CSE5PM
Workshop 3

Semester 1, 2024

What is Scrum?

Scrum is an agile project management framework that helps teams' structure and manage their work.

While the scrum is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular.

Scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.

Agile vs. Scrum

People often think scrum and agile are the same thing because scrum is centered around continuous improvement, which is a core principle of agile. However, scrum is a framework for getting work done, whereas agile is a philosophy.

Scrum Roles

A scrum team's size is typically small, at around 10 people, but it's large enough to complete a substantial amount of work within a sprint.

There are three main roles in every Scrum project:

- Product Owner
- Scrum Master
- Scrum team member.

Scrum Roles - Product Owner

Product Owners are focused on understanding business, customer, and market requirements, then prioritising the work to be done accordingly.

Effective product owners:

- Closely partner with the business and the team to ensure everyone understands the work items in the product backlog.
- Give the team clear guidance on which features to deliver next.
- Decide when to ship the product/feature.

Some examples of Product Owner you'll likely interact with during your course (specifically during the Capstone) are:

- Industry Project clients
- Lecturer
- Tutors
- Project supervisor

Scrum Roles - Scrum Master

The Scrum Master is the person who makes sure the project is rolling on schedule by working with the team to resolve blockers, identifies members' needs, and supplies them with the necessary resources.

Scrum Masters are responsible for:

- teaching and making sure each member understands Agile and their role in the development
- implementing the scrum process
- ensuring the members are completing their assigned tasks.

Scrum Roles - Scrum team

Scrum teams get the work done.

Strong scrum teams are self-organizing and approach their projects with a clear ‘we’ attitude.

All members of the team help one another to ensure a successful sprint completion.

Each member must truly commit to the project’s success.

Additionally, everyone needs to listen, help each other fix their problems and add values to the product. The more involved and committed the members, the more values they can add.

Scrum Artifacts

Scrum artifacts are important information used by the scrum team that helps define the product and what work to be done to create the product.

Three artifacts in scrum:

- product backlog (what all tasks to do)
- a sprint backlog (what all tasks to take on now)
- product increment (what gets “done”)

Scrum Artifacts – Product backlog

The product backlog is a list of new features, enhancements, bug fixes, tasks, or work requirements needed to build a product. It's compiled from input sources like customers, competitor analysis, market demands, and general business analysis.

Product backlog is usually populated at the start of the project but also are ever-changing in nature.

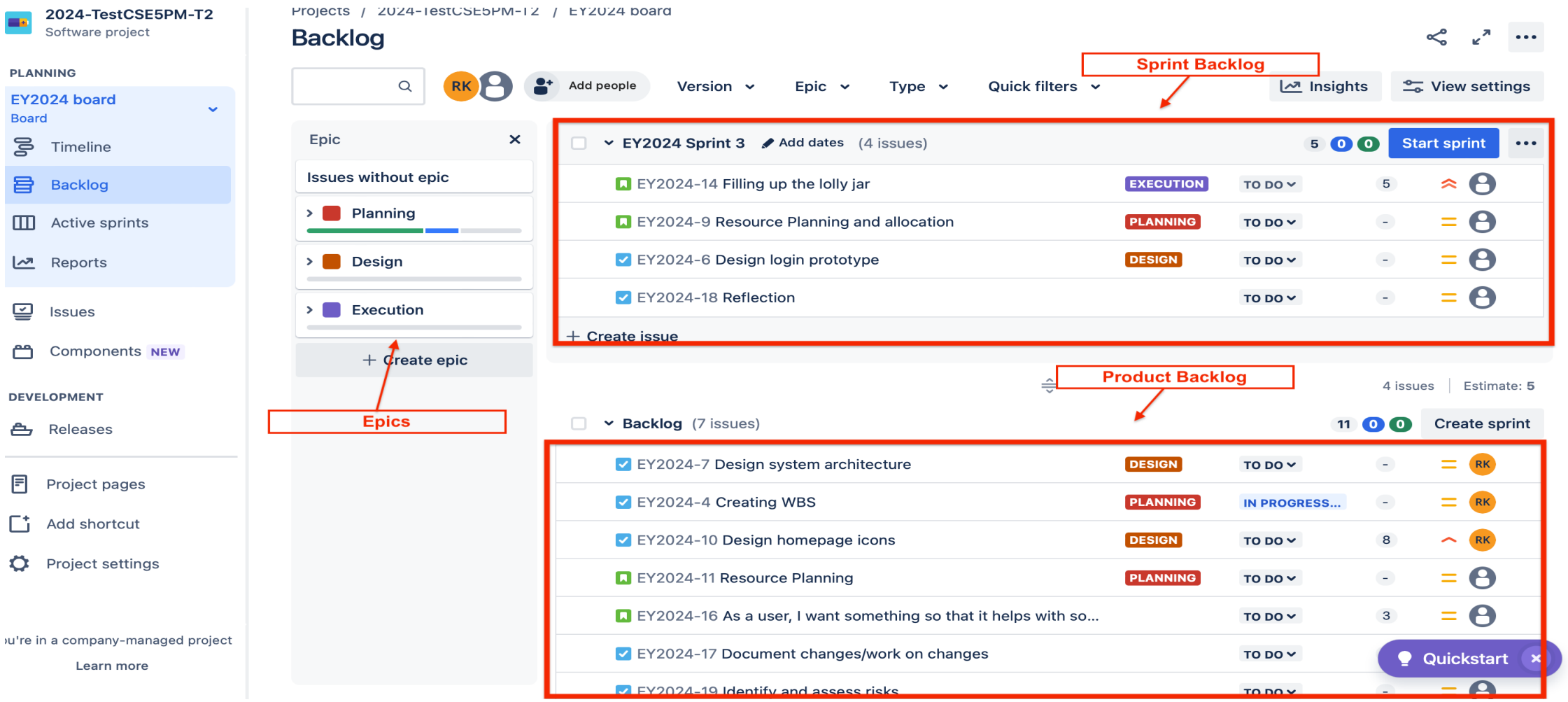
Scrum Artifacts – Sprint backlog

When building software, your work timeline is usually boxed into iterations, which are called sprints.

At the beginning of each sprint, all members must contribute to sprint planning by specifying which features from the product backlog they want to build, then, once the sprint starts, they will focus on building those features.

Every day, the team holds a short (10-15 minutes) meeting, which is called daily scrum meeting or daily stand-up meeting, to update each other on their progress towards completing the features; this is usually the time you bring up any blockers that are stopping you from completing your tasks, so that the team can discuss on how to unblock your work.

JIRA Product and Sprint backlog



Scrum Artifacts – Product increment

A product increment is the customer deliverables that were produced by completing product backlog tasks during a sprint.

It’s important that teams have a clear definition of “done”.

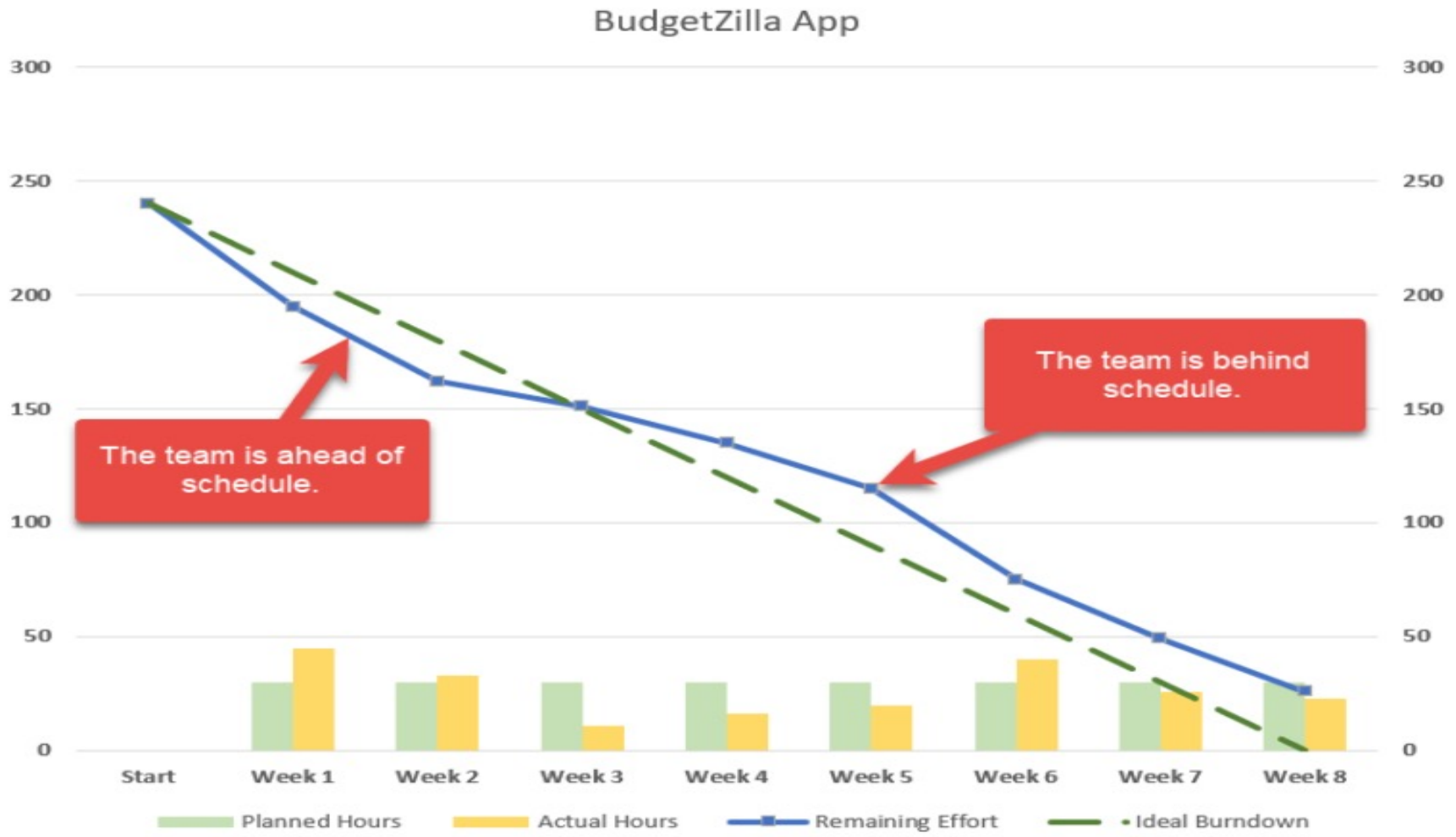
Burndown Chart – Extended Artifact

Burndown chart is a way for anyone to see, at a glance, and assess how the sprint is progressing/went.

A sprint burndown (or burnup) chart is not an official scrum artifact but many teams use it to communicate and track progress toward the sprint goal during the sprint.

Burndown charts are graphs that display tasks completed over the duration of a sprint (see next slide for example).

Burndown Chart – Extended Artifact



 how-to-create-a-burndown-chart-in-excel-768x603.png

Scrum Meetings

As stated in the sixth principle of the Agile movement, the best way to communicate is through face-to-face meetings. With Agile, there are two main types that you will definitely utilise:

- daily stand-up meetings
- Retrospective/sprint reflection meetings.

A quick rule of thumb is that everyone should answer the following questions during the daily stand-up:

What have I done since the last daily scrum?

What will I do until the next daily scrum?

What roadblocks are in my way?

Epic, Story and other issue types

Let’s say you and your team want to do something ambitious or are working on a big project. To do so, you’ll need to structure your work: from the largest objectives down to the smallest task. You’ll want to be able to respond to change, report your progress, and stick to a plan. Epics, stories, and initiatives are precisely the tools you’ll need to do so.

- Epics are large bodies of work that can be broken down into a number of smaller tasks (called stories/tasks).
- Story (this is where you state a problem to be solved in the project, written from the user’s perspective)
- Task (this may be an issue, or simply something on the project to-do list)
- Bug (something already identified in the project that requires fixing)

User Story/Story

Stakeholders and users cannot stand unpredictability.

Even if your team show absolute commitment to the project and put in a lot of effect, they will not want to use your project if it doesn't resemble what they were promised at the beginning, and we don't want to disappoint the stakeholders, do we? Remember the manifesto?

Therefore, a good place to start at the beginning of any Agile project is getting the team to think, view, see as the users. There's a very effective tool to help the team do this: user story.

User story is:

- a basic scenario of using the product
- a feature of the product
- a task that needs to be done to incrementally deliver a working product.

A basic user story template

As a **<type of user>**, I want **<some goal/feature>** so that **<some reason/benefit>**

or

As a **<who>**, I want **<what>** so that **<why>**

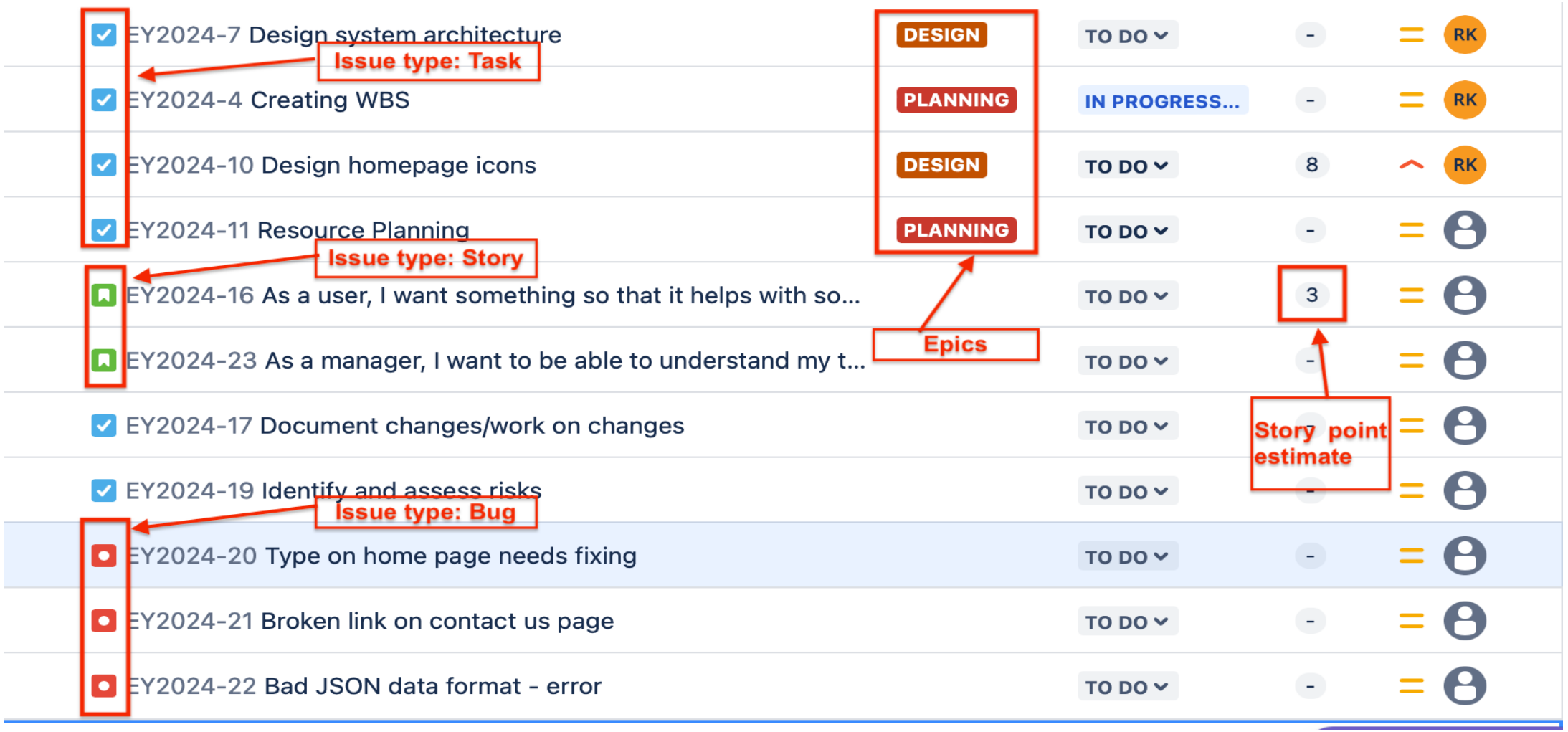
Story Points

Story points is an effective tool to help the team estimate how much work they can do in a sprint.

Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty.

> *"While there's no hard-and-fast rule about how many points to assign to each story, some teams assign between 1 and 5 story points to any user story. (The 5 value is arbitrary—other teams will give stories between 1 and 10 points, or use another number, as long as they use a consistent rule from sprint to sprint. Still other teams use numbers from the Fibonacci sequence, or exponential numbers. You can pick any scheme that works, as long as everyone on the team is comfortable with it.) One 3-point story should require about as much work as another 3-point story"*
>
> *(Stellman A., Greene J., Learning Agile. 2014, p. 146)*

JIRA Example – Issue types, epics, story points



Students to form their final groups by end of this week.

Team size: 3-5 members

One member per team to register their finalised team details along with team name using form on LMS by 1st April. An announcement will be made with form link/details to register teams.

Work through this week's activity