

Scrum Training

The Pragmatic Guide

Agenda

Introduction

New Tools

Team Agreements & Expectations

Q&A

INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS

WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION

RESPONDING TO CHANGE OVER FOLLOWING A PLAN

While there is value in the items on the right, we value the items on the left more.

What is Scrum?

A simple framework for effective team collaboration on complex projects. It provides a small set of rules that create just enough structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.

Purpose of Scrum at YouLend



Framework

Providing a simple framework for managing teams expectations & deadlines



Focus

Define clear and achievable goals for the Scrum team through a prioritized backlog and Sprint Goal



Empower Teams

Make everyone better, Encourage, support and enable the team to reach their full potential and abilities

Core Scrum Elements

The Team

The Dedicated cross functional team that works together to get the Sprint Backlog to done. Team members agree on optimal ways of working

The Sprint (Iteration)

A time-boxed period, during which a dedicated team works to complete a set amount of work from the backlog

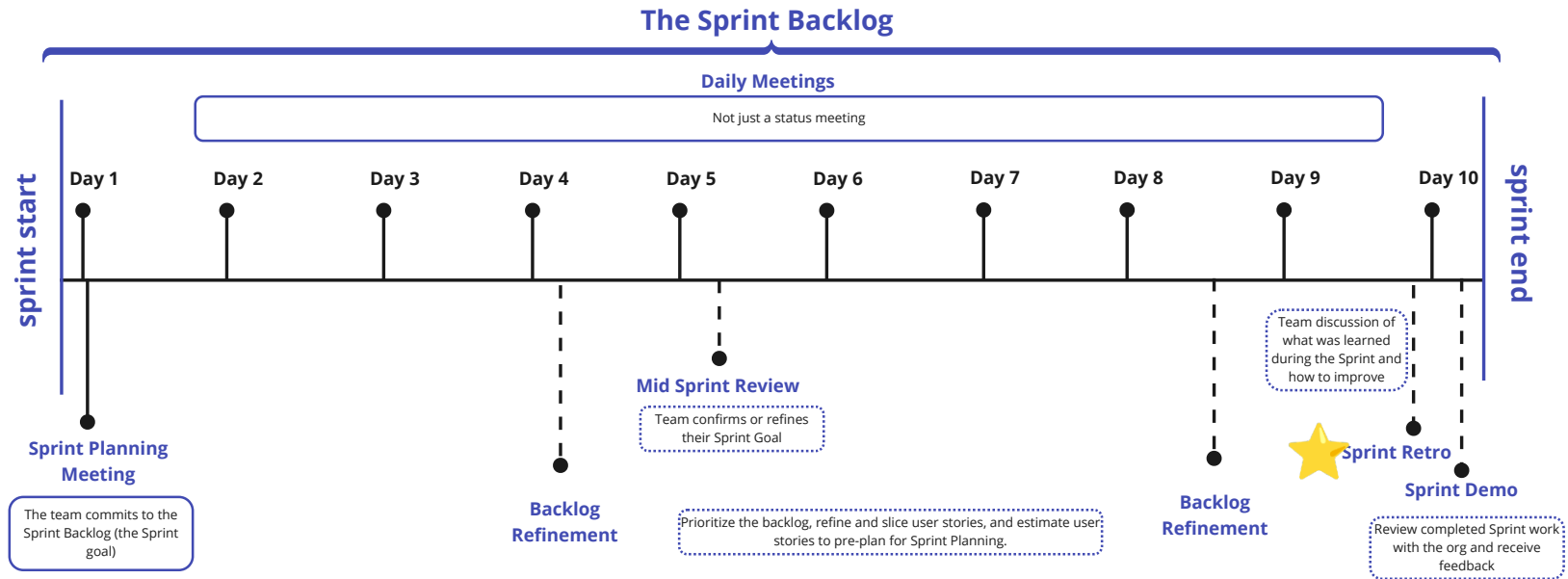
The Backlog

A prioritized list of tasks, features, and requirements, awaiting development, serving as an, evolving plan for delivering the product

The Sprint Backlog (Goal)

The Product Backlog items that the Screen Team agreed to complete within the Sprint

Scrum Lifecycle



Core Scrum Meetings

Sprint Planning

The purpose is for the scrum team to agreed to an achievable Sprint goal. Together the team agrees to this goal

Daily Meetings



Your daily meeting changes. Each team member discusses their status in the context of the Sprint goal. The whole team consider the Sprint goal and if they need help or can help team members meet the goal. Scope changes should be discussed daily.

Mid-Sprint Review

Reaffirm an achievable Sprint goal. Ensure that we minimize work in progress and minimize story rollover

Refinement meetings



Make sure that we have a backlog of appropriately sliced and clearly defined stories. Well defined stories are the key to a successful scrum process

How we organize our work
helps to facilitate the rest of the
process

Backlog Heirarchy

Epics

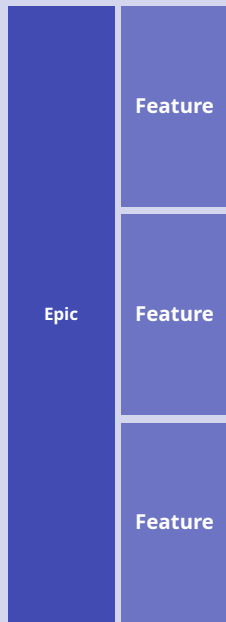
Epic

A large body of work that can be broken down into smaller “pieces” (Features). Normally represents the **WHY** of a project and its high-level objectives.

Epics **may span months or even years** and can require continuous improvements.

VRP, FFL, Production Line 2.0 and Amazon integration are all examples of Epics

Features



A unit of work that delivers business value and contributes to the objective of an Epic. Focuses on **What** we're building

Features can **span multiple sprints, taking weeks or months** to complete, and have a clearly **defined endpoint**.

Open Banking Domain, Add language to leads and Lender Allocation are examples of features

User Stories



Generally restricted to a smaller code and well-defined area. Often, they represent incremental value the team must deliver to create a Feature, but it is also possible for a user story to directly deliver business value. The main focus of a User Story is on the **How** of a general project. They have acceptance criteria that defines the stories outcomes.

User stories are typically completed within a **single sprint** by a **Scrum team** and generally take **less than two weeks** to finish.

New metric in credit risk, create a landing page for a partner, add a new API endpoint, Automatically mark KYC as completed are examples of user stories

User Stories are the body of work that are taken into a Sprint

Subtasks



Subtasks are reserved for the team members who are executing the work on a user story, serving as a means to break the story into manageable parts for clear progress tracking throughout the sprint. They help refine the **How** by focusing on specific, well-defined areas or sections of code.

Subtasks normally take **less than 1 day to complete**

Examples: Update test case, write documentation, install package, test dashboard

Subtasks are done by team members to complete User Stories.

User Stories - INVEST

Stories should be "full stack" and include UI+API+ET+Domain

Independent

Stories shouldn't depend on others, allowing them to be developed, or prioritized in any sequence

Negotiable

A Story is a conversation, not a set contract.

Valuable

Each ~~Story~~ Feature must provide clear, tangible value to the end-user or business.

Estimatable

Enough information to estimate the size or complexity of the work. Be consistent

Small

Small enough to be completed within a single sprint

Testable

Clear ACs, allowing the team to verify it is "done" and functional.



Testability is the most important quality of a user story

How we estimate effects our ability
to plan work

Estimating

Story Points

We estimate User Stories using Story Points

Story points help differentiate between interrupted and uninterrupted time

Any work outside the sprint or on unpointed tasks is considered interrupted time

Fibonacci

**The Fibonacci Sequence will be used to
estimate user stories**

1, 2, 3, 5, 8, 13, 21

- How much work it is
(including QA & bug
fixing)
- How difficult is it
- Consider Unknowns

Estimating

Relative

- People are better at the relative comparison
- Consider effort level in relation to other stories
- Includes complexity, effort and risk



Self Correcting

- Aim for consistency, not accuracy
- Velocity is the greater equalizer
- Velocity corrects estimation errors



Collaborative

- Multiple minds are better than a single mind
- **Share the stories ahead of time**
- **Team discussion about how the story will be tested**



Working with Story points is highly dependent on appropriately sliced user stories

The team should collectively discuss and reach consensus on specific guidelines and agreements

Agreement

Guidelines



Definition of Done

Establish a shared understanding of what it means for a user story to be completed to ensure that the work item is **truly finished**. Try to avoid dependencies outside of the team. You may have multiple definitions of done, depending on the type of work. **Be consistent!**

Sprint Boundaries

Set the Sprint start and end to times to land precisely at the Sprint Planning meeting.

Respect the Sprint planning meeting.

Subtasks

Subtasks not only breaks down complex tasks into manageable parts and improve transparency but also enables multiple team members to work on different aspects of a larger task simultaneously **fostering team work**. Subtasks facilities the movement to **shift left**.

Respect The Sprint

Do not add points to the sprint without discussion and removing the equivalent number of points if needed to ensure you still meet your Sprint goal.

During the early days of Scrum, we should avoid adding unpointed or interrupted work because it will distract from completing the Sprint goal.

Standards & Expectations

Interrupted Time

Agree to the work that is considered inside and work that is outside the Sprint. Any work outside the sprint or on unpointed tasks is considered interrupted time and could **impact meeting the the sprint goal**

Engage

the Scrum framework relies on frequent communication and feedback loops among team members. Engagement at all levels is important, from daily meetings to sprint planning and refinement sessions to 1:1 or group team meetings. Aim to have your camera on and engage. **Celebrate your team Wins together!**

Get Stories To Done

Even within the Sprint, *Work in progress is a liability*, we work together and establish plans and action items to get Stories Done.

Team Work

Working together as a team is crucial because it leverages collective skills and knowledge to efficiently tackle complex tasks, fostering innovation and enabling faster problem-solving.

Getting to know your new
tools

Tools

The Scrum Board

The screenshot shows a Jira Scrum Board interface. At the top, the 'Taskboard' tab is selected and circled with a red circle and a callout '1'. The board has columns labeled 'Open', 'Doing', 'Dev done', and 'PR'. A callout '2' points to the first column 'Open'. A callout '3' points to the 'Doing' column. A callout '4' points to the 'Sprint 3' selector in the top right. A callout '5' points to the 'Person: Ali' selector in the top right. The board contains several tickets, including '25165 Financial info page re-design (release: 21 Feb)' and '25351 Automation testing for plaid'.

You'll use this board to manage your Sprints

1. Swim Lanes

Similar to your Kanban board except you move subtasks instead of stories through the swim lanes. Make sure to **keep the ticket and subtask statuses updated**.

2. User Stories & Bugs

This is the parent ticket which holds the business requirements. It gets the story points and is the body of the work the team **works to get done**. Assigning parent tickets gives people ownership but make sure it doesn't stop them from working together as a team.

3. Subtasks

ADO forces you to break stories into sub tasks. Through this training, you'll learn their value. They make the **work more visible**. Include QA tasks to expose the shift left. You do NOT point subtasks.

4. Sprint Selector

Typically, you select the active Sprint

5. Person Selector

A convenient way to find the work for an individual.

The Backlog

The screenshot shows the Jubilee Backlog interface. Callout 1 points to the 'Story Points' column header. Callout 2 points to the 'Jubilee Sprint 4' dropdown menu. Callout 3 points to the 'Jubilee Sprint 5' entry in the sprint list. Callout 4 points to the '+ New Sprint' button at the bottom.

Order	Title	State	Assigned To	Story Points
1	Bug fix for language selector	Closed	Lavinia Dimit...	
2	Enable France as a geography [release: 11 Mar]	Active	Ben Roberts	8
3	Add language selector to merchant website [release: 22 Feb]	Closed	Lavinia Dimit...	5
4	New lending partner: Tpay (Poland) [release: 22 Feb]	Closed	Lucas Teixeira	2
5	Sodexo rebranding to Pluxee - changes on merchant website + Notifications [release: 22 Feb]	Closed	Andrew Lewell	3
6	New lending partner: Grubhub (US) [release: 28 Feb]	Closed	Ben Roberts	3
7	New lending partner: secupay (Germany) [release: 28 Feb]	Closed	Lucas Teixeira	2
8	Update partnershipname + logo for eBay (UK) [release: 28 Feb]	Closed	Praise Hart	1
9	New lending partner: Dibble (US) [release: 7 Mar]	New	Lucas Teixeira	2
10	Add website field on company details page [release: 5 Mar - daytime]	Closed	Andrew Lewell	3
11	Fix scrolling issue	Closed	Andrew Lewell	3
12	Unit tests	Doing		2

Planning
Drag and drop work items to include them in a sprint.

Jubilee Backlog

Jubilee Sprint 4 (Current) 2/22/2024 - 3/6/2024
Planned Effort: 33
10 working days
10 6 2 45

Jubilee Sprint 5 3/7/2024 - 3/20/2024
Planned Effort: 13
10 working days
13 1 1 5

Jubilee Sprint 6 3/21/2024 - 4/3/2024
Planned Effort: 6
8 working days
4 4 1

Jubilee Sprint 7 4/4/2024 - 4/17/2024
10 working days
No work scheduled yet

Jubilee Sprint 8 4/18/2024 - 5/1/2024
10 working days
No work scheduled yet

Jubilee Sprint 9 5/2/2024 - 5/15/2024
10 working days
No work scheduled yet

+ New Sprint

The Planning View - Another handy view of the work

1. Story Points

Add the story points column so that you can make sure that all **work taken into the sprint is pointed**.

2. Sprint Selector

You can easily see the work scheduled for the current and future Sprints. You drag and drop work items from Sprint to Sprint to plan future work. This will be handy as you progress and starting planning.

3. Planned Effort

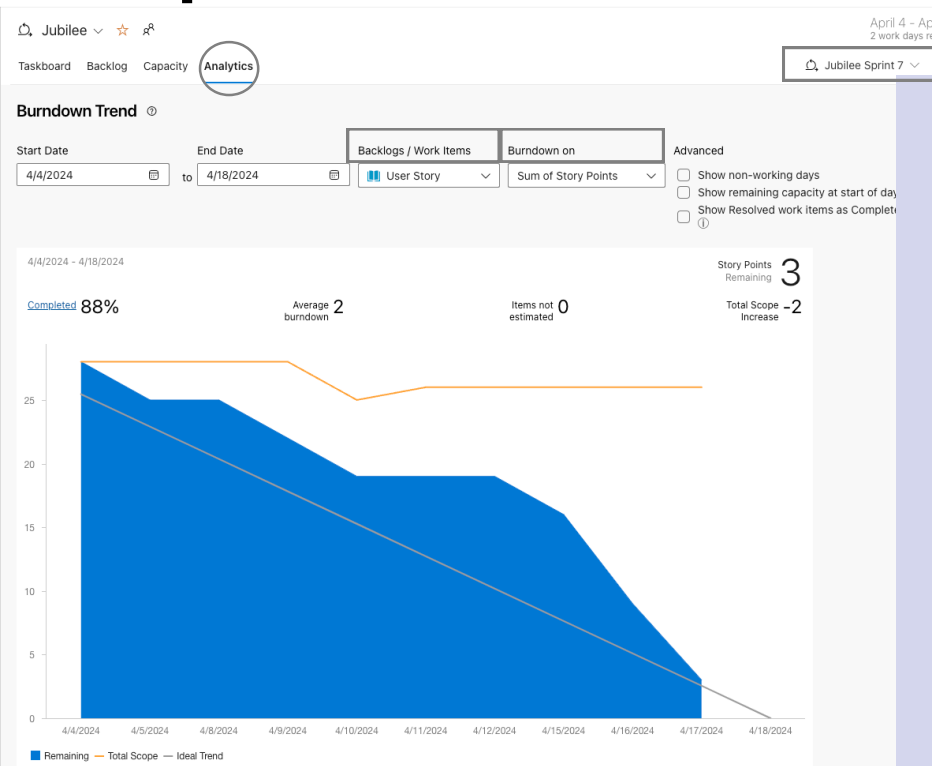
You can see the total story planned for each Sprint. Monitor this closely for the active Sprint.

4. Creating Sprints for your team.

Individual teams manage their sprint creation, facilitating customized sprint cycles.



Sprint Burndown



Monitor your progress

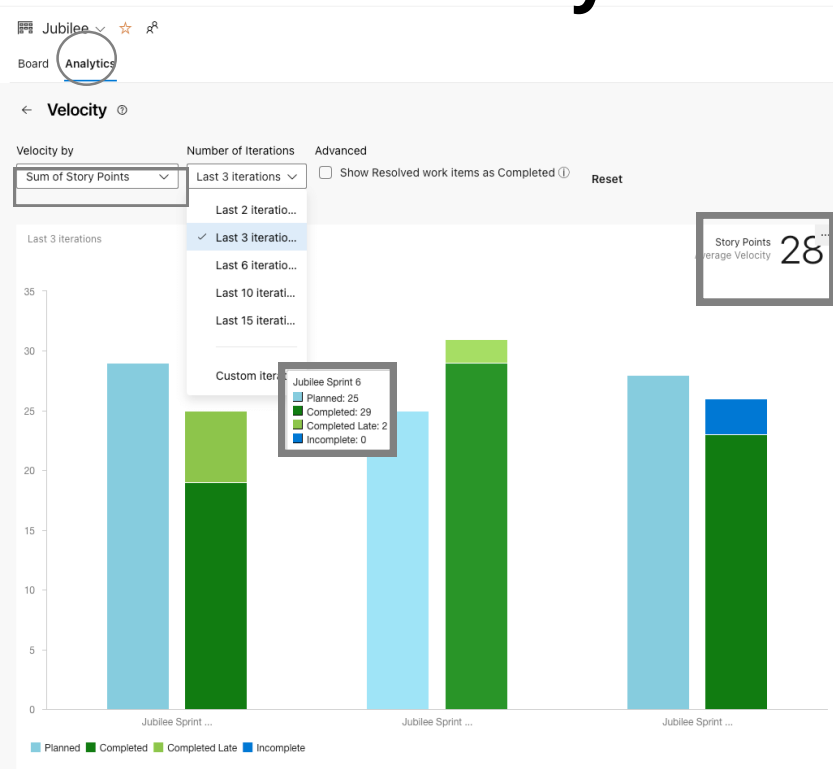
Its finicky and the default values are not helpful

You can find burn down chart under Analytics

- Select the current Sprint
- Choose User Story
- Burn-down on Sum of Story points
- Hover over a date to see the burn down for that date

Ideal burndowns are hard to achieve, but this is a healthy burndown

Team Velocity



Track your team's progress across sprints

Available from the main team board

- Navigate back to the main team board
- Choose Analytics
- Then Velocity
- View full report
- Choose Velocity by Sum of Story Points
- Select your last X Sprints

Planned vs Completed

Focus in what you planned vs what you completed

Average Velocity

We'll use this to plan sprints after you gave a few good sprints complete.

Q&A

One of my teammates is on leave for the entire Sprint what do we do?

- A Tell them to have fun
- B Publicly chastise them for ruining the Sprint
- C The whole team should go on Holiday together
- D Reduce the Sprint points based on the percentage of the team on leave

At the Sprint Planning meeting a Story from the ending Sprint is not done

A

Move it to the backlog

B

Blame the Business Owner

C

Move it to the new sprint with an action plan to get it done

D

Release it anyway

A story rolled over from the Previous Sprint how do we account for the points

- A** Take them all in their original Sprint
- B** Take no credit at all for the story in any Sprint
- C** Repoint it in the new sprint for the work remaining
- D** Take the full number of points in the sprint it gets completed

What if a I run out of work before the sprint ends

- A Take the rest of the Sprint off
- B Exhaust all methods to help your teammates complete their stories
- C Review candidate stories for the next Sprint to get a head start
- D Draw from an already groomed and prioritized story in the next sprint

Our team is responsible for regular maintenance work how does this fit into Scrum?

A

Just stop doing the maintenance work

B

Assign all the maintenance work to your Team Lead

C

Handle it as interrupted time

D

Allocate a fixed number of points each Sprint for the work

Its Mid-sprint and a critical issue is found and assigned to your team

A

Refuse to take the work

B

Verify the criticality of the bug

C

Review the open stories to see if there's extra capacity in the Sprint

D

Remove an un-started story from the Sprint

A team member gets sick in the middle of the Sprint, what do we do?

- A Do nothing
- B Determine the priority of their work in progress
- C Move their story out of the Sprint
- D As a team, try to take over their work

You discover a new edge case for a story in the current Sprint

- A Ignore it
- B Increase the story points to reflect the increased complexity
- C Slice it out as a separate story and put it in the next Sprint
- D Move the entire ticket into the next Sprint while you re-evaluate the design plan

The Sprint is starting and you don't have the all the requirements for a story but it has a tight deadline

- A Move the ticket to the following Sprint
- B Take it into the Sprint and make up the requirements
- C Move the ticket to the backlog
- D Delete the ticket

We have complex architecture work that can't be done in one Sprint

- A Force it into one Sprint and do a crappy job
- B Allow the story to span multiple Sprints
- C Try to slice it into smaller testable outcomes
- D Create a spike story to analyze and plan the work in smaller bits

There's a large story in the Sprint, but its hard to have multiple devs in that part of the code

- A** Let one team member struggle on their own to get it done
- B** Let the story roll across multiple sprints
- C** Shift left with testing and get an early start on testing
- D** Shift left with peer review and pair program on complex bits

A story is blocked waiting for a peer review what do you do?

- A Move the story out of the Sprint
- B Tell your team lead to work faster
- C Ask your another team member for help
- D Shift left and start testing your work

We just finished a big project, successfully, the team worked hard but had to take some shortcuts, what do we do now?

- A** Take a long holiday
- B** Take a smaller sprint goal for the next Sprint or 2
- C** Allocate one or more Sprints to bugs fixes and paying back tech debt
- D** Monitor your velocity to make sure its not impacted