# Tip Sheet: Understanding the "Definition of Done" for the Success of Your Project

pmm
pmmajik.com

## 1. What is the Definition of Done (DoD)?

The Definition of Done (DoD) is a shared understanding among the project team, sponsor and stakeholders about what criteria must be met for a User Story (or any work item) to be considered complete. It is a checklist of tasks, deliverables, or quality standards that must be fulfilled before moving on from the work.

When applied correctly, the DoD brings consistency across the team and avoids misunderstandings about the scope or quality of completed work. This is especially important when teams work across different locations or time zones or when collaborating with external stakeholders.

In Agile, DoD is more than just finishing development or passing a unit test—it also involves meeting various quality and acceptance standards, ensuring the work is functional, tested, and, in many cases, usable in production.

## 2. Why is it important?

The **INVEST model** is often used to evaluate the quality of a user story. It stands for:

- **Independent**: The story should be self-contained, avoiding overlap with other stories.
- **Negotiable**: A user story is not a detailed contract; it's an invitation to discuss.
- **Valuable**: It should deliver value to the user.
- **Estimable**: The team must be able to estimate the effort required to implement the story.
- **Small**: Keep the story small enough to be completed in a sprint or iteration.
- **Testable**: You should be able to verify that the story has been implemented correctly.

## 3. Key considerations of DoD

**1. Code Completion**

- Code Review: Has the code been peer-reviewed for standards and best practices?
- No Known Bugs: Are critical bugs resolved, with no showstoppers remaining?
- Adherence to Coding Standards: Has the developer followed code conventions and patterns?

**2. Functionality**

- Acceptance Criteria Met: Does the functionality meet all user story acceptance criteria?
- Integrated with Other Features: Is the functionality properly integrated, without breaking existing features?

**3. Testing**

- Unit Testing: Are individual components tested, with sufficient code coverage (e.g., 80%)?
- Integration Testing: Does the feature work correctly with other system components?
- Acceptance Testing: Has end-to-end functionality been verified from a user perspective?
- Automated Testing: Are there automated tests to prevent future regressions?

**4. Documentation**

- Code Comments and Documentation: Is the code properly commented, especially in complex areas?
- User Documentation: Has user-facing documentation (e.g., help guides, API docs) been updated?
- Update of Specifications: Are design/specification documents updated to reflect completed work?

**5. Non-Functional Requirements**

- Performance Benchmarks: Does the feature meet performance and scalability expectations?
- Security: Are potential security issues mitigated, and is the code secure?
- Usability: Has the feature been tested for ease of use, accessibility, and UX issues?

**6. Deployment and Release**

- Ready for Deployment: Is the feature packaged and ready for staging or production deployment?
- Release Notes: Are release notes or changelog entries created for the feature?
- Rollback Strategy: Is there a clear rollback strategy if something goes wrong in production?

## 4. Example of checklist

Here is an example of what a DoD checklist might look like for a typical software development project:

- Code has been written, peer-reviewed, and complies with coding standards.
- All acceptance criteria are met.
- All unit tests pass, with a minimum of 80% code coverage.
- Integration tests have been executed to ensure no new bugs are introduced.
- User documentation is complete.
- Performance benchmarks are met.
- Security considerations have been reviewed and addressed.
- The feature is deployed to the staging environment without issues.
- Release notes are created, and the product owner has signed off on the feature.

## 5. More Information

For more information, visit the following blog post:

https://www.pmmajik.com/understanding-the-definition-of-done-for-the-success-of-your-project/

If you are interested in a complete Agile Template Framework, visit the link below:

https://www.pmmajik.com/members/agile-template-framework-info/