Scrum Framework

Empiricism

- Empiricism says knowledge comes from experience and
- Making decisions based on what is known.

Transparency

- Everyone has a common understanding of what is happening.
- By sharing updated information – Scrum teams create transparency.
- Transparency is with respect to:
    - Product Backlog – which lets anyone know what are the requirements that are part of this Project or Product – which evolve dynamically.
        - Scrum Teams use Backlog refinement (an activity) to work on future Backlog Items – adding more detail, understanding them better, coming up with what is the work needed to deliver them, if possible, doing an estimation etc.
    - Sprint backlog – owned by Developers – the work that Developers will be doing in the Sprint.
        - Developers select which backlog items they can work in the Sprint from Product Backlog based on the order (or priority) set by the Product Owner
        - Sprint Backlog evolves as the Sprint progresses – at any time, backlog items can be added to it by Developers, more tasks can be added as they discover more.
        - Sprint Backlog should be used in Daily Scrum – every 24 hours Developers plan what they are going to do to get closer to Sprint Goal
    - Increment – what is delivered in this Sprint + whatever was delivered earlier
        - Definition of Done – a checklist that helps Scrum Teams to know what all the work they have to do to produce a usable increment. This includes various levels of testing, documentation, focus on Non-Functional Requirements (security, performance, scalability etc.)

Inspection

- Frequently checking what is happening
- Each Scrum Event is an opportunity to inspect and adapt.
- Never skip any Scrum event.

Adaptation

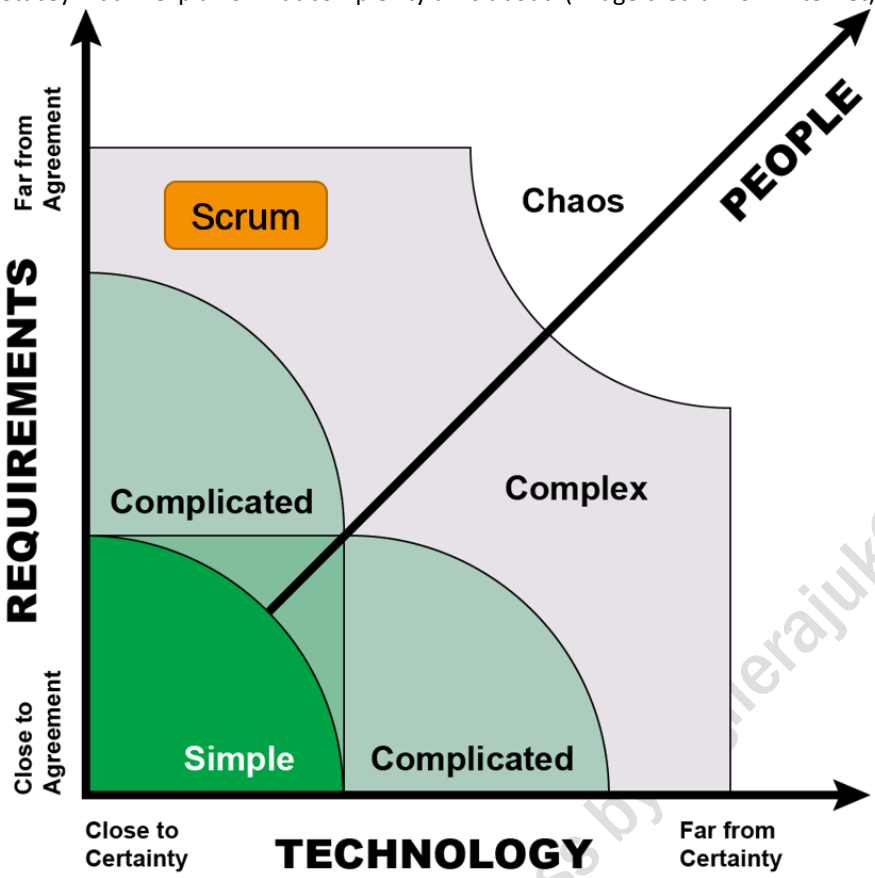
- Adjusting the work (project) so it is back on track.
- For ex:
    - In Daily Scrum, Developers discover that one particular backlog item has not received Security approval. Without this approval, the Increment is not Done. There is a possibility that this approval may come after the Sprint.
    - They negotiate with PO on what to do:
        - Can they carry this item to Next Sprint and complete it?
        - Can they work on some other Backlog Item in this Sprint?
    - This must be updated in Sprint Backlog for transparency.

Complex Projects/Product Development

- For you to understand Scrum better – consider Product mindset instead of Project mindset.

Project Mindset	Product Mindset
• **Project is temporary with an end date.**	• If the Product has demand, it will be there in the market.
• **Project will complete by some time, then to maintain or support the deliverable, support projects are launched.**	• If Product has demand, a Scrum team works on it to enhance it, support it, maintain it etc.

- Stacey matrix explains what complexity all is about: (Image credit: from Internet)



Core elements of Scrum

Accountabilities or Roles	Events	Artifacts	Rules and others

• **SM**
• **PO**
• **Developers**
• **Scrum Team includes SM, PO, and Dev**
• **Size of Scrum Team: 10 or fewer**
	1. Sprint
	2. Sprint Planning
	3. Daily Scrum
	4. Sprint Review
	5. Sprint Retro
		• Product Backlog
		  ○ Product Goal commitment
		• Sprint Backlog
		  ○ Sprint Goal commitment
		• Increment
		  ○ Definition of Done commitment
			• Empiricism
			• 5 Scrum Values
			• Timeboxes
			• Self-management
			• Cross functional team

Differences in ways of working b/n Non-Scrum & Scrum

	Non-Scrum	In Scrum
Project	A temporary endeavor/work with an end date and an objective to meet	Uses Product concept. As long as the Product exists, Scrum Teams would continue to support/enhance/maintain it as needed.
Project Team	Team members who work on producing Project Output	Scrum Team – PO + Dev + SM only
Team members	Usually called by skill, role (BA, QA, SW Engr, Sr SW Engr, Consultant, System Admin, DBA etc.)	In Scrum – anyone who is developing value and part of team is called Developer. So even BA, QA etc. will be called Developers.

Project Sponsor	Who invests in the Project, who brings funding and budget to the project	This is PO's responsibility.
Project Manager	A dedicated role focused on managing the Project	Each role does its own management – there is no need for a dedicated Project Manager
Stakeholder Engagement	Primarily done by PM	Should be done by PO, but Developers can interact with Stakeholders as needed (for ex: understanding a requirement, showing a demo, clarifying on a bug etc.)

Scrum Master

- Ensure that the team follows Scrum as defined in Scrum Guide.
- SM serves the Scrum Team:

  The Scrum Master serves the Scrum Team in several ways, including:

  - Coaching the team members in self-management and cross-functionality;
  - Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
  - Causing the removal of impediments to the Scrum Team's progress; and,
  - Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
-
- SM serves the PO:

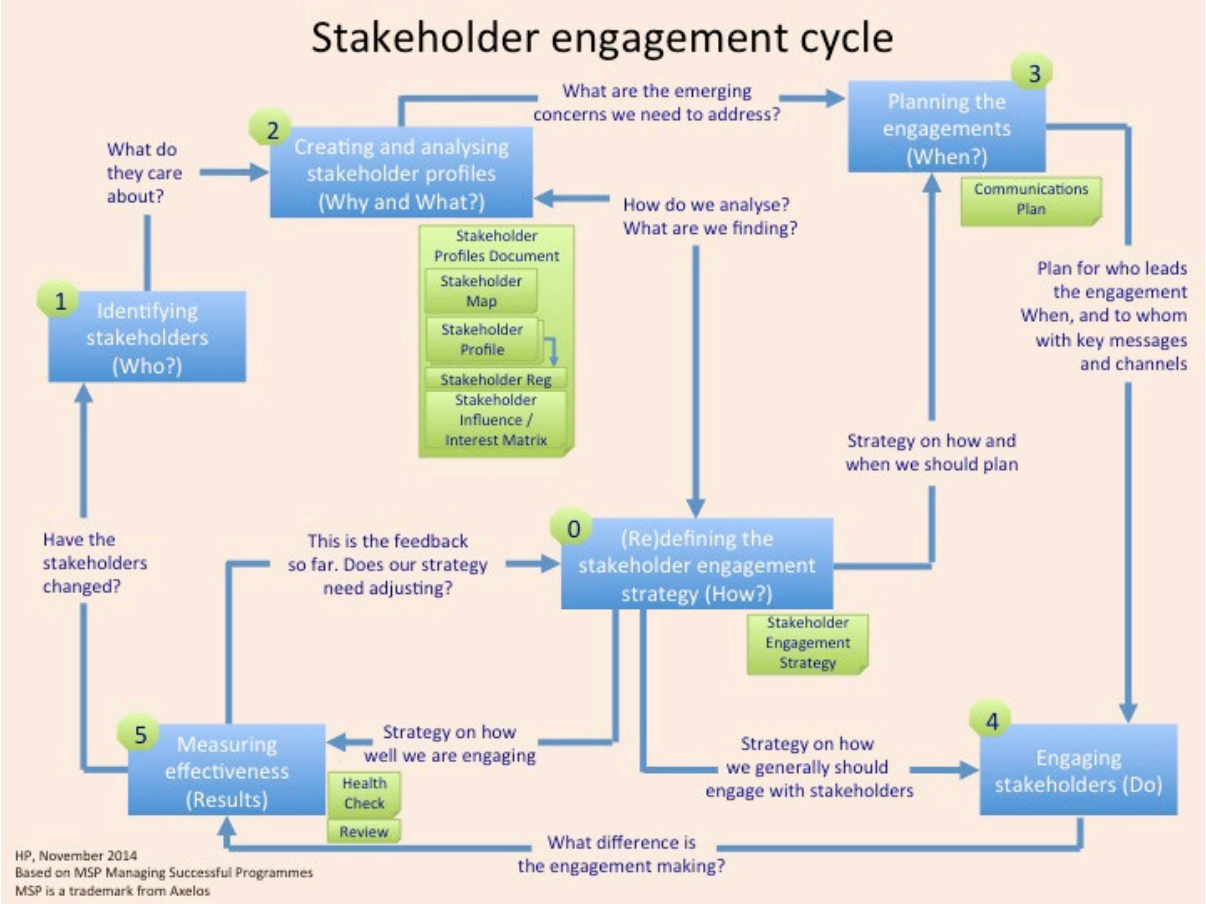
  - Helping find techniques for effective Product Goal definition and Product Backlog management;
  - Helping the Scrum Team understand the need for clear and concise Product Backlog items;
  - Helping establish empirical product planning for a complex environment; and,
  - Facilitating stakeholder collaboration as requested or needed.

- SM serves the Org (outside the Scrum team):

  - Leading, training, and coaching the organization in its Scrum adoption;
  - Planning and advising Scrum implementations within the organization;
  - Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
  - Removing barriers between stakeholders and Scrum Teams.

Stakeholder Engagement

- SM can coach the PO about Stakeholder engagement following this example approach:



Certain things that SM should avoid:

- Assigning backlog items to team members (because Developers pick the backlog items themselves)
- Providing solutions to Developers
    - First SM should allow Developers to explore on their own (because they are self-managing, they should try first from their end)
    - If it doesn't work then SM can intervene, help them to find solution and support it
- SM is not a Project Manager.
    - Each role in Scrum does its own management.
        - PO → Scope management, Cost management, Stakeholder engagement
        - Dev → Work management (who is going to work on what – they will decide)
        - PO + Dev → Quality Management
        - SM → w.r.t. to Scrum: Process Management, Communications Management

Product Owner

- Like your Project Sponsor, Business SPOCs (Single point of Contacts who takes necessary decisions regarding project/module/functionality etc.), Application or Service Owner (from customer who will own an entire application like SAP MM, IT service Desk, CRM Application etc.)

Primary Responsibilities

- Requirements management
    - Ensuring that requirements are stored in Product Backlog, available to Developers with required details
    - Ordering the requirements – which one should be done first
        - Few techniques
            - Value

        - Ex: For a subscription model, if we rollout this enhancement, how much value we are delivering to our customers?
            - If each subscription is $10, if we get 100 subscribers, value is $1000
    - Risk
        - Which is high risk item – given higher priority
        - Ex: there is a regulation or law which we must meet. So, it is given higher priority. For ex: GDPR implementation.
    - Time criticality
        - Ex: Valentine's day is coming – so those deals are more important at this point of time (written on 11-Jan)
- Connecting Dev to required Stakeholders
    - Ex: for arranging Job shadowing so developers know how the work is done, helping them in testing etc.

- Monitoring the progress towards Product Goal
    - Product Goal – end objectives that we are trying to achieve.
        - Template: https://www.romanpichler.com/blog/product-goals-in-scrum/

	1st quarter	2nd quarter	3rd quarter
DATE			
NAME	MVP	Version 2	Version 3
PRODUCT GOAL	Help the users to understand their eating habits and acquire an initial user base.	Help the users improve their eating habits and grow the user base.	Help the users get fitter and generate revenue in form of in-app purchases.
FEATURES	• Healthy eating dashboard • Integration with smart watches and fitness devices	• Recipes and shopping lists • Social media integration	• Personalised advice (coach) • Weight loss • Integration with scales
METRICS	Be in the top 15 diabetes apps six weeks after launch	Activations, engagement, NPS	Revenue from in-app purchases two weeks after release

    - In Sprint Planning for each Sprint – PO collaborates with Dev and SM to create a Sprint Goal [Important: **Sprint Goal is created by all 3 roles- PO + SM + Dev**]
        - Product Goal is product objective whereas Sprint Goal is objective for each Sprint. Sprint Goal is a steppingstone towards Product Goal
    - In Sprint Review, provides updates about Budget, what is happening in the market, customer feedback, CSAT scores etc.
- Product Backlog management
    - Developing and explicitly communicating the Product Goal;
    - Creating and clearly communicating Product Backlog items;
    - Ordering Product Backlog items; and,
    - Ensuring that the Product Backlog is transparent, visible and understood.

PO and BA Differences

Product owner	Business Analyst
• Key Scrum accountability & Role • Accountable for Delivering (or maximizing value)	• Part of Developers • Accountable for documenting requirements, helping Developers understand requirements, may do functional testing

• Typically comes from Business (i.e., your customer like Business Owner, Application owner, SPOC etc.)
• Main activities:
  ○ Coming up with Product vision and product goal
  ○ Getting budget for the Project
  ○ Engaging with Stakeholders for their requirements, sharing them Project progress, inviting them for Sprint Review etc.
  ○ Ordering Product Backlog
  ○ Communicating backlog items clearly to Developers
  ○ Deciding on Release
  ○ Sign off on requirements
	• Typically, may come from offshore supplier teams (from Customer they can be known as SMEs)
	• Main activities:
	  ○ Documenting scope
	  ○ Creating User stories (one example)
	  ○ Connecting Developers with Stakeholders
	  ○ Doing functional testing
	  ○ Help the PO to come up with Product Vision and Goal
	  ○ Facilitate Customer Journey mapping etc. to understand requirements better

• Common things between PO and BA:
  ○ Create User stories (anyone can create user story)
  ○ Connect Devs to Stakeholders
  ○ Doing functional testing
  ○ Help the Devs to understand the requirements better	

Developers

- A group of people – team – that has all the skills required to deliver Project output (Product)
- Cross functional team – as a team they have all the skills required
- Self-managing – developers decide how to work, no task assignment by someone
- No titles – Testers, Business Analysts, Team leads etc. are not used.
- No sub-teams. For example, a dedicated QA Team which does only testing is an **anti-pattern**.

Scrum Events

Example 2-week Sprint cycle – events

	Mon	Tue	Wed	Thu	Fri
Sprint N	**Sprint Planning**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**
	Daily Scrum	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum** **Sprint Review** **Sprint Retro**
Sprint N + 1	**Sprint Planning**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**
	Daily Scrum	**Daily Scrum**	**Daily Scrum**	**Daily Scrum**	**Daily Scrum** **Sprint Review** **Sprint Retro**

Sprint

- Container in which all the remaining 4 events take place.
- Sprints always run back-to-back i.e., no gaps in between Sprints.
  "Time, tide and Sprints wait for none"
- Each Sprint must produce some value. The value need not be in production (when to release PO takes final call, but Scrum Team should make sure that each Sprint they have something that is ready to be used)
- Running dedicated Sprints like these is **NOT** Scrum
  - **Sprint 0** – project kickoff only for onboarding, requirement gathering, initial design, architecture and infra setup, any procurements if required etc.
  - **Hardening Sprint** – only for integration and testing

- **Release Sprint** – only for release
- Do not extend the Sprint beyond timebox.

Factors influencing Sprint Length

- Maximum duration of a Sprint is 1 month.
- The amount of risk acceptable to the Product Owner
- The team's ability to produce an usable increment
- To be in sync with Business requirements (for ex: release schedules)

Sprint Planning

	Participants	Inputs	Event	Outputs
Sprint Planning	**Scrum Team (all 3 roles) mandatory** **If needed, technical architects, SMEs**	1. **Developers Capacity** 2. **Ordered product backlog** 3. **Team's past performance (Velocity or throughput)** 4. **Improvement items from prev. Sprint Retro** 5. **DoD**	• **PO shares WHY this Sprint is valuable (PO can share Product Goal)** • **WHAT - All 3 roles create Sprint Goal** • **HOW – ONLY Developers plan how to work**	• **Sprint Backlog (Backlog items + Tasks)** • **Sprint Goal (All 3 roles)**

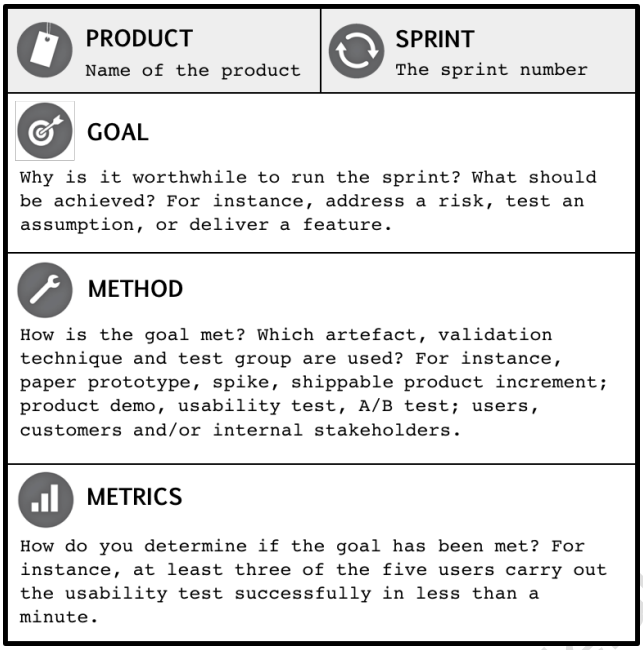
Sprint Goal

- A commitment towards Sprint Backlog
- It answers WHY the Sprint is valuable and WHAT the team is trying to achieve.
- Crafted by all 3 roles, doesn't change during the Sprint
- If due to any reason, the team can't meet the Goal:
  - In Retro – the Scrum Team should analyze why they were not able to meet the goal
  - What they can do to achieve it in next Sprint
- If the Goal becomes obsolete (useless) only PO can decide to cancel the Sprint.
  - Ex: in one of a Project, the Scrum Team was trying to incorporate Bitcoin and Digital currency payments into their e-commerce platform. During the Sprint, Local Government banned the digital currency. Now this Sprint Goal is obsolete. PO decides to cancel the Sprint and start a new one with a revised goal.

How to craft an effective Sprint goal:

- Sprint Goal doesn't reflect all Sprint Backlog items – it should reflect the most important ones selected for the Sprint that must be delivered. PO can use MoSCoW Prioritization (Must have, Should have, Could have [nice to have], Won't have [now, may be later])
- Sprint Goal must focus on functionality but not individual backlog items.
  - Ex – bad Sprint Goals – In Sprint 5, deliver PBI# 89, 76, 54. Fix Defects# 8, 90. Instead of such goal, arrive at this one – In Sprint 5, we deliver Chat Bot functionality for Service Request module enabling self-service with AI and ML.
- Sprint Goal must be used during the Sprint:
  - Daily Scrum – use to check if it can be achieved or not
  - Sprint Review – check if it is met or not
  - Sprint Retro – how to meet it if we have not met

- Sprint Goal – template https://www.romanpichler.com/blog/sprint-goal-template/



Daily Scrum

- For Developers to assess their progress towards Sprint Goal
- Plan for next 24 hours

	Participants	Inputs	Event	Outputs
Daily Scrum	**ONLY for Developers** **If SM and/PO are also doing some Development, they can attend. Otherwise, NO.**	1. Sprint Backlog (updated, transparent) 2. Impediment's tracker 3. Sprint Goal	• Developers assess their progress towards Sprint Goal • Earlier version of Scrum Guide there were 3 questions. • SM should ensure that team has this event every day, it is completed in 15 minutes	• Updates to Sprint Backlog • Updates on Impediments
	• Any discussions, trying to find solutions for problems etc. should be done after Daily Scrum – NOT during Daily Scrum. • Daily Scrum is NOT a Status meeting. • Allowing outsiders of team (Architect, Managers etc.) is an anti-pattern.			

Doing effective Daily Scrum

- Do it in front of Sprint Backlog or Scrum Board
  - Physical co-located team – they do in front of the Board or share the screen
  - For virtual – do it with screen sharing or all Dev looking at their Backlog on their machines
- Keep it only for updates on Sprint Backlog items – no problem solving, no discussions etc. These can be had after 15 minutes.
- Developers can look at the trend of Sprint Backlog items progress and discuss.

- Example – trend is shown in () with + or -. Example Illustration from a project

Squad	On hold	Start	Closed	Dev	Story Detailing	Estimation	Ready for UAT	Total
Names are masked	0	0(-3)	0	7(+3)	0	0	0	7
	2(+1)	0(-2)	1	5(+1)	0	0	0	8
	0	3	0	3	1	0	0	7
	2	6(-2)	0	8(+2)	0	0	1	17
	0	0	0	9	0	0	0	9
	0	4	0	1	0	0	0	5
	0	0	1	5	0	3	0	9
Total	4(+1)	13(-7)	2	38(+6)	1	3	1	62

- Use CFD (Cumulative Flow Diagram)

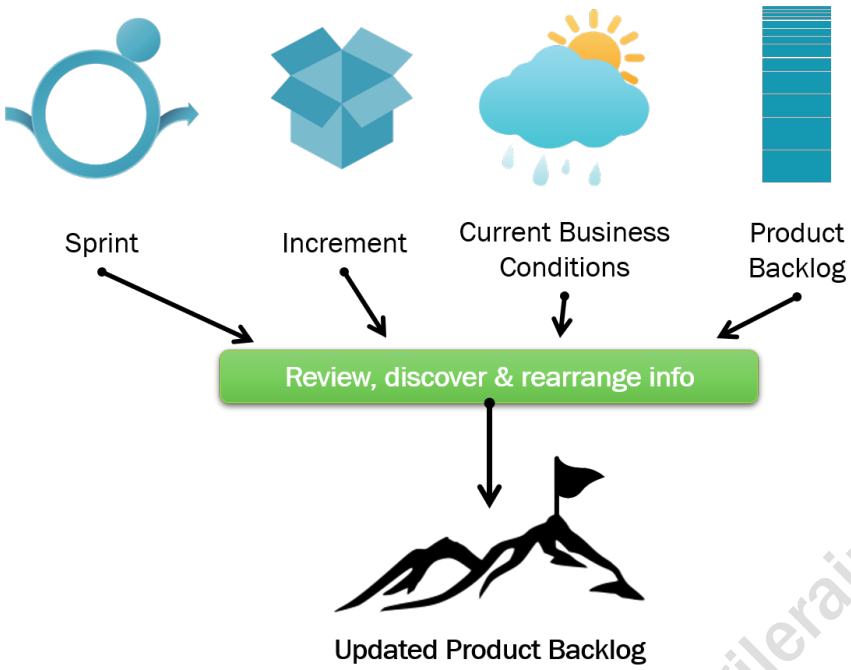
- Daily Scrum without 3 questions: https://www.youtube.com/watch?v=VhOPIWmRU0g
- Another visualization for Daily Scrum: https://www.amazon.com/Visualization-Examples-Jimmy-Janl%C3%A9n/dp/9188063011
- Work Item age – (flow – Kanban) metric tells you how many days these backlog items are with teams. The longer it is, the longer team is spending time with them. In Daily Scrum, WIA can be used to know which item they can focus today.

Sprint Review

- For Scrum Team and Stakeholders (invited by PO) to review the increment, arrive at what to do next

	Participants	Inputs	Event	Outputs
Sprint Review	**Scrum Team and Stakeholders.**	**1. Current Increment - Deliverable 2. Market Conditions by PO 3. Customer feedback by PO 4. Latest Product Backlog**	**• Stakeholders and Scrum Team (all 3 roles) collaborate on what was done and what to do next**	**• Updates to Product Backlog**
	• Sprint Review is more than a Demo • An event where outside team members can participate (Stakeholders are invited by PO) • PO can talk about market conditions, customer feedback, budget etc.			

A Graphical way to look at Review: (Image courtesy: Scrum.org)



Example Agenda – 3-part meeting

Product Discussion Points	Work Discussion Points	Feedback Opportunities
• The goal of this Sprint • Progress towards the Product Goal • What was done • State of the Product Backlog • Projections of likely release targets	• The actual Increment of product • What happened in the Sprint • How problems were addressed and the effect on the Increment	• From everyone! • Questions from Scrum Team for stakeholders • Thoughts on Increment from stakeholders

Sprint Retrospective

- For the Scrum Team (ONLY all 3 roles) to evaluate how did they perform as a team in last Sprint

	Participants	Inputs	Event	Outputs
Sprint Retro	**ONLY for Scrum Team (3 roles)**	1. Current process 2. Working agreements 3. DoD 4. Sprint Backlog and Goal	• All 3 roles to see how they can improve • How they can do better in next Sprints	• Identified improvement items for future sprints
	• No one outside Scrum team is allowed here • Several techniques exist to do Retro.			

Handling incomplete Backlog Items in a Sprint

- First Developers re-estimate them for remaining work
- They are presented with PO – PO decides
    - Whether to continue these incomplete items in next Sprint
        - If yes, they are added to next Sprint Backlog (Spillover items)
    - They are not of priority now, they can be completed later
        - Added to Product Backlog

All events at a glance

Event	Inspection	Adaptation	Who Attends	Timebox for 1 Month
Sprint Planning	Product Backlog, Product Goal	Sprint Backlog, Sprint Goal	Scrum Team	8 hours
Daily Scrum	Progress toward Sprint Goal	Sprint Backlog	Developers	15 minutes (always)
Sprint Review	Increment, Sprint, Product Backlog, Progress toward Product Goal	Product Backlog	Scrum Team Stakeholders	4 hours
Sprint Retrospective	Sprint, Definition of Done	Actionable improvements	Scrum Team	3 hours

- TIMEBOX shown here is for 1 month. For shorter Sprints, they are shorter.

Timebox

- Maximum amount of time an event can take place
- Do not exceed this time
- All event's timebox is shown next

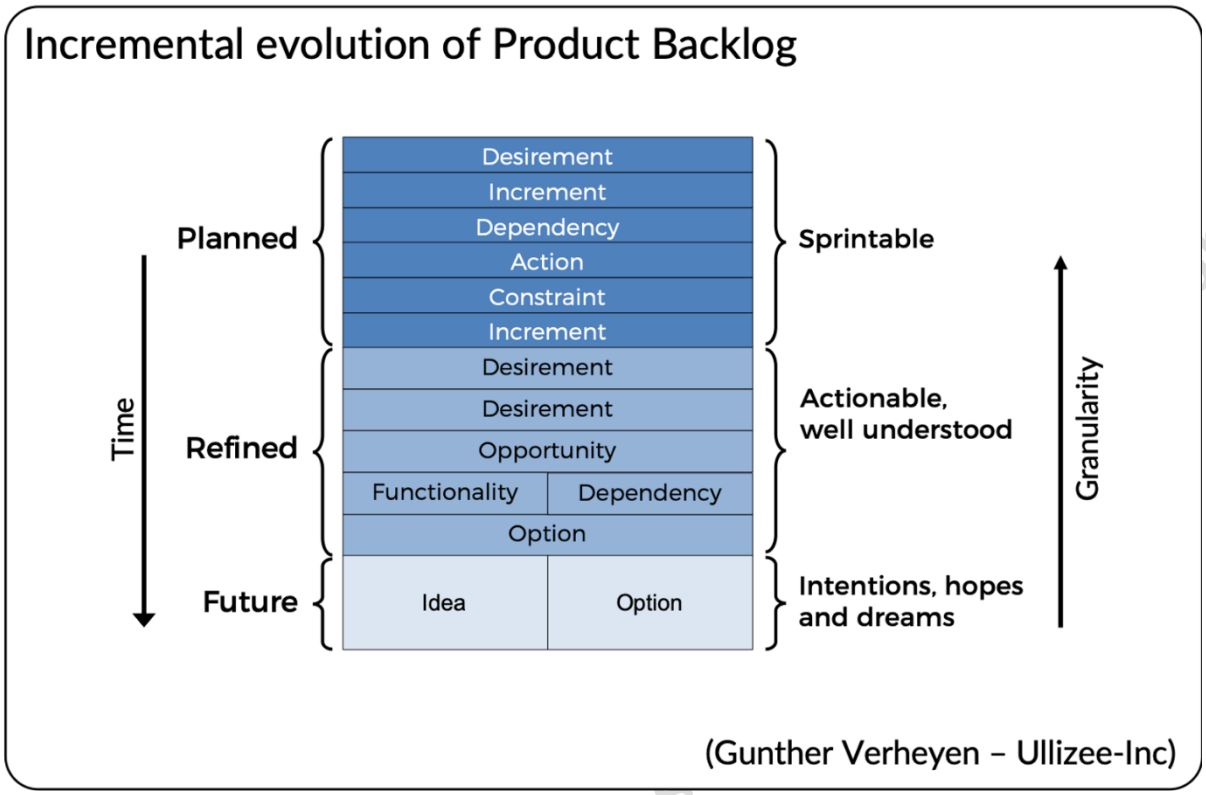
Event	One Month	3 Weeks	2 Weeks	1 Week
Sprint Planning	8 hours	*Less than 8 hours* (~6 hours)	*Less than 8 hours* (~4 hours)	*Less than 8 hours* (~2 hours)
Daily Scrum	15 minutes			
Sprint Review	4 hours	*Less than 4 hours* (~3 hours)	*Less than 4 hours* (~2 hours)	*Less than 4 hours* (~1 hour)
Sprint Retrospective	3 hours	*Less than 3 hours* (~2 hours 15 mins)	*Less than 3 hours* (~1 hour 30 mins)	*Less than 3 hours* (~45 mins)

Artifacts

Product Backlog

- Contains complete scope i.e., requirements of the Product or Project
- Enhancements, Change Requests, Defects etc. can be part of Product Backlog
- Owned by PO, ordered by PO, anyone can add Product Backlog items (for Sprint Backlog, only Developers can add)
- Items on top are high ordered when compared to lower-level items
- Items on the top should have enough details so Developers can work on them
- In traditional Project Management: BRD, SRS, Project Scope Statement of PMP®, Project Product Description of PRINCE2® All are equal to Product Backlog

- Product Backlog evolves as we progress. There is no need to do complete Requirement gathering at the beginning of the Project/Product Development itself in Scrum.

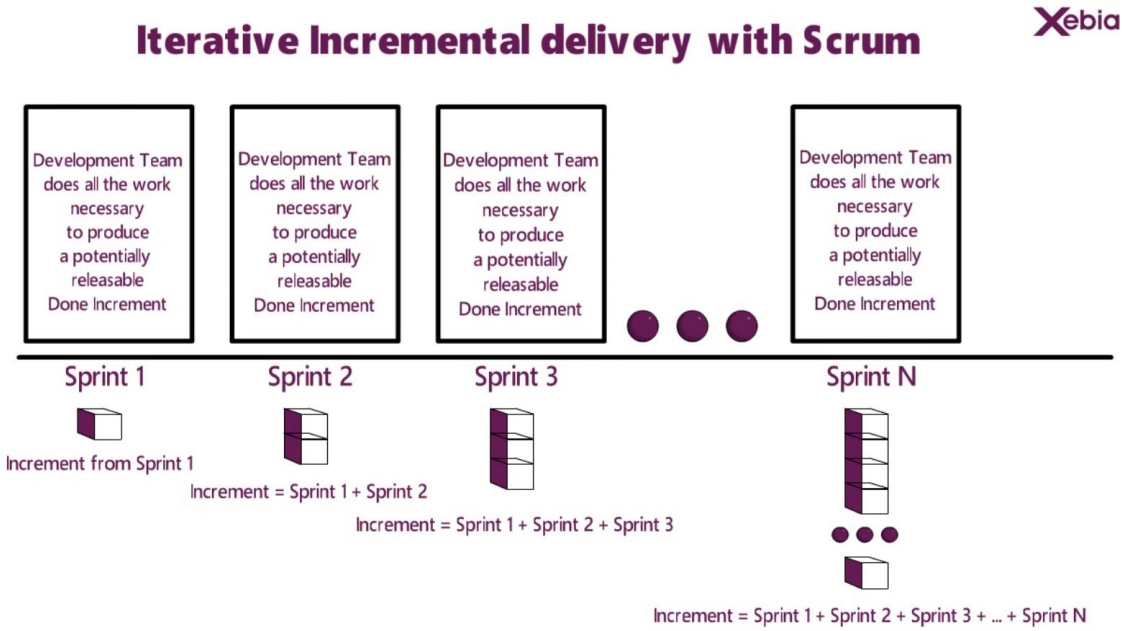


Sprint Backlog

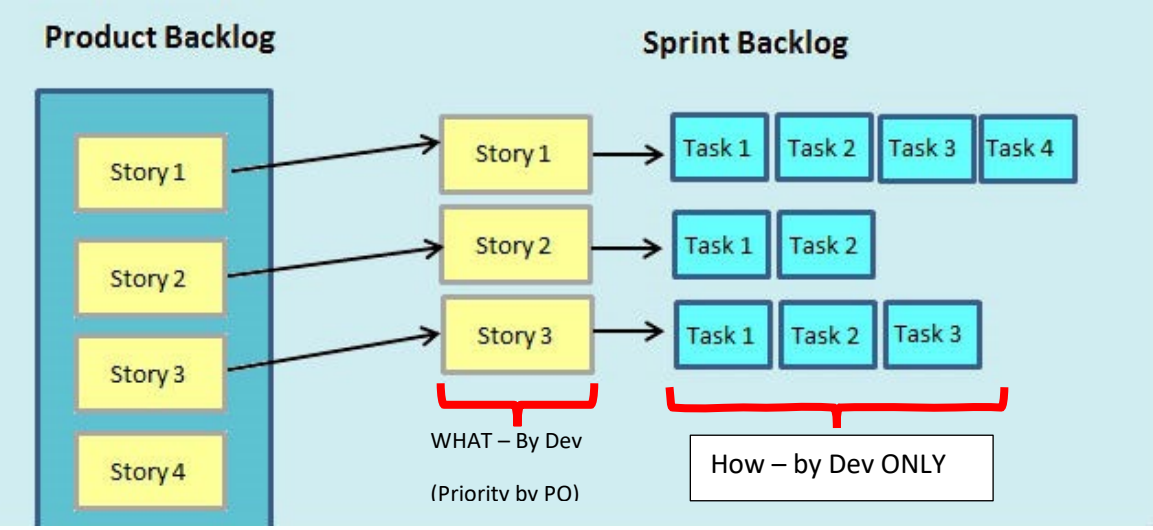
- Selected Backlog Items (**and may include Tasks)** – a plan for Developers how to work in the Sprint
- Only Developers can add, move content – even PO, SM, Stakeholders etc. (for example even a CEO) – approaches the Developers to add something into Sprint Backlog
- For scope changes during the Sprint, Developers will negotiate with PO

Increment

- Deliverable from Sprint which is well integrated with all earlier increments.



Product Backlog and Sprint Backlog comparison



Other elements

5 Scrum Values CCFOR

- Courage
- Commitment
- Focus
- Openness
- Respect
  - All these are behaviors

3 Commitments

- Product Goal
- Sprint Goal
- Definition of Done

Initial Sprints for a Project as per Scrum Guide

- Initial Sprint – Scrum Team selects few backlog items that can help them to demonstrate usable value i.e., increment along with working on necessary architecture and design
- In each Sprint, you should balance Value Delivery with Infrastructure etc. work

Handling architecture and Infra structure work in each Sprint

- XP (eXtreme Programming – another agile method) Practice of Spike – Developers dedicate some time in Sprint (or more than one Sprint) to learn, experiment, demonstrate new technology or tool etc.

- Example: your team wants to demonstrate Chatbot to customer. So, your team wants to build a PoC or Prototype to show this. Team will dedicate some time each Sprint for this and work on this. This is called Spike.

- You can create backlog items or Stories for all this architecture and infra work. Scaled Agile calls these as Enablers:

Figure 4. Example enabler story

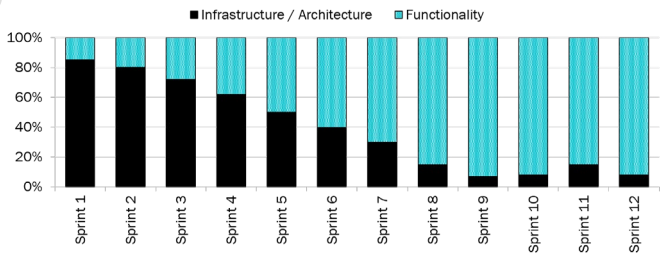
There are many other types of Enabler stories including:

- Refactoring and Spikes (as traditionally defined in XP)
- Building or improving development/deployment infrastructure
- Running jobs that require human interaction (e.g., index 1 million web pages)
- Creating the required product or component configurations for different purposes
- Verification of system qualities (e.g., performance and vulnerability testing)

- Refer: https://www.scaledagileframework.com/story/

Handling Architecture and Infrastructure in Scrum

- Unlike Non-Scrum, Architecture is not finalized before Sprints
- In Scrum, as per theory, Architecture evolves along with the product
  - So, start with just-enough architecture to support the increment in 1st Sprint
  - Gradually scale and add more architectural components

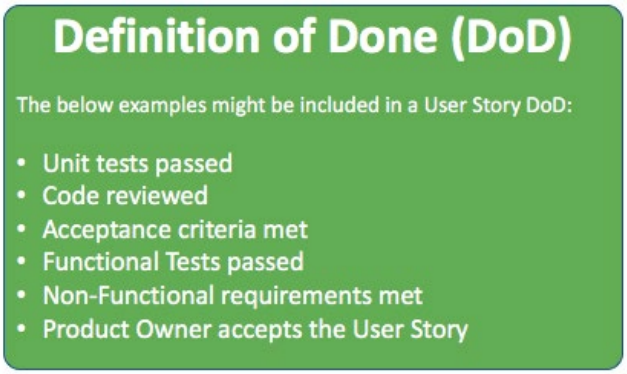


But in each Sprint, you must deliver some value

Definition of Done

- Generic for all backlog items part of Increment

	Definition of Done
❑	Design reviewed
❑	Code completed
❑	Code refactored
❑	Code in standard format
❑	Code is commented
❑	Code checked in
❑	Code inspected
❑	End-user documentation updated
❑	Tested
❑	Unit tested
❑	Integration tested
❑	Regression tested
❑	Platform tested
❑	Language tested
❑	Zero known defects
❑	Acceptance tested
❑	Live on production servers



Creating the Definition of Done

- Product Development (Projects) are typically executed between two parties (a customer and a supplier/vendor). The Supplier/Vendor (may be called as Development Organization) may provide initial DoD that can be derived from its contract with Customer and/or any other practices (for ex: ISO Standards that the vendor follows) that are practiced
  - For example, imagine "BlahCustomer" is a company that produces some Products. They have a requirement for a new Product "NewBlah". This Product will be developed by the company "VenSupp".
  - "VenSupp" may define initial DoD for this Product Development based on its contractual requirements with "BlahCustomer"
- If the Development Organization doesn't provide one, Scrum Team creates DoD. This is very important and please remember. Scrum Team creates the DoD – i.e. all 3 Accountabilities SM, PO and Dev together create the DoD.

Updating the Definition of Done

- Scrum Team can update their DoD as they progress in their Product Development.
- Typically this is done in Sprint Retrospective.

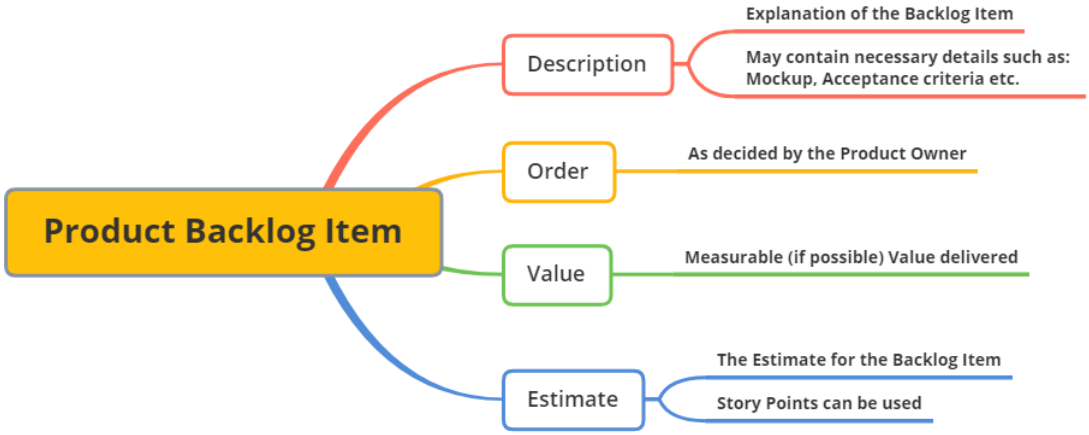
Acceptance Criteria and DoD Difference: One example –

- You want to recruit a JAVA Developer for your project.
  - Acceptance criteria (for this particular position) – role specific for ex: Must have 6+ years exp in Java, expert in ReactJS Framework etc.
  - Definition of Done (all employees in Your Company) – BG Verification, Induction ceremony, submission of documents, reference checks etc.

Product Backlog Refinement

- Activity to arrive at Backlog items with enough details for upcoming Sprints
- Spend some time in current Sprint, adding details (for ex: Finalizing Screen, Finalizing Business Logic, arriving at Database Table structure etc.) for backlog items that may be picked up in upcoming Sprint

- Add these to Backlog Items – DOVE



- One example Sprit with Backlog Refinement – 2-week Sprint

Week	Mon	Tue	Wed	Thu	Fri
1	Sprint Planning	Daily Scrum	Daily Scrum **Backlog refinement**	Daily Scrum	Daily Scrum
2	Daily Scrum	Daily Scrum	Daily Scrum **Backlog refinement**	Daily Scrum	Daily Scrum

- Example 1:
  - Backlog refinement for Order Management module – on Tuesday,
    Backlog Refinement for Service Request module – on Fridays etc.
- Example 2: for a global application
  - Backlog refinement for North America region – Tuesday
  - Backlog Refinement for EU Region – Wednesday
  - Backlog Refinement for APAC Region – Friday

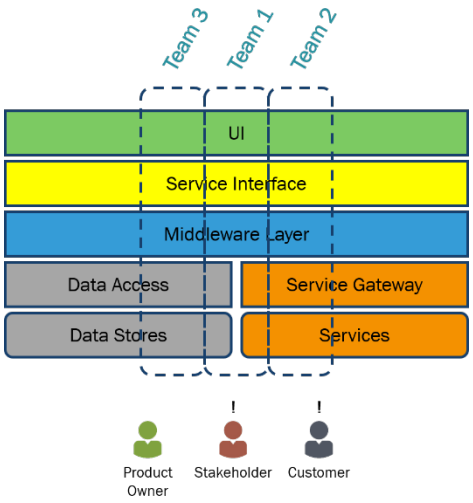
Handling Nonfunctional requirements in Scrum

- NFR's are capabilities of the system – scalability, reliability, performance, security, availability etc.
- To handle them there are 2 approaches:
  - Approach 1:
    - Create Backlog items for each NFR, Add them to Product Backlog (in consultation with PO)
    - In each Sprint take some NFRs and deliver
  - Approach 2:
    - If the NFRs impact several or all backlog items (ex: Security, encryption, page load times etc.) DoD can contain NFRs

Scrum Teams – Formation

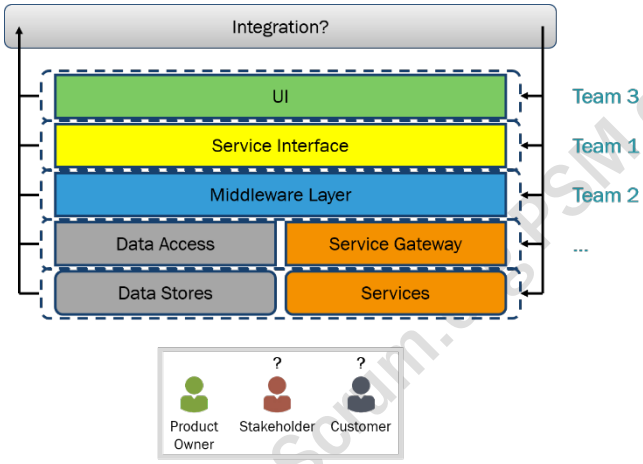
- Important factors to consider when creating Scrum Teams:
  - Ensure that all teams are cross functional i.e., as a team they have all skills required to produce the Increment in each Sprint
  - Minimize dependencies

Feature Team



- Team will have all the skills required to deliver Business functionality
- Value will be delivered quickly
- Difficult to form

Component Team



- Team created on technical skillset
- Dedicated in working one technical layer
- For example:
  - DBA Team
  - React JS Team
- Easy to form
- You might need extra time for integration so value will be delivered late.

- You can have a mix of feature and component teams as per your project needs

---

- *Team members decide to which team they belong.*
  No management or HR assigns team members

Impact on Productivity and Velocity when team composition changes

- Situation – two new members are added to Scrum team
- Productivity and velocity will initially decrease due to KT involved, later they will increase

Team health check: https://engineering.atspotify.com/2014/09/16/squad-health-check-model/