Definition of Ready

# Introduction

The definition of ready (DOR) specifies the entry-criteria for an feature that is currently in in the backlog so that it's possible to work on it. A Definition of Ready enables the team to specify pre-conditions that must be fulfilled before a story is estimated. The goal is of this document is to provide a definition of done that:

* Helps the development team to properly estimate stories without inferring or discussing what features would look like during the estimation session.
* Helps story writer to identify minimum requirements that should be addressed by addressing gaps and unknowns on stories that lead to rework, change of scope or disruptive changes
* Provides clear requirements so that stories are better and more commonly understood by the team
* Helps identifying and prevents problems early enough so they don’t surface after development starts
* Reduce the time spent discussing stories. With clear requirements and dependencies met, the development team shouldn’t ask questions that potentially modify the scope and the size of the work
* Identifies dependencies and impacts reducing risk

# Requirements

We would like to propose that the following requirements are satisfied to meet the definition of ready:

* The story provides a description, personas affected and clear acceptance criteria.
* If the story requires front-end modifications, designs, mock ups, diagrams and explicit changes of any screens affected by the story are provided.
* All external dependencies have been resolved and were documented.
* If necessary, copies are written by the author. It helps decreasing external dependencies on stakeholders and helps building a shared knowledge by the team.
* Necessary spikes were already addressed and no technical questions remain.
* No business-related outstanding questions exist. They usually increase the scope of the story and invalidate the previous estimates.

We understand that for each story meeting the definition of ready, the development team can provide an estimate without neither going in too much detail nor skipping unmet dependencies/gaps in the contents of the story. With the estimate provided, the story can then be assigned to a sprint. We would also like to propose that the maximum size for stories is equivalent to half of the team’s velocity.

# Definition of Done – Template

We would like to propose the following table as a guideline to assert if the definition of ready is met.

|  |  |  |
| --- | --- | --- |
| Requirement | Y/N | Comments |
| The story has an objective description including who, what, why? Ex. Includes affected systems, objective screen names, urls, required data changes, etc? |  |  |
| The affected personas were listed? |  |  |
| The story has *sufficient* acceptance criteria? |  |  |
| If requiring front-end changes, were the copies provided? |  |  |
| If requiring front-end changes, were the wires provided? |  |  |
| Are there outstanding questions? |  |  |
| Were all the technical impediments (usually identified as Spikes or unknowns by the dev team) removed? |  |  |
| Are all the statements in alignment with each other (no conflicting statements)? |  |  |
| Is the vocabulary in use non-ambiguous? |  |  |
| Is the story *potentially releaseable?* |  |  |
| Has the story dependencies on other stories? Which? Why? |  |  |
| Do you understand what business value the story provides? |  |  |

The next sections will provide a better understanding on some of the above requirements.

# Acceptance Criteria

We would like to propose the following guidelines for the Acceptance Criteria:

1. What are the inputs and outputs?
2. What changes are expected when the story is ready?
3. Can you provide a list of high-level test cases related to the story that are understood by all team members?
4. Not-expected: list of requirements, business rules, etc.

Expected format:

* Be generic, in form of bullet points
* Avoid duplicated, conflicting or ambiguous statements
* Have at maximum, 6 requirements. We understand that more than that, potentially the story is too big.

# Wires

Wires are very important for understanding, estimating and developing the story. We propose that all stories that require front-end modifications provide wires and copies before they can be estimated by the development team.

## Exceptions

Not every requirement requires wires. As an example, these requirements that don’t require mockups:

* Changes on the business logic on the backend
* New jobs or changes in them
* Email Copy changes
* Changes on the database (patches, new columns without front-end requirements, etc)

# Copies

We usually get stories with copies like “*Please insert your title here*”, “*Placeholder for copy*”, etc. We propose that the author of the story writes the copy for front-end and emails so that we don't rely on the business to provide them because:

* changing copies on that last-minute forces tests to be redone and requires last minute builds, deployments and regressions that could be avoided.
* depending on the business for providing the copy usually delays the start of the story by the development team

We also think that providing the copies ahead of time will help developers to better estimate the story. Sometimes copies describe features the developers didn’t think of were they not provided.

# Dependencies

As previously discussed, the team should ensure that external dependencies are met (except for copies). That includes business logic, external dependencies and potentially technical requirements. Unknown requirements, complexities and technical dependencies should be excluded from the estimates thus simplifying the discussions about the actual requirements and not on what they should be. In case there are unknowns, the team should reject the story proposing spikes or follow-up discussions.

# Changes

Be noted that the objective of the Definition of Ready is not to create hard dependencies and roadblocks for the stories to be estimated. That said, we do expect that stories accommodate slight changes as it’s normal and expected that some refinement happens during development and we are open to them.

# For the Development Team

In order to accommodate potential disruptive changes (see section Changes for more details) during the development of a story, we would like to propose that the development team adheres to the following practices:

* all stories are developed on a separate branch. That would protect the team in case either disruptive changes are required for a given story or if the development team cannot finish the story on time.
* any significant disruptive change is evaluated and if big enough, the team will request that the story is removed from the sprint
* spikes are created for the complex technical dependencies identified when discussing the stories
* the maximum size for stories is equivalent to half of the team’s velocity
* each story is treated as a potentially releasable feature

# References

<https://github.com/ca-cwds/devops/wiki/Definition-of-Ready>

<https://github.com/camptocamp/c2cgeoportal/wiki/Definition-of-Ready>

<https://github.com/ONSdigital/dp/blob/master/DEFINITION_OF_READY.md>

<https://github.com/citrusframework/citrus/blob/master/CONTRIBUTING.md#definition-of-ready>