Table of Contents

What Does It Mean To Be Done?

Or what is the "Definition of Done"?

Premise

One of the key ideas in Scrum and agile is to focus delivering valuable work (or more basically, running, tested features.) There are two key tools in understanding when something is complete:

- Acceptance Criteria (or Conditions of Satisfaction): From an end-user perspective, what does it mean for this slice of a feature to be considered complete. Here we talk about the workflow that is being supported. In other words, the CoS describe the "extrinsic" quality standard of the work, what external people directly see.
- Definition of Done (DoD): From a Scrum Team perspective, have we done everything we need to in order to say this is a good piece of work. Here we talk about testing, coding standards, documentation and so on that tells us that all the work is complete. In other words, the DoD describe the "intrinisic" quality standard of the work, what external people don't usually see, except afterward when they see defects, for example.

Purpose

The goal of the DoD is for every team to be able to say "we are potentially shippable at the end of every Sprint." The idea is that we can then let the business make decisions on when to release based on the value provided as opposed to the current place we are in the release plan. While this is important from a business perspective, the real benefit of the "potentially shippable" standard is that at any given time we know the status of the software and have low risk going forward because we have done all the work required in order to release our software. At many organizations I have worked at most of the (legacy systems and) software is not in this state and so part of this discussion is that we need to be working to get closer and closer to this state.

The "Definition of Done" (DoD) helps us:

- Understand what it means to be done from an internal perspective.
- Reduce the build-up of Technical Debt by making sure we don't cut corners in our work. This is because our detailed plan for the Sprint takes into consideration all the things that are part of the DoD.

Sources

Teams are responsible for maintaining their quality standard. They do this by:

- Working to their DoD
- Working to make their DoD tighter and tighter as they become more effective at delivering software.

The Scrum Team's DoD will not be a purely team creation. Definitions of Done will be influenced by the business. The input will come in to support differing levels of need:

1. Corporate: Corporate standards that need to be met
2. Portfolio: Portfolio standards that need to be met
3. Release: Criteria that must be met to consider a Release ready for production
4. User Story: Criteria that must be met to know that all the work is complete
5. Scrum Team: Things the team learns it needs, especially at a technical level

Teams have a known, public DoD. Teams are explicit both in being visible about their DoD (available on the Scrum Team's page). They are also transparent about the practice they use to ensure their Definition of Done is addressed. It is up to the team to determine their approach.

Example approaches we have seen include:

- Teams create tasks associated with their definition of done so that when the sub-tasks are done, the story is done.
- Teams maintain a spreadsheet and check-off the definition of done at the end of the Sprint.
- Teams create a task that is the Definition of Done checklist that is updated as work proceeds on the story

Process

As said, the aim of this DoD is to ensure we are getting closer to "potentially shippable". The DoD typically is created to cover most of the work that the team typically does. So if you are a company doing new feature or maintenance work, then your DoD would be created to match most of that work. Other DoD's might be required for other kinds of work:

- If your team does work that is not necessarily new feature or maintenance oriented. For example a documentation or QA team would have different considerations. (Note: this is not an ideal set up but still does happen - see What Should We Consider When Forming a New Team for more discussion here.)
- If your team identifies different types of work. For example, a research user story (spike) would have different DoD than development work.

The key thing for the DoD is that all the stakeholders understand what it means when the team says it is

done so communication is a required part of the on-going work associated with the DoD.

Sample

Here is an example DoD for a team, for new product and on-going maintenance development work:

- Code is complete:
    - Build and package changes are communicated to build master (e.g., when introducing a new file)
    - Code meets product coding standards
    - Code has been run through static and/or dynamic analysis tools (e.g., Coverity)
    - Code is checked into source control (with developer and peer reviewer names recorded)
    - Internationalization considerations have been met
    - Internal and external documentation is updated with applicable support information included such as help files and public APIs
    - Code is integrated with the main trunk (if not already done)
    - Code (or other deliverables) has been peer reviewed (e.g., pair programming counts as peer review)
- Code meets quality standards
    - Test plans are in place for: Unit, Function, GUI
    - Test plans have been run and passed and (hopefully) automated
    - Test plans are in place for the '-ilities' including performance, stress, load, security, etc
- No new defects
- End-user documentation is complete
    - Internal and external documentation is updated with applicable support information included such as help files and public APIs
- "Acceptance Criteria" (or "Conditions of Satisfaction") or have been met
    - Some organizations also add in "Product Owner has accepted the story" as part of this thinking
- Item statuses in related tracking systems (Jira, Siebel, etc.) have been updated

Want to Know More?

- How Do We Initially Setup Our Definition of Done

Consultant, Tools, DefinitionOfDone, DoD, FirstSprint, FAQ