Eleven real-world definition of done examples

Get ideas for your own definition of done by checking out what other teams have come up with.

Scrum Masters from a range of projects have kindly shared these examples. Keep in mind that they show actual practice, not best practice. To find out how to make your definition of done most effective, check out our guide to creating a definition of done.

The examples

Example 1: Printed resources (non-software project)

- Design researched
- Style guide checked
- Document set up
- Concepts designed
- Concepts reviewed with product owner
- Content researched and written
- Document designed
- Content copy-and-pasted to avoid typos
- Designs checked by content writer
- Designs peer reviewed
- Acceptance Criteria checked off
- Feedback from product owner implemented
- Spell-check run
- Proofread
- Print-ready PDF created
- Test print approved
- Final files supplied to product owner for printing
- Put up for acceptance
- Design files saved and filed
- Any out-of-date files archived
- Any out-of-date hard copies removed

Example 2: Web application

- New styles documented in style guide
- Images optimised for web
- Spell check when at the design level
- Assets are filed correctly
- AA accessibility considered
- Review approach of each story with Product Owner at regular intervals
- Code is peer reviewed
- Had a design review
- Outcomes available on staging
- Merged to master
- Static security vulnerability scans are passing
- Integration tests for all features are passing
- Unit tests for business logic are passing
- Tested on the following devices:
    - Android
    - Samsung Internet
    - iOS
    - Safari
    - Chrome
- Tested on the following browsers:
    - Chrome - latest version
    - Safari - latest version
    - IE11
    - Microsoft Edge - latest version
- Recorded in Sprint report

Example 3: Internal web application and external website

- All code committed to Git
- Unit tests for all functionality
- Deployed to staging site
- Cross browser tested
  - Internal interface: IE10+, Safari, Chrome & Firefox
  - Public interface: IE8+
- Cross device tested
  - Mac, iPad, Android tablet & Windows
- Scaleable 760px +

Example 4: Web application

Before being added to the backlog

- Refine and size story

When picking up a story

- Consider how many stories are in progress
- Start story on Rally
- Ping PO [Product Owner] on Slack
- Discuss approach with the team

Start Development

- Create new branch and pull request
- Develop and test
- Code review
- All test passing
- Cross browser testing

Put up for acceptance

- Acceptance template filled and sent on Basecamp
- Ping PO on Slack
- Move story to completed in Rally
- PO to accept or reject
- Move to accepted on Rally
- Feedback implemented
- Deploy to staging

To be done

- Merge branch to master
- Deploy to production

Example 5: iOS mobile app

- All acceptance criteria are met
- Must have BDD [Behaviour Driven Development] tests written
- Must have TDD [Test Driven Development] tests written
- Must be pushed to Testflight
- iOS only
- Tested on iOS versions 6 and 7
- Must be committed to repository
- Acceptance email to be sent to Product Owner

Example 6: Website

Design

- Content copied and pasted
- Review with Developer (if applicable)
- Check with content writer
- Design peer review — if required
- Proofread — if print
- Check with PO
- Supply and check all assets are available to developer
- Meets accessibility standards

Content

- Checked by subject matter expert/stakeholder/client if needed
- Search-engine-optimised
- Content linked to relevant pages
- Meta title, description, slug, alt text and navigation label and position supplied
- New pages submitted to Google search console
- CMS [Content Management System] content (not needing developer) loaded
- Proofread
- Links and metadata checked by writer
- Content published or live
- Posted on Facebook, Twitter, LinkedIn
- Share with team and/or client

Developer

- Create new branch and pull request, update master branch
- Content copied and pasted from design document
- Optimise images and add URL to image for accessibility
- Meta title, description, slug, alt text and navigation taken from content document
- Design and content review
- Proofread
- Meets accessibility standards
- Tested (links, mobile, tablet and cross-browser testing)
- Code reviewed if required
- Use Developers.Google.com for optimisation
- Merge to master
- Notify team once live

Example 7: Web application

- All stories are for one persona, with details as appropriate
- All stories have clear, achievable acceptance criteria
- All stories define error handling where relevant with acceptance criteria
- User story have been discussed and agreed by Scrum team
- Test driven development in place for each story
- Acceptance criteria met
- Design implemented and reviewed
- Frontend work is checked for accessibility
- Code reviewed twice (include external)
- Browser tested by priority
    - Chrome
    - IE 11
    - Firefox
    - Safari
- Mobile devices tested (include device orientation)
- Error messages are shown when necessary
- Dev documentation updated
- Worked is merged and deployed
- Release notes updated when stories are merged
- Acceptance message in Pivotal detailing what was done and how to test

Example 8: Website and web application

Pending (Start)

(To be brought in to the sprint backlog)

- Story has been refined by team, except for meteors
- Tasked before picked up

Pending > Develop (Finish)

(To move into the development phase)

- Check in with Product Owner
- Story started in Pivotal Tracker

Develop > Deliver (Finish)

(To move into the Develop to Deliver phase)

- Branch has been made (except for non-dev stories)
- Demo the story with Product Owner on local or UAT
- Testing the functions of the feature by Product Owners
- Tests written

Deliver > Code Review (Deliver)

(Before being put up for code review)

- Check acceptance criteria has been met
- Cross browser testing on Firefox, Chrome, IE11, IE Edge, Safari
- Mobile testing on Android, iPhone and iPad on latest Chrome and Safari
- Accessibility errors and warnings checked and fixed
- Deployed to staging or UAT (depending on story) and tested
- Developer and Product Owner to discuss test strategy and Product Owner to test
- Feedback from Product Owner is resolved

Code Review > Acceptance (accept/reject)

(Before being put up for acceptance)

- WIP [Work in Progress] merge request opened including notes on solution and screenshots
- CI [Continuous Integration] build passing
- Code reviewed then peer reviewed by 2 people (one in and one

- outside team) and checklist updated
- Systems-related stories to be peer reviewed by Systems team
- Feedback from peer review resolved (if major change return to Product Owner for
- testing)
- Documentation updated (if applicable)
- Acceptance message written with instructions needed to test feature
- (with reason for size change if applicable)

Acceptance > Done

(Before it is done)

- Accepted by Product Owner
- Merged to master and branch deleted
- Discuss production timing with Product Owner
- Deploy to cloud staging, production and preview
- Product Owner to review in production and notify stakeholders

Example 9: Website and web application

- All code committed to GIT
- Deployed to staging
- All external files compressed
- Tests written and pass BDD & TDD
- All static content uses internationalisation
- Cross browser/device tested (IE9+, iOS, Android & Litmus)
- PO accepts story on staging
- Clear Basecamp message outlining changes and links to test
- Rally updated

Example 10: Website, web application and API

This definition of done describes the work that we do on every story, unless otherwise agreed with the product owner. Product owners expect this work to be complete before stories are sent for acceptance.

This definition of done is not static, but will change over time.

To be brought into the sprint backlog

- Story has been groomed by team

To move into the analysis phase

- Story has been discussed and analysed by the team. Approach agreed
- Accessibility requirements discussed

To move into the development phase

- Branch has been made (except for non-development stories)
- Work in Progress merge request opened
- Notes of analysis phase in merge request
- Story started in Pivotal Tracker

Before being put up for acceptance

- Code reviewed then peer reviewed by 2 people
- Tests written
- Continuous Integration build passing
- Deployed to staging or production (depending on story) and tested*
- Acceptance message written (with reason for size change if applicable)
- Product owner given the required info needed to test feature
- Cross browser testing on (Firefox, Chrome, IE11, IE Edge, Safari)
- Mobile testing (Android, iPhone and iPad on latest Chrome and Safari)
- Documentation updated
- Check acceptance criteria has been met
- Accessibility errors and warnings checked and fixed

Before it is done

- Merged to master and deployed to production
- Accepted by Product owner
- Branch deleted

Testing

** It is to be decided whether or not to make all functions work across all devices (i.e. do internal editors need to work on phones?)*

- Tests should be written to protect the system.
- Tests are written first (for features and bugs).
- Specs cover all new code
- Clear unit and integration tests for all business requirements
- Integration tests cover user interaction where possible.
- Acceptance message should state the number of tests added/modified.

Documentation

Where appropriate, documentation includes:

- Updating wiki and/or GitHub documentation site.
- Clear and descriptive code
- Descriptive commit messages
- Comments on commonly use interfaces
- Comments on unclear or exceptional pieces of code.

Example 11: Mobile apps

- App testable on iOS and Android.
- Code and design peer reviewed.