# DEFINITION OF DONE

According to the Scrum Guide, “When a product backlog item or an increment is described as ‘done’, **everyone must understand what ‘done’ means**. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of ‘done’ for the Scrum Team and is used to assess when work is complete on the product increment.”

Mostly, a **bare-minimum Definition of Done should produce a complete set of product functionalities includes designing, coding, integrating, testing, and documenting**, which at the end will result in delivering a validated value to the customer. However, the tasks can be further refined to get a more specific checklist. The product is “releasable.”

1. PROJECT INFORMATION
   1. Project Name
   2. Date of document submission
   3. Version of document (if any changes)
   4. Client (name of client company)
2. PROJECT TEAM

|  |  |  |
| --- | --- | --- |
| Team Member | Role | Responsibilities |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

State which team member is responsible for writing which section in this document.

1. PROJECT SCOPE AND DEFINITION

Outline the strategic vision, goals & objectives and ideally include a high-level mission statement. This will help align the team on the approach and keep these goals in mind during solutioning. It will also help in defining additional work and potential project enhancements as the team keep this context in mind.

1. Define the project. What is the project about?
2. What is the scope of the project?
3. What is the problem to be solved?
4. What specific outcomes will be achieved? How will they be measured?
5. What are the deliverables?
6. What are the milestones for the deliverables?
   1. If possible, identify projected increments by Sprint
7. Are there any constraints that may influence your deliverables and schedule?
8. COMMON TERMS AND DEFINITIONS
9. HOW IS SUCCESS DEFINED?

How do you define success?

Are there any metrics that can be defined that measure success?

Explain any processes or plans that ensure product quality.

1. DEFINITION OF DONE: CHECKLIST
2. List the primary Epics in your Product Backlog.
   * Look at the use case diagrams (big picture)
   * Break that down into user stories
   * Version creation, ease of use, sprint increment steps (what they expect to accomplish), testing (using debugger, code, coming up with sample inputs),
   * Performance testing, unit testing, (describing types of testing)
3. Explain the (preliminary) logical progression of increments in development.
   * i.e. Sprint 0, Sprint 2, … Sprint 5
   * What are the potential shippable products for each Sprint?
   * expectations
4. How will “code” for each increment be tested? Describe how you may incorporate the following tests.
   * Code builds with no errors
   * Code peer review
   * Documentation/comments
   * Functionality testing. Test everything. See if anything crashes.
   * Unit testing. Method, class, or component. test across a range of valid and invalid inputs
     + Test tidbits, then how they work altogether
     + Is this working the way everyone thinks it’s working
     + Keep going until found errors
     + Comments in all scripts
   * Performance testing
5. [Functional testing guide](https://www.browserstack.com/guide/functional-testing)
6. [Functional testing types with examples](https://www.simform.com/blog/functional-testing-types/)
7. BURN UP CHART

Provide the most recent Burn Up Chart *with estimated trajectory* for the project.

1. ACCEPTANCE CRITERIA
   1. Explain the difference between Epic, User Story, Acceptance Criteria, and Tasks.
      1. Explain their definitions
   2. How do the acceptance criteria in the Product Backlog help define “Done?”
2. PRODUCTION ENVIRONMENT
   1. Explain how the shippable product will be “rolled out” from the development environment to the production (live) environment.
      1. How is this going to be delivered as a final product
      2. Docker container? Live environment?
      3. Test environment? Manual testing locally?
   2. Explain whether the team will be using a testing environment? If so, describe how the product will be tested in that environment?
3. USER DOCUMENTATION
   1. Briefly describe what will be included in a user documentation for the final product.
      1. For the user manual created at the end
      2. Using the product for the first time (visual)
      3. Not a lot, but just a basic outline

If we don’t have the same kind of project, request