

Definition of Ready (DoR) vs. Definition of Done (DoD)



Brian Will | [Follow](#)
Providing Agile Transformati...



124



4



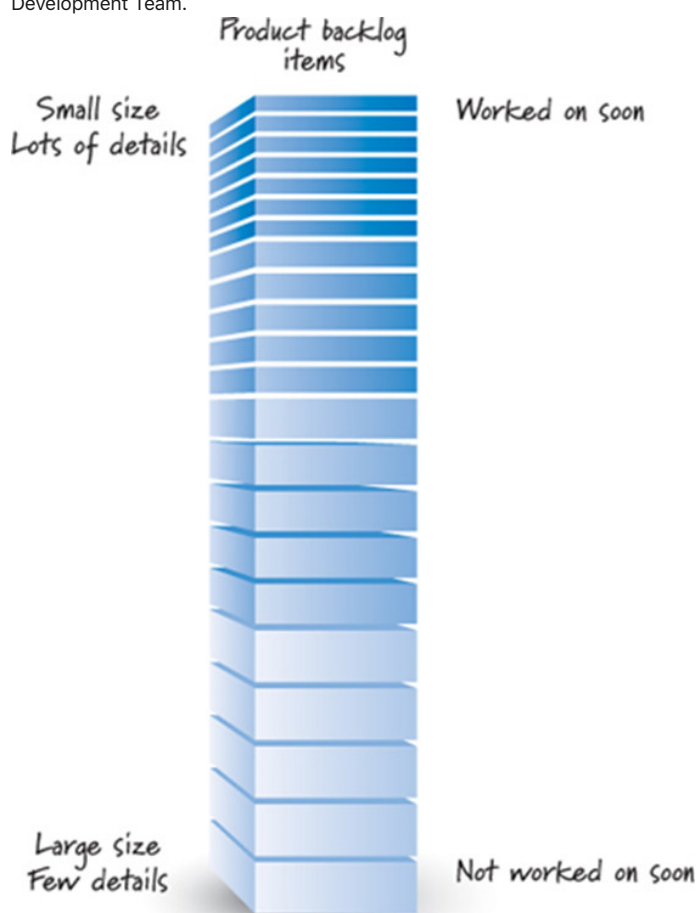
26

The Big Picture

One of the key artifacts of Agile / Scrum is the idea of backlogs – both at the Product level as well as at the Sprint level.

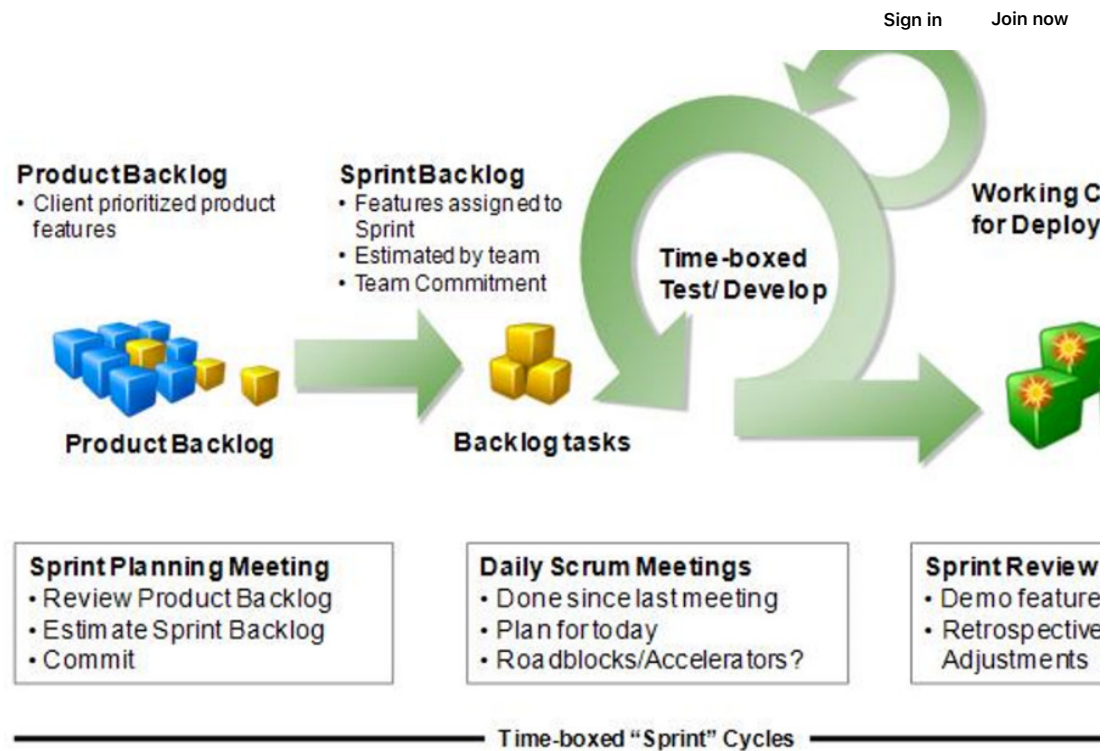
The Product Backlog is essentially the project's To-Do list, or requirements repository.

All items that are deemed in scope for the project, regardless of level of detail, are in the Product Backlog, and they are ordered, not just prioritized – meaning the one on top is more important than the one in 5th position, which is more important than the one in 23rd position. Order is decided by the Product Owner and usually driven by business value – in consultation with the Development Team.



What is known of the projects scope, is written down and documented in the form of user stories, with the expectation that discovery will lead to change. The Product Backlog is a living document (or repository) and is owned by the Product Owner.

The Product Backlog is the source for the Sprint Backlog. Whereas the Product Backlog represents the requirements repository for the project, the Sprint Backlog is the agreed upon scope for the next upcoming sprint and as such represents the Development Team's deliverable commitment.



Once agreed upon and committed to, the Sprint Backlog usually does not change in order to ensure the Development Team can deliver against their commitment.

The Development Team works through the Sprint Backlog top to bottom (most important to least important) and if it runs out of work would usually look at the Product Backlogs' top items in order to pull additional work into the current sprint.

If items are not finished during the current sprint, then they revert back to the Product Backlog to be reconsidered for the next sprint in the upcoming Sprint Planning Meeting.

Newly discovered requirements are added to the Product Backlog in the form of user stories, to be considered for future sprints.

Definition of Ready (DoR) vs. Definition of Done (DoD)

A sprint is a time-boxed development cycle that takes high-priority items off the Sprint Backlog and turns them into a product increment. However, in order to successfully pull items into the current sprint, it is important that the defined user stories are "ready" – pulling unfinished or unrefined user stories into a sprint causes problems during the implementation cycle, as it follows the old principle of "garbage in, garbage out". If developers work off of insufficiently detailed or defined user stories, they are unlikely to produce high quality code.

A "ready" backlog item needs to be clear, feasible and testable:

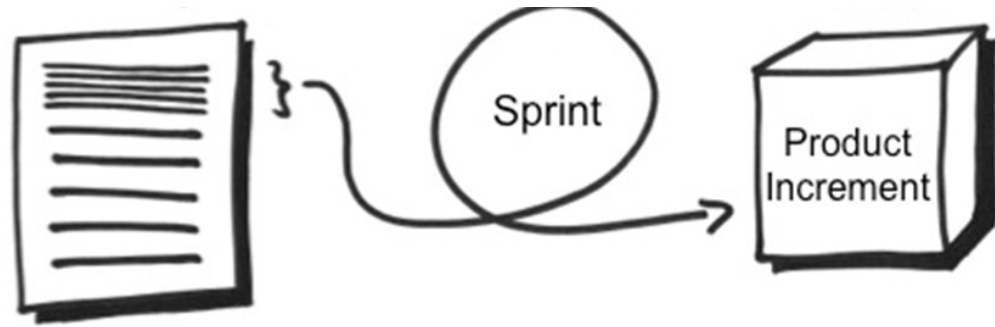
A user story is **clear** if all Scrum team members have a shared understanding of what it means.

Collaboratively writing user stories, and adding acceptance criteria to the high-priority ones facilitates clarity

An item is **testable** if there is an effective way to determine if the functionality works as expected. Acceptance criteria ensure that each story can be tested

A user story is **feasible** if it can be completed in one sprint, according to the Definition of Done. If this is not achievable, it needs to be broken down further

Simply stated, the Definition of Ready defines the criteria that a specific user story has to meet before being considered for estimation or inclusion into a sprint.

[Sign in](#)[Join now](#)

Whereas a Definition of Ready is focused on user story level characteristics, the Definition of Done is focused on the sprint or release level. Essentially, a DoD represents the acceptance criteria for a sprint or release. It spells out what the Development Team has to cover in order for the product increment to be considered “done”.

The Definition of Done is an agreement between Development Team and the Product Owner on what needs to be completed for each user story – and it is often standardized across the company in order to guarantee consistent delivery of quality.

Things that commonly addressed in the Definition of Done are:

Operating environments and at what level of integration are user stories expected to work (what specific version of Linux, what specific version of Android, iOS, or browser)?

What level of documentation is required (automatically generated Javadoc vs. fully edited end user documentation)?

What are the quality expectations (basic functionality works for demo purposes vs. fully designed and bullet proofed app)?

What are the security expectations (no security implemented vs. security vetted at all levels, from code reviews, code scans, up through network security configuration)?

Scalability expectations (scalable for demo purposes up to 10 concurrent users vs. scalable to 100,000 concurrent users)?

Essentially the Definition of Done are the agreed upon acceptance criteria that the Product Owner will use to accept the product increment at the end of the sprint.

Please note that the DoD may be different for sprints vs. releases, meaning intermediate sprints might have a less stringent DoD than the final couple of sprints before you are planning to release to market.

Sample Definition of Ready (DoR)

- User Story is clear
- User Story is testable
- User Story is feasible
- User Story defined
- User Story Acceptance Criteria defined
- User Story dependencies identified
- User Story sized by Development Team
- Scrum Team accepts User Experience artefacts
- Performance criteria identified, where appropriate
- Scalability criteria identified, where appropriate
- Security criteria identified, where appropriate
- Person who will accept the User Story is identified
- Team has a good idea what it will mean to Demo the User Story

Sample Definition of Done

- Code produced (all ‘to do’ items in code completed)
- Code commented, checked in and run against current version in source control
- Peer reviewed (or produced with pair programming) and meeting development standards
- Builds without errors
- Unit tests written and passing
- Deployed to system test environment and passed system tests
- Passed UAT (User Acceptance Testing) and signed off as meeting requirements
- Any build / deployment / configuration changes are implemented / documented / communicated
- Relevant documentation / diagrams produced and / or updated
- Remaining hours for task set to zero and task closed

Other uses of the Definition of Ready and Definition of Done

The original intent of DoR and DoD was to create brief documents representing internal agreements between the projects’ stakeholders, the Product Owner, and the Development Team.

expectations as to what needs to be done.

These are useful tools for negotiating project scope as they define expectations and hold both parties accountable; the DoR helps the customer for producing well written user stories that are ready to be consumed by the Development Team, and the DoD helps the implementation partner for producing working product increments according to all project requirements, not just the specific user story functionality.



Brian Will
Providing Agile Transformati...

[Follow](#)

4 comments

[Sign in](#) to leave your comment

DSK Chakravarthy
Change practitioner with 16+ years of IT experience, worked in all phases of software development

5mo

There could be some stories that may not Defined as Ready, but, they can be DoR enabled after the discussion with stakeholders during the sprint planning. At time, it is possible that PO might not be ready with business knowledge / requirement, but, if the BA (or) any team member provide a decent idea in the interest of the business, that can be considered as DoR for that story. Thus, in few cases, DoR will evolve.

Like Reply | 1 Like

Aditya Pandey
Agile Practitioner - AVP at JPMorgan Chase & Co - PSM | ICP-ACC | Data Science

1y

please elaborate on scalability criteria and security criteria in DOR

Like Reply

Aditya Pandey
Agile Practitioner - AVP at JPMorgan Chase & Co - PSM | ICP-ACC | Data Science

1y

Hey Brian Sir, I must say this article of yours is incredibly helpful as it is too the point and hit the spot. Also you have used very simple language to share this knowledge. Thanks a lot.

Like Reply

Satyanarayana
Yogachar

Satyanarayana Yogachar
Agile Coach at Wipro Limited

1y

Nice information

Like Reply

More from Brian Will [30 articles](#)

Why #NoEstimates gets it #AllWrong

[Why #NoEstimates gets it #AllWrong](#)

April 6, 2017

Agile Certification Puppy Mills

[Agile Certification Puppy Mills](#)

March 2, 2017

Delivering Good Product Increments

[Delivering Good Product Increments](#)

February 22, 2017