DoD = Definition of Done

The DoD is usually a short document in the form of a checklist, that defines when a product backlog item (i.e. user story) is considered “done”. It has various rationales and various ways to explain it:

- You need a **common definition of what “done”** (= “this user story is finished”) **means**. Otherwise it will mean something else for every person on the team.
- All your **non-functional requirements** reside in the DoD.
- A **general list of acceptance criteria** to be added to every story’s specific acceptance criteria.
- Many **improvements** you find in your retrospectives end up in the DoD.

Most teams start with no or a very simple DoD. They then add to the DoD after each sprint as needed. *Tip: Don’t paralyze yourself with an excessive DoD!* But keep in mind: “done” in an agile project means **“no more work needs to be done before shipping”**. So if someone says “the feature is done, but it only needs to be integrated, tested, deployed, …” it would NOT be considered “done” in an agile sense!

The best **check whether something is “done” is to simply ship it**! If you can ship it, it’s really done; if you cannot ship it, simply do the work missing before you can ship it to make it “done”. *Mind you: you don’t need to actually ship it, but you need to make believable that you could.*

A typical DoD might look like this example:

- Automated tests are written and all tests are green
- Code is refactored and reviewed
- Code is integrated with master branch
- Deployed to staging environment
- Translated into English and German

A concise definition of done will help you deliver quality, keep your slate clean and react flexibly to changing requirements.

DoR = Definition of Ready

The DoR is the little cousin of the DoD. It is a checklist of what needs to be done to a product backlog item **before the team can start implementing it** in the next sprint. You can view the definition of ready as the “DoD” the Product Owner has to fulfill so that the Development Team accepts the story in the Sprint Planning meeting.

A typical DoR might look like this example:

- PO and Dev Team need to have talked about the story at least once
- Story must have clear business value
- Effort needs to be estimated
- Story must be broken down enough to fit a single sprint
- Story needs at least one acceptance criterium

In all honesty, you will likely not need to write this down. When you talk about new user stories in a backlog refinement session, you will intuitively drive stories towards being “ready for sprint”.

- **Independent**. A user story should be independent from other stories. If you really write “user stories” as opposed to traditional work items or tasks, you will end up with drastically fewer dependencies automatically. Occasionally you might still have dependencies — in that case simply note the dependencies.
- **Negotiable**. A user story should describe *what* the customer needs, not *how* the developer should implement it. The development team should always be able to propose alternative solutions/implementations to deliver the business value for the customer.

- **Valuable**. The business value must be stated. This is often the "…so that…" part of the user story format: "As a *persona* I want *feature* so that I get *business value*".
- **Estimable**. The development team has to be able to roughly estimate the effort of the user story. This often means that the development asked the product owner a few clarifying questions and came up with a rough idea of how it could be implemented.
- **Small**. It has to be small enough to be done within a sprint. If it is estimated to be bigger than a sprint, keep splitting the user story until you have small stories.
- **Testable**. You need to be able to test, whether the user story is done and fulfills its purpose. This usually means you need a set of clear acceptance criteria that you turn into test cases.

Again, don't over-theorize the DoR. Stick with INVEST or agree on a simple format the development team needs before they can sensibly start work. Both the product owner AND the development team are responsible for getting a story ready in the sense of your DoR.

When do you create the DoD & DoR?

Both, DoD and DoR, are typically edited and extended in the most important meeting in Scrum: the retrospective.

In my experience it pays to start with a very simple (or even empty) DoD and no formal DoR. Most teams quickly write this down collaboratively during an initial meeting before the project starts.

As the project continues, you will often find solutions to impediments during your regular retrospectives (a meeting to improve your work process held after every sprint). Many of these solutions end up in the DoD or DoR.