

symb<sup>o</sup>x<sup>®</sup>

# AGILE DELIVERY

*a practical guide*

The aim of this document is to outline the **practical** experience gained from our agile software deliveries so we can use it as a **guide** for future projects.

Please note that this is not to replace any **self learning** and/or **research** on 'agile delivery'. The purpose is simply to introduce **harmonisation** across teams.



To avoid complexity, the SCRUM methodology has been chosen for this guide.



SCRUM is one of the simpler methodologies to implement and hence it's a good start



This is a must read for SCRUM masters *AND* new joiners who work within agile teams



The key stages defined in this document such as planning, show & tell and retro **MUST** be followed by all. SCRUM masters can, however, diversify from using the recommended tools if/when required





# PLANNING & ESTABLISHMENT

Initiate each project with an **establishment** phase, the duration of this phase really depends on the scope of your project. Remember you are taking your customer on a journey and this is the first step. We are calling it establishment because in addition to the backlog you are also establishing vision, expectation, value, transparency.

## *Collaborate*

with the customer to create, define and shape the backlog

## *The customer*

is the ultimate owner of the backlog, therefore don't turn this into a silo activity. User stories and acceptance criteria have to be defined collectively

## *Define user stories*

Using a business language and not technical language. A generic format can be: "As a ... [define your personas].. I would like to .. so that I can achieve ...."

## *Try and avoid*

estimation at this stage. If you need to estimate, make sure everyone is aware that these are rough estimations

## *By the end of this phase*

you should know the prioritisation for sprint 1 and have a set of stories that meet the definition of 'Ready'

## *Avoid*

creating hefty documents. User stories and wireframes should suffice

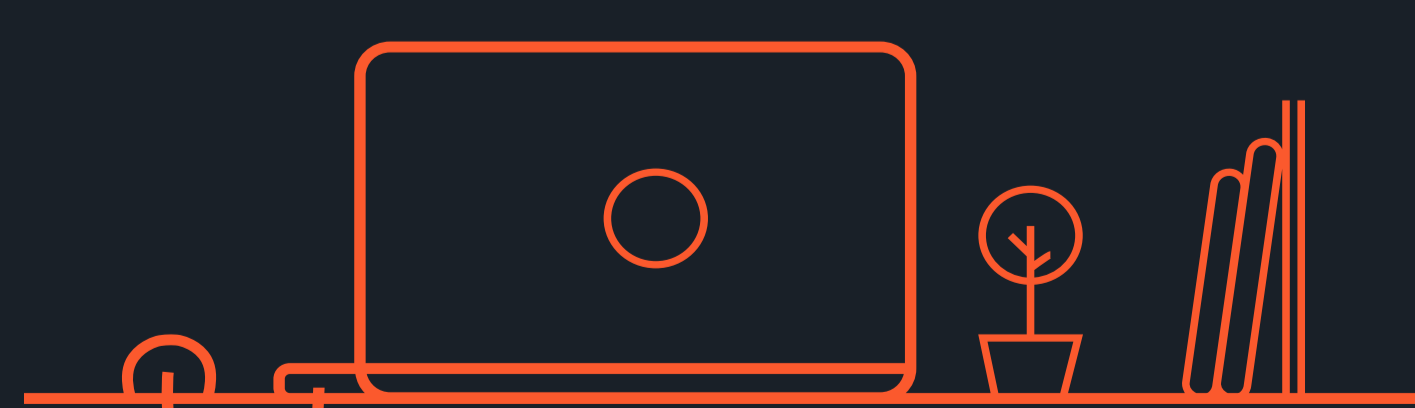
## *Set target velocity*

based on the size of the backlog and size of your team

### NOTES

Its always better to start with a lower velocity and increase the velocity overtime if required.

The backlog is a living artefact, it will change and improve as you go along



### SUGGESTED TOOL: JIRA

Your chosen tool will be the centralised repository for your back log, sprint scope





# THE SCRUM TEAM

## *symbox team*

dev / config  
team



scrum  
master



## *customer / client team*

qa / test  
manager



product  
owner



### SCRUM MASTER

Facilitate Scrum Meetings, sprint planning, retrospectives and show and tells

Team performance and metrics

Team coaching and coordination

### DEV TEAM (PS CONSULTANT)

Estimation

Translation of business driven backlog items to logical units of work  
Implementation of backlog items

Creation of unit tests and stubs

### DEV TEAM (QA)

Creation of test plans

Estimation

Prevents defects from entering the release

Continually integrating the API code based with automated API regression tests

### PRODUCT OWNER

Creating and maintaining the product backlog

Owning the product vision and roadmap

Defining user stories and acceptance criteria

Defining sprint goals

Prioritising product backlog as part of sprint planning

Daily stand ups

### QA MANAGER

SIT Testing

UAT Testing

# SPRINT PLANNING



## Lock in the scope

Start every sprint with one day of planning. At the end of planning, the **scope** for that sprint should be **locked in**



## Prioritise the requirements

Prior to sprint planning, the product owner must have already **prioritised** the requirements to the top of the backlog, these requirements should all meet the definition of **Ready**. For sprint 1, this is done as part of the establishment phase, for any consecutive sprints this is done during the prior sprint and in parallel to delivery



## Break it Down

Break down the user stories to as many **sub tasks** as possible, this way we narrow down the smallest piece of business value and tremendously reduce the risk of overall failure



## Get everyone's input

Ensure your product owner and each and every individual within your scrum team attends this session, you will need everyone's input



## Clairty of Business Value

Review and **refine prioritised user stories**, ensure every one in the team is clear and understands the purpose and the **BUSINESS VALUE** of the story



## Collaborative Estimations

**Estimate** the requirements in order of prioritisation. Every team member **MUST** take part in this estimation. Once the estimation is done, identify how many requirements can be delivered in the sprint - this is to be done based on target/expected velocity



## SUGGESTED TOOL: PlanITPoker

For remote teams use PlanITPoker, for co-located teams, use Poker cards



# SPRINT DELIVERY



## daily stand ups

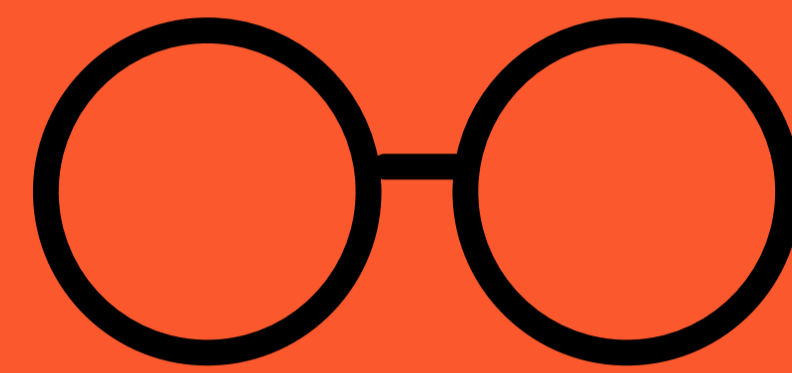


Setup **daily stand ups** with the product owner and the team

Purpose is to review progress, agree actions to remediate issues or risks and clarify design queries

Be clear on how you deal with **scope change** in the **middle** of sprints. If one of your user stories is changed, are you going to drop it from the current sprint and if not how are you incorporating the change without impacting velocity? Make these decisions collectively with the team and the product owner. This is a team sport so make decisions as a team

*Transparency is key*  
customers DO appreciate transparency



*Ensure good communication*

flows within the team, in particular the handover between developers and QA is going smoothly

Ensure everyone is clear on the terminologies and processes you are using such as the definition of **'Done'** and the internal **'Development Process'**

Note: If in doubt ask questions and raise concerns. If there are issues with completing your tasks then raise them ASAP. This will help the scrum master with setting expectation with the customers.

**IMPORTANT POINT!**

The shorter the sprint duration is the more significant the daily stand-ups become.



## BEST PRACTICE

Work with two environments, one for build and one for QA

Get the developers to deploy every feature to the QA environment when ready for testing. Ideally, automate this and use that same automation for Production Deployment too, in order to achieve consistency





# SPRINT SHOW AND TELL

This is a **demo** to all customer **stakeholders** (product owner and more importantly the user group) showing the outcome of the sprint. Demos are integral to the sprint success as they provide the customer with the precise idea of how the system works

## *Last working day*

of every sprint before the application is deployed on the UAT / SIT / ORT / PROD environment

## *It's not unusual*

for the customer to have a change of mind about some requirements during the show & tell. Be clear about the fact that changes should be added to the backlog and prioritised by the product owner

## *Do*

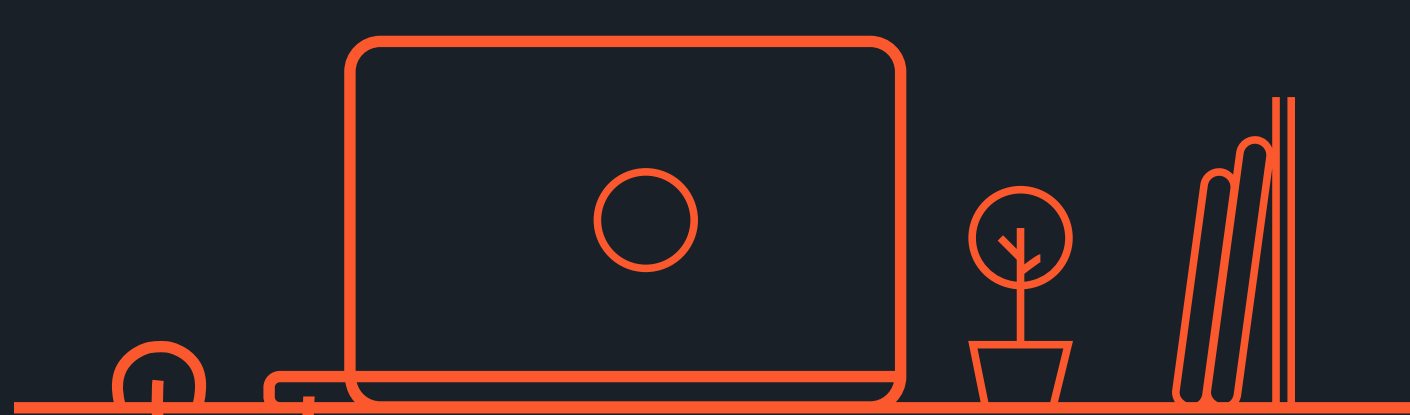
agree during Demo whether a Story has met the Definition of Done or not

## *Don't get frustrated*

by the change requests, remember this is why we are doing agile in the first place. To deliver what works best for the users!

### NOTES

Review your user stories and slide decks at the end of each sprint to ensure they are updated.



**SUGGESTED TOOL**  
**Demo the real working system**





# THE RETRO



Scrum retrospectives should be scheduled at the end of each sprint, after Demo (feedback) and before Planning (decisions). That way you can make sure to use the learnings of a previous Sprint in the next

The team as a whole decides on which measures to take on and which ones are most important. The Scrum Master's and Product Owner's votes count the same as anyone else's, although those are the two roles mainly responsible for executing the actions

The aim is for everyone to share what has gone well, what has not gone well and suggest behaviours, processes and tools that they believe will improve overall team output

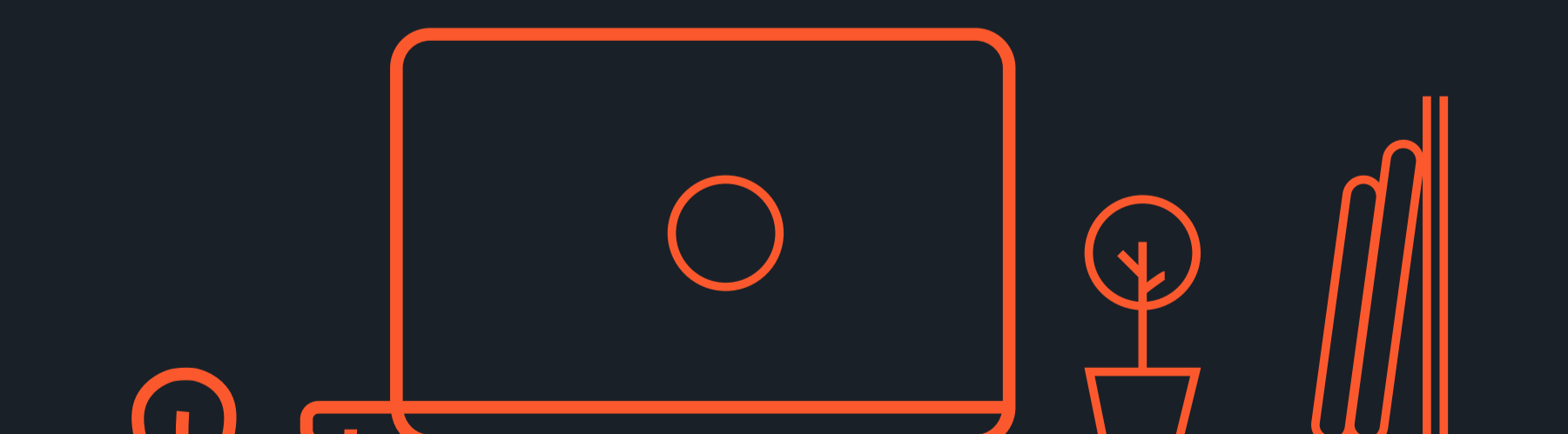
The aim is to ensure the team is continuously learning and improving its performance by slowly removing every impediment that is preventing them from reaching their full potential

Similar to the sprint planning sessions, retros are recommended to be attended by both product owner and every team member within the scrum team



*Repeat the cycle*

... go back to planning for the next sprint



**SUGGESTED TOOL: TRELLO**

or a simple excel sheet to keep track of your actions/mitigations



# KEY DEFINITIONS



## ready

An agreement between the team and the Product Owner that defines the basic requirements for acceptability. This is generally based around what the team expects to ensure quality delivery, like having wireframes for UX changes and clearly understandable Acceptance Criteria for everything else.

It is both Symbox team's responsibility to call out any requirement that does not meet the Definition of Ready and block it from prioritisation into a Sprint. It is the Product Owner's responsibility to meet this agreement, but it is the Scrum Master's responsibility to help the Product Owner achieve it in any way possible.

## done

An agreement between the team and the Product Owner that defines at which point a Story can be considered Done. More than often, done is assigned to a Story that meets all of its Acceptance Criteria and is capable of delivering the Business Value.

## development process

An agreement between all members of the Scrum that defines how the team will work with each other (UX, Dev, QA) to ensure that it reliably and consistently delivers business value with the lowest chance of failure.





# 12 FUNDAMENTAL *agile principles*

Based on the Agile Manifesto

01

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

02

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

03

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

04

Business people and developers must work together daily throughout the project.

05

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

06

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

07

Working software is the primary measure of progress.

08

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

09

Continuous attention to technical excellence and good design enhances agility.

10

Simplicity--the art of maximizing the amount of work not done--is essential.

11

The best architectures, requirements, and designs emerge from self-organizing teams.

12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.