|  |
| --- |
|  |
| Agile Project Planning Template |
|  |
| From initiation to Closure |

Contents

[2. Project Chartering 2](#_Toc361918677)

[Lift off planning 2](#_Toc361918678)

[Purpose 2](#_Toc361918679)

[Alignment 2](#_Toc361918680)

[Context 3](#_Toc361918681)

[3. Overall direction 3](#_Toc361918682)

[Timeline summary 3](#_Toc361918683)

[4. Milestones and milestone management 3](#_Toc361918684)

[Project Milestones 3](#_Toc361918685)

[Milestone tracking 5](#_Toc361918686)

[Example 6](#_Toc361918687)

[5. Risk and Issue management 6](#_Toc361918688)

[Example 6](#_Toc361918689)

[6. Assumption management 7](#_Toc361918690)

[7. Dependency management 7](#_Toc361918691)

[8. Communication plan 8](#_Toc361918692)

[Minutes of meetings 8](#_Toc361918693)

[9. Quality Plan 8](#_Toc361918694)

[User Stories 8](#_Toc361918695)

[Definition of Done (DoD) 9](#_Toc361918696)

[Creation of the DoD 9](#_Toc361918697)

[Review of the DoD 9](#_Toc361918698)

[Example 9](#_Toc361918699)

[Design 9](#_Toc361918700)

[10. Test Strategy 10](#_Toc361918701)

[The 4 quadrants 10](#_Toc361918702)

[The pyramid of tests 10](#_Toc361918703)

[11. Software framework 11](#_Toc361918704)

[12. Release Management 11](#_Toc361918705)

[13. Defect Management 11](#_Toc361918706)

[14. Roles and Responsibilities 11](#_Toc361918707)

[15. Resource Management and Planning 13](#_Toc361918708)

[16. Budget 13](#_Toc361918709)

[People 13](#_Toc361918710)

[Software 13](#_Toc361918711)

[Hardware 13](#_Toc361918712)

[Trainings 14](#_Toc361918713)

[Travels 14](#_Toc361918714)

[Miscellaneous 14](#_Toc361918715)

[Summary 14](#_Toc361918716)

[17. KPIs (Project) 14](#_Toc361918717)

[Team Maturity 14](#_Toc361918718)

[Metrics 14](#_Toc361918719)

[18. Compliance constraints 14](#_Toc361918720)

[19. Reference 14](#_Toc361918721)

*Note from author: an agile project plan is a high level view on how the project will be managed and it should never contrevine the Agile Manifesto and principles. This document should remain lightweight and only use as guidelines to drive the project to deliver valuable working software in short iterations to the customers.*

# Project Chartering

## Lift off planning

This document’s value is in the lift off session [5] where stakeholders, team members, product owners, scrum masters and project managers will create or consolidate the project chartering: purpose of the project, alignment of the team around the project and context of the project.

Initially, the Product Owner should draft the purpose of the product and the context. Project chartering will be an output of the lift off session.

## Purpose

* What is the vision? Value to Attain
* What is the mission? Result to Accomplish
* What are the mission tests? Criteria for Success

## Alignment

* Values & Principles - Beliefs & Ideals about Work
* Working Agreements - Operational Guidelines
* Core Team - Cross-functional group with a common purpose

## Context

* Committed Resources - Organization Support
* Boundaries & Interactions - Seeing the Systems
* Prospective Analysis - Initial Projections

# Overall direction

For the first user story implemented, we will deploy this feature to live. “There is only value if features are delivered to the customers”.

Quality is not only for the code and test code, this is also about the quality of the user stories, acceptance criteria, meetings, etc

To align the design of the solution with the user stories, we will use domain driven design ideas and keep the design readable by the product owner of the product.

## Timeline summary

Potentially Shippable Increments at the end of each sprint as per DoD

Final release

MVP

(Minimum Viable Product)

External Releases:

TG3.2

Internal milestones:

TG5

1st release

2nd release

TG2

TG3.1

Speculate & Adapt

TG4

TG3

TG1

# Milestones and milestone management

## Project Milestones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Milestones | Gate | Definition | Phase | Planned date | Type |
| Budget approved for Speculation phase |  | Resource is assigned for Initiation phase to be completed | Envision |  | Tracking |
| Initial product backlog complete |  | Epics are defined including architecture and non-functional requirements at high level | Envision |  | Tracking |
| Envision phase  Complete\* | **TG1** | Stakeholders approve go-ahead to Initiation phase | Envision |  | Go/No Go |
| T-shirt sizing done |  | High level estimate is done (x4 for uncertainty) for budgeting purpose of the product backlog (at epics level) | Speculate |  | Tracking |
| Budget approved for completion of the project |  | Budget is estimated and agreed | Speculate |  | Tracking |
| High Level Architecture agreed |  | Including SW and HW architecture, using DDD [7] | Speculate |  | Tracking |
| Non-functional requirements (KPIs) defined |  | Performance, monitoring, load, reliability, stability, usability, Security, other “ilities” | Speculate |  | Tracking |
| Test Strategy defined |  | High level test strategy | Speculate |  | Tracking |
| Resource plan agreed |  | Resource plan agreed for the duration of the project | Speculate |  | Tracking |
| Speculate phase complete\* | **TG2** | Overall scope is defined and estimated, Resource available, team can start detailing the plans | Speculate |  | Go/No Go |
| Lift Off session complete |  | Team is aligned with a clear purpose and understand the context of the project. Team has defined their working “rules”. Project chartering is live | Speculate |  | Tracking\* |
| Definition of Done created by the team |  | Quality standards are set | Release Planning |  | Tracking |
| Product backlog is DEEP |  | Product backlog has been estimated and prioritised | Release Planning |  | Tracking |
| Release Planning complete\* | **TG3** | There is a high level plan covering the next 3 sprints Team is ready to sprint starting with Sprint Planning for sprint 1 | Release Planning |  | Go/No Go |
| End of Sprint 1 |  |  | Explore & Adapt |  | Tracking |
| End of Sprint 3\* | **TG3.1** | Review product progress to stakeholders | Explore & Adapt |  | Go/No Go |
| End of Sprint 6\* | **TG3.2** | Review product progress to stakeholders | Explore & Adapt |  | Go/No Go |
| Stable velocity\* |  | When velocity is known and can be used for planning purposes | Explore & Adapt |  | Tracking |
| First release to customers\* |  | First external release, not for commercial use: key features are: … | Explore & Adapt |  | Tracking |
| Second release to customers\* |  | Second external release, not for commercial use: key features are: … | Explore & Adapt |  | Tracking |
| MVP: Minimum Viable Product\* |  | Product that has enough features it can be deployed to end customers: key features are: … | Explore & Adapt |  | Tracking |
| Final release to customers\* |  | Final release, product is ready to go into maintenance mode | Explore & Adapt |  | Go/No Go |
| Speculate & Adapt phase complete\* | **TG4** | Last sprint is complete; project can be close | Explore & Adapt |  | Go/No Go |
| Closure phase complete\* | **TG5** | Project can be close; all pending actions are close | Close out |  | Go/No Go |

|  |  |
| --- | --- |
| TG | Tollgate (formal milestone for decision making) |
| Go / No Go | Formal review meetings; dates will be tracked |
| Tracking | Milestones will be tracked only i.e. no decision required |
| \* | Key milestones that will be tracked using milestone tracking tool |

## Milestone tracking

“Milestone Slip chart” tool [2] will be used to track milestones marked with ‘\*’ on a weekly or sprint based depending on the phase of the project (weekly for all except Explore and Adapt: at the end of each sprint)

## Example

# Risk and Issue management

Risks and Issues will be updated on a weekly basis and shared with all stakeholders

Risk and Issue tracking tool [3] will be used.

Example

# Assumption management

The project team members must identify and document all of the assumptions being made during the project planning process, and then on a one by one basis, identify the risks that exist as a result of each assumption to the project based on the potential inaccuracies or inconsistencies that the assumption may exhibit.

Assumptions will be managed as Risks using Risk and Tracking tool [3]

# Dependency management

Release planning sessions, sprint planning meetings and during sprints will reveal dependencies.

Release planning with other teams as well as scrum of scrums will help negotiate such dependencies (due date, type, urgency, etc)

Other dependencies will be managed in the table below:

|  |  |  |
| --- | --- | --- |
| Description | When by | Comments/Risks |
| Continuous Delivery Environment | From sprint 1 | Lack of an effective Continuous Delivery environment will slow down development and prevent us from having stable velocity |
| Stable platform | From sprint 1 | Unstable platform will mean more time to identify issues and will slow down the team |
| 3rd party… |  |  |
|  |  |  |

# Communication plan

|  |  |  |  |
| --- | --- | --- | --- |
| What | Description | When/Frequency | Who to: |
| Live demo of working software | Demo of what has been delivered according to the definition of Done at the end of the sprint | At the end of each sprint | All |
| Sprint reports | Velocity  Release burn-down chart  Risks and issues  Test results  Unit test coverage  Number of defects (inflow and outflow)  Technical debt | At the end of each sprint | Stakeholders |
| Toll Gates | For each gate, a meeting will be hold including last sprint report (if applicable) and a review of the business case.  Decision to continue (Go) or cancel (No Go) the project can be made at these meetings | See TG dates and MVP date | Stakeholders |
| Risks and Issues | Risks and Issues on the project | Weekly | Stakeholders |
| Milestone Tracking | Estimated milestone dates | At the end of each sprint | Stakeholders |
| Scrum of scrums | Rotating team members | Twice a week | Other team members from scrum teams |

## Minutes of meetings

For meetings with stakeholders, meeting minutes will be recorded using “Easy Minutes” [6]

# Quality Plan

## User Stories

User stories will:

* Follow the INVEST Model (Independent, Negotiable, Valuable, Estimable, Sized appropriately and Testable)
* Have personas
* Have conditions of satisfaction
* Have acceptance criteria using specification by example (Given/When/Then)
* Etc

This is not about matching the criteria above that makes a good user story, this is the quality and appropriate level of information. Hence we will also maintain a DEEP product backlog (Detailed Appropriately, Estimated, Emergent and Prioritised).

## Definition of Done (DoD)

### Creation of the DoD

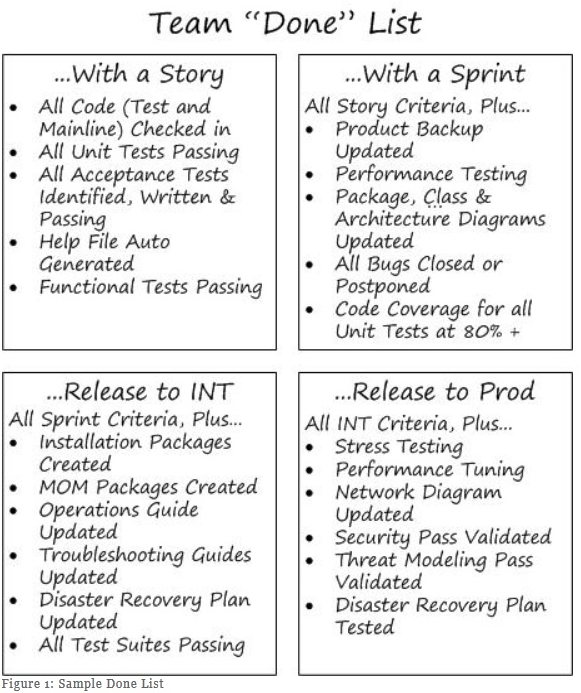
Team creates their “Definition of Done” for:

* User stories
* Sprints
* Release (to production)
* The DoD is approved by the PO and display on the wall

### Review of the DoD

The team will review their DoD at the end of each sprint

### Example



## Design and Testing

Design and testing are aligned with the user stories of the product backlog as per [7]

# Test Strategy

## The 4 quadrants

For test planning, we will base our planning on Agile Testing Quadrants [9] and keeping in mind the success of the project relies on a very collaboration between all members of the team. Testing is the responsibility of the team, not only to test engineers working in the team.



## The pyramid of tests

From UI Tests (hard to maintain, long to run) to Unit Tests (Easy to maintain, quick to run) [8].

**UI tests**

**Functional tests**

**Unit tests**

# Software framework

We will be using Scrum with (as a summary):

* Sprint Planning
* Daily stand-ups
* Sprint review
* Retrospectives

Sprints will be **?** weeks long. Each sprint will finish on Wednesdays 10am.

# Release Management

Sprint cycle and release cycle will be decoupled giving the team the opportunity to release software any time. In general, a release will be made at the end of each sprint where integration issues will be addressed.

# Defect Management

Defects will be adding to the product backlog (if they are not fixed within the sprint). Defects will be estimated in story points and prioritised among other defects and user stories.

No points will be given for fixing defects. Points are only for estimation purpose and will not be added to the velocity.

# Roles and Responsibilities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RACI Matrix | Functional Manager(s) | Scrum Master | Product Owner | Scrum Team | Project Manager |
| Ensure consistency of scrum practices across teams | I | C | C | I | R/A |
| Provide vision and goal for the product | I | I | R/A | I | I |
| Provide resource with right skills and mindset | R/A | I | I | C/I | C |
| Prioritize and manage the product backlog | I | F | R/A | C | F |
| Remove impediments | R | R | R/A | R | R |
| Manage the implementation of the project plan | I | I | C | C | R/A |
| Make sure scrum practices are used and improved within the team | R | R/A | C | R | F |
| Create, apply and continuously improve the Definition of Done | C | F | R | R/A | F |
| On time reporting to management | I | F | R/A | I | F |
| Define acceptance criteria | I | F | R/A | C | F |
| Write acceptance tests | I | F | C | R/A | F |
| Ensure quality of the product | R | R | R/A | R | R |
| Manage Risks | C | C | R/A | C | R |
| Approve user stories (user stories meet the acceptance criteria) | I | F | R/A | C | F |
| Decide on release date and goal | I | I | R/A | I | I |

Note:

1. The above RACI matrix doesn’t cover all the activities within the scrum framework; therefore always check the responsibilities for each role.
2. The RACI matrix may differ per project due to structural and/or organizational constraints.

Responsible =

Those who do the work to achieve the task. There is typically one role with a participation type of *responsible*, although others can be delegated to assist in the work required

Accountable =

The one ultimately answerable for the correct and thorough completion of the deliverable or task, and the one from whom *responsible* is delegated the work. In other words, an *accountable* must sign off (approve) on work that *responsible* provides. There **must** be only one *accountable* specified for each task or deliverable.

Consulted =

Those whose opinions are sought, typically [subject matter experts](http://en.wikipedia.org/wiki/Subject_matter_expert); and with whom there is two-way communication.

Informed =

Those who are kept up-to-date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

Facilitator =

Helps teams and individuals to continuously improve and understand their roles within the Scrum framework. They help team members change their behaviour and act as a coach and a change agent.

# Resource Management and Planning

% time allocated and phase



# Budget

## People

Phase duration (weeks) \* Total FTE / 4 weeks (per month): 146.5 man/weeks = 37 man/months

Cost per team member: 50k

Total cost = 50\*37 = 1850k

## Software

## Hardware

Hardware System requirements will be captured under [1].

The cost will be provided:

## Trainings

## Travels

## Miscellaneous

## Summary

|  |  |
| --- | --- |
| Type | Estimated Cost (£k) |
| People | 1850 |
| Software |  |
| Hardware |  |
| Trainings |  |
| Travels |  |
| Miscellaneous |  |
| Total |  |

# KPIs (Project)

## Team Maturity

We will update the checklist [4] on a monthly basis.

## Metrics

Velocity will be recorded on a sprint basis

# Compliance constraints

# Reference

|  |  |  |
| --- | --- | --- |
| Description | ID | Link/File |
| Hardware System Requirements | [1] |  |
| Milestone Slip Chart tool | [2] |  |
| Risk and Issue Management tool | [3] |  |
| Scrum Checklist | [4] |  |
| Lift-off: Launching Agile Teams & Projects | [5] | Link: [here](http://www.amazon.co.uk/Liftoff-Launching-Agile-Teams-Projects/dp/097792016X) |
| Easy Minutes tool  (check where to store Add-In templates with your version of Windows Microsoft) | [6] |  |
| Aligning user stories, design and testing | [7] |  |
| The forgotten layer of the test automation pyramid | [8] | [the-forgotten-layer-of-the-test-automation-pyramid](http://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid) |
| Agile Test Planning | [9] | [Agile Testing Planning link](http://lisacrispin.com/downloads/AdpTestPlanning.pdf) |