Cheat Sheet

# Cheat sheet

| | Story Points |
|---|---|
| **Velocity** | Amount of work, usually in SP, that the team delivers within a Sprint |
| **WIP** | Work In Progress |
| **Lead Time** | Total time from when a task is requested until the task is Done |
| **Cycle Time** | Time from when the team starts working on a task and the task is Done |
| **Ageing** | Time a task is in a specific status |
| **Throughput** | Number of tasks that the team gets to Done in a period of time |
| **SLA** | Service Level Agreement - amount of time agreed to deliver |
| **Backlog** | List of tasks that needs to be done<br>Organised by priority |
| **Board** | Visual tool that shows the status of the tasks<br>The basic one is: To Do, In Progress, Done |
| **Burnup Chart** | Visual tool that show the completed work vs the work planned in a period of time<br>Highlights how much remains to be done |

Discarded

K 7

Scrum

Agile Management Minor 2024

## Scrum

What we know:

- Velocity average = 11 SP (Story Points)
- Sprints of 2 weeks

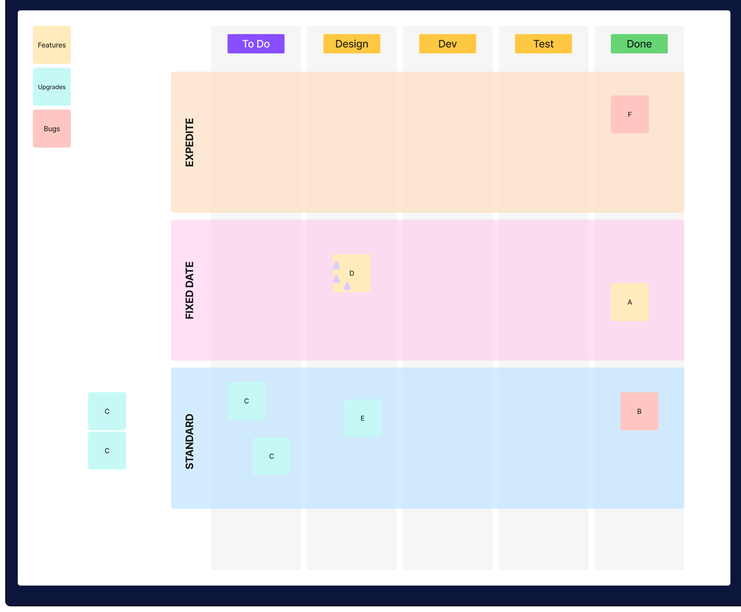
Kanban

Agile Management Minor 2024

## Kanban

What we know:

- SLA of Features = 2 weeks
- SLA of Bugs Prio 1 (critical) = 2 days




Scrum Estimations

## Estimations

Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987

Each number is the sum of the previous two numbers.

### Fibonacci

In scrum teams usually use:
1, 2, 3, 5, 8, 13, 21

### Why?

- Small tasks are easier to estimate, but bigger, are harder. The gaps between Fibonacci numbers helps to reflect that uncertainty
- Encourage the discussion. For example, if a task is more than a 5 but not really an 8. This improves the understanding of the task to be done
- Faster to estimate. Instead of losing time if is an 5 or a 6, team picks the closest number

Scrum Estimations

## Example

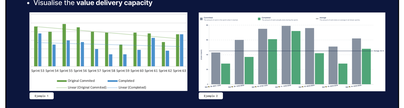
🍇 1
🍌 2
🥥 13
🍎 5
🍍 8

Scrum Metrics

## Measure

Commitment vs Completion

- Visualise the **value delivery capacity**





**Burn Up/Down chart**

- Visualise flow health

**Product Metrics**

- Visualise value impact
- Example: number of active users, time expended on the app

Classes of Services

## Classes of Services

Categories that are used to prioritise the work that enters in the system
Help us to decide how we manage the work

The most common ones

- **Expedite:** high priority. Can skip the queue
- **Fixed date:** must be done in an specific date
- **Standard:** normal work, manage in order as they enter the system
- **Intangible:** low priority

Kanban Metrics

## Measure

**Ageing**

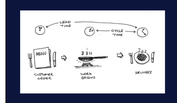
- Time the ticket is in a column (specific status)

**Throughput**

- Amount of tickets done on period of time

**Lead time and Cycle time**

- Amount of tickets done on period of time



**Cumulative Flow Diagram**

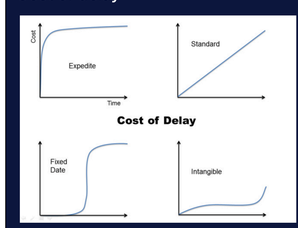
- Visualise how the work is "moving" on the different status over time

Burnup

- Which one you consider is the "healthiest" one?
- Which one you consider is the "un-healthiest" one?
- Meaning of: red line, green line and changes on each one of them

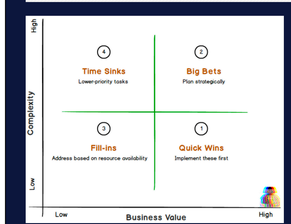
Cost of Delay

## Cost of delay




Helps to measure the impact of not delivering a task or feature sooner

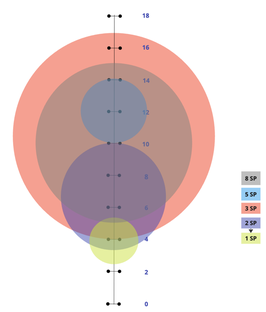
Prioritisation




Estimations

Story points behaviour vs days

1 SP 4 -> [2.5;4]
2 SP 5.5 -> [3;10]
3 SP 8.5 -> [4;17]
5 SP 9 -> [10.4;14]
8 SP 12 -> [5.5;15]




Common tools

## Tools

Jira
- Free up to 10 users

Trello
- Free with premium options

Asana
- Free with premium options

Notion
- Free with premium options