objects) for backlog items in the selected release.

Definition of project team(s) for the building of the new release.
Assessment of risk and appropriate risk controls.

Review and possible adjustment of backlog items and packets.

Validation or reselection of development tools and infrastructure.

Estimation of release cost, including development, collateral material, marketing, training, and rollout.

Verification of management approval and funding.
Review assigned backlog items.

Identify changes necessary to implement backlog items.

Perform domain analysis to the extent required to build, enhance, or update the domain models to reflect the new system context and requirements.

Refine the system architecture to support the new context and requirements.
Identify any problems or issues in developing or implementing the changes

Design review meeting, each team presenting approach and changes to implement each backlog item.

Reassign changes as required.
A sprint is the basic unit in Scrum
  lasts between one week and one month

Each sprint is preceded by a meeting
  where the tasks for the sprint are initiated for the sprint goal

The work items come from the product *backlog*
  which is a prioritized list of requirements

During the sprint planning meeting, the Product Owner informs the group of the items in the product backlog that needs to be completed
  the ones with the highest priority
The group then determines how much of this they can commit to complete during the next sprint and records this in the sprint backlog.

During a sprint, no one is allowed to change the sprint backlog, which means that the requirements are frozen for that sprint. If requirements are not completed for any reason, they are left out and returned to the product backlog.
More Game phase (1)

- **Develop:**
  - Defining changes needed for
    - the implementation of backlog requirements into packets, opening the packets, performing domain analysis
    - designing, developing, implementing, testing, and documenting the changes.
  - Development consists of the micro process of discovery, invention, and implementation.

- **Wrap:**
  - Closing the packets, creating an executable version of changes and how they implement backlog requirements.
More Game phase (2)

- Review:
  - All teams meeting to present
    - work and review progress
    - raising and resolving issues and problems
    - adding new backlog items
  - Risk is reviewed and appropriate responses defined.

- Adjust:
  - Consolidating the information gathered from the review meeting into affected packets, including different look and feel and new properties.
When the management team feels that the variables of time, competition, requirements, cost, and quality concur for a new release to occur, they declare the release “closed” and enter this phase.

This phase prepares the developed product for general release.

Integration, system test, user documentation, training material preparation, and marketing material preparation are among closure tasks.
The various roles in Scrum team

- **Core Roles**
  - Product Owner
  - Development Team
  - Scrum Master

- **Ancillary roles**
  - Stakeholders (customers, vendors)
  - Managers
The Product Owner represents the voice of the customer

- is accountable for ensuring that the Group delivers value to the business

- Writes customer-centric items (user stories),
  - prioritizes them
  - and adds them to the product backlog

- Scrum groups should have one Product Owner
  - She may also be a member of the Management Group
  - It is recommended that this role not be combined with ScrumMaster
Development Team

- Responsible for delivering potentially shippable product increments at the end of each Sprint.
- It is made up of people with cross-functional skills who do the actual work
  - analyze, design, develop, test, technical communication, document, etc
- It is self-organizing
  - even though they may interface with project management organizations (PMOs).
Scrum Master

- Scrum Master is accountable for removing impediments to the ability of the group to deliver the sprint goal/deliverables.
- She acts as a buffer between the group and any distracting influences.
- The Scrum Master is the enforcer of rules.
- A key part of the role is to protect the Development Team and keep it focused on the tasks at hand.
The ancillary roles in Scrum groups are those with no formal role and infrequent involvement in the Scrum process but nonetheless, must be taken into account.

Stakeholders (customers, vendors):
- People who enable the project and for whom the project produces the agreed-upon benefit[s]
  - that justify its production
- They are only directly involved in the process during the sprint reviews

Managers:
- People who control the environment
Meetings Overview
The following is a list of meetings in Scrum development

- Daily Scrum
- Backlog grooming: storytime
- Scrum of Scrums
- Sprint planning meeting
- Sprint review meeting
- Sprint retrospective
Happens each day during the sprint is a project status meeting

This meeting has specific guidelines:

- The meeting starts precisely on time
- All are welcome, but normally only the core roles speak
- The meeting length is set to 15 mins
- The meeting should happen at the same location and same time every day
During the meeting, each group member answers three questions

- What have you done since yesterday?
- What are you planning to do today?
- Any impediments/stumbling blocks?

Scrum Master should facilitate resolution of these impediments, although the resolution should occur outside the Daily Scrum itself to keep it under 15 minutes.
Backlog grooming: storytime

- This is the process of
  - estimating the existing backlog using effort/points
  - refining the acceptance criteria for individual stories
  - and breaking larger stories into smaller stories

- Meetings should not be longer than an hour

- Meeting does not include breaking stories into tasks

- Group can decide how many meetings are needed per week.
Scrum of scrums

- Each day normally after the Daily Scrum.
- These meetings allow clusters of groups to discuss their work, focusing especially on areas of overlap and integration.
- A designated person from each group attends.
- The agenda will be the same as the Daily Scrum, plus the following four questions:
  - What has your group done since we last met?
  - What will your group do before we meet again?
  - Is anything slowing your group down or getting in their way?
  - Are you about to put something in another group’s way?
Sprint planning meeting (1)

- Takes place at the beginning of the sprint cycle
- The Sprint Planning Meeting is attended by:
  - the product owner
  - Scrum Master
  - the entire Scrum Team.
- There are two defined artifacts resulting from this meeting:
  - A sprint goal
  - A sprint backlog
- A sprint goal is a short, one- or two-sentence, description of what the team plans to achieve during the sprint.
  - It is written collaboratively by the team and the product owner.
The team asks enough questions so that they can turn a high-level user story of the product backlog into the more detailed tasks of the sprint backlog.

Sprint Backlog is prepared with details of the time it will take to do a particular work.

Identify and communicate how much of the work is likely to be done during the current sprint.

Eight hour time limit:
- (1st four hours) Product Owner + Group: dialog for prioritizing the Product Backlog
- (2nd four hours) Group only: hashing out a plan for the Sprint, resulting in the Sprint Backlog
Sprint review meeting (1)

- Held at the end of each sprint during which the Scrum team shows what they accomplished during the sprint.
- Typically this takes the form of a demo of the new features.
- It is intentionally kept very informal, typically with rules forbidding the use of PowerPoint slides.
- A sprint review meeting should not become a distraction or significant detour for the team
  - rather, it should be a natural result of the sprint
Participants in the sprint review typically include:
- the Product Owner
- the Scrum team
- the ScrumMaster
- management, customers, and developers.

The project is assessed against the sprint goal determined during the Sprint planning meeting.

It is important that the team has achieved the overall goal of the sprint.
Sprint Review Meeting (3)
The sprint retrospective is usually the last thing done in a sprint.

Many teams do it immediately after the sprint review.

This is a period at the end of each sprint to deliberately reflect on how the team is doing and to find ways to improve.

The entire team, including both the ScrumMaster and the product owner participate in this meeting.

It has three hour time limit

Two main questions asked in the sprint retrospective are:

What went well during the sprint?

What could be improved in the next sprint?
Artifacts

- Product Backlog
- Sprint Backlog
- Burndown Charts
Product Backlog(1)

- It is an ordered list of "requirements" maintained for a product.

- These items are ordered by the Product Owner based on considerations like risk, business value, dependencies, date needed, etc.

- The values in product backlog are often stated in story points using a rounded Fibonacci sequence.
Those estimates help the Product Owner to gauge the timeline and may influence ordering of backlog items.

The Product Backlog, and business value of each listed item is the responsibility of the Product Owner.

The estimated effort to complete each backlog item is determined by the Development Team.
As a <stakeholder>
I want <what>
so that <why>

As a buyer
I want to save my shopping cart
so that I can continue shopping later

Default Definition of Done
- Acceptance tested
- Release notes written
- Releasable
- No increased technical debt

As a booker
I want to receive notifications when
new available slots appear in the calendar
so that I don’t have to keep checking manually

(... etc ...)

= I haven’t messed up the codebase
Sprint Backlog

- It is the list of work the Development Team must address during the next sprint.
- The list is derived by selecting stories/features from the top of the product backlog.
- The Development Team should keep in mind the velocity of its previous Sprints when selecting stories/features for the new sprint.
The stories/features are broken down into tasks by the Development Team

should normally be between four and sixteen hours of work

Tasks on the sprint backlog are never assigned;

rather, tasks are signed up for by the group members as needed during the daily scrub.

Often an accompanying task board is used to see and change the state of the tasks of the current sprint,

like “not checked out”, “checked out” and “done”.
Burndown Charts

- **Sprint burndown chart**
  - It is a publicly displayed chart (updated daily) showing remaining work in the sprint backlog.
  - It gives a simple view of the sprint progress.

- **Release burndown chart**
  - shows the amount of work left to complete the target commitment for a Product Release

- **Alternative release burndown chart**
  - which basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline.
Burndown Charts
Scrum-ban is a production model based on Scrum and Kanban.

It is suited for maintenance projects or (system) projects with frequent and unexpected user stories or programming errors.

In such cases the time-limited sprints of the Scrum model are of no appreciable use, but Scrum’s daily meetings and other practices can be applied.

Visualization of the work stages and limitations for simultaneous unfinished user stories and defects are familiar from the Kanban model.
Using these methods, the group’s workflow is directed in a way
- that allows for minimum completion time for each user story or programming error,
- and on the other hand ensures each group member is constantly employed.

The major differences between Scrum and Kanban are
- in Scrum, work is divided into sprints that last a certain amount of time
- whereas in Kanban the workflow is continuous.
Advantages (1)

- The SCRUM methodology is designed to be quite flexible throughout.
- It provides control mechanisms for planning a product release and then managing variables as the project progresses.
- This enables organizations to change the project and deliverables at any point in time, delivering the most appropriate release.
- The SCRUM methodology frees developers to devise the most ingenious solutions throughout the project, as learning occurs and the environment changes.
- Small, collaborative teams of developers are able to share tacit knowledge about development processes.
Advantages (2)

- Object Oriented technology provides the basis for the SCRUM methodology.
- Objects, or product features, offer a discrete and manageable environment.
- Procedural code, with its many and intertwined interfaces, is inappropriate for the SCRUM methodology.
- SCRUM may be selectively applied to procedural systems with clean interfaces and strong data orientation.
Some difficulties

Scrum works... but not alone

Scrum is simple... but hard

Scrum can be painful! waste is ruthlessly exposed

Scrum is different! old habits die hard

The old tool was better!
Get set go!

1. Product vision
2. Product owner
3. Co-locate everyone.
4. GO! Start first sprint.
Ken Schwaber. “SCRUM Development Process”
