Brandon de Bruyn 6854

PRG522 FA2

Title

[Question 1: 3](#_Toc80003011)

[1.1 Describe the following terms as they are used in DevOps. 3](#_Toc80003012)

[a. Technical debt and its sources: 3](#_Toc80003013)

[b. Maven packages: 3](#_Toc80003014)

[1.2 Describe source control and its benefits 3](#_Toc80003015)

[1.3 Distinguish between Git and Team Foundation Version Control 3](#_Toc80003016)

[1.4 What is a pull request? Discuss. 4](#_Toc80003017)

[1.5 Describe a pipeline is Azure DevOps. 4](#_Toc80003018)

[1.6 Describe the benefits of continuous integration in DevOps. 4](#_Toc80003019)

[Question 2: 5](#_Toc80003020)

[2.1 5](#_Toc80003021)

[2.2 6](#_Toc80003022)

[2.3 7](#_Toc80003023)

[2.4 8](#_Toc80003024)

[2.5 9](#_Toc80003025)

# Question 1:

## Describe the following terms as they are used in DevOps.

### Technical debt and its sources:

Technical debt is that the accumulation of sub-optimal technical decisions remodelled the lifetime of an application. Eventually, it gets harder and harder to alter things: it’s the ‘sand within the gears’ that sees IT initiatives grind to a halt.

When it involves software development, technical debt is that the concept certain necessary work gets delayed during the event of a software project to hit a deliverable or deadline. Technical debt is that the coding you want to do tomorrow because you took a shortcut to deliver the software today

### Maven packages:

Maven packaging is an important aspect of any project. It specifies the type of artifact the project produces. Generally, a build produces a jar, war, pom, or other executable. Maven offers many default packaging types and provides the flexibility to define a custom one.

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation.

## Describe source control and its benefits

Source control is important for maintaining a single source of truth for development teams. Plus, using it helps facilitate collaboration and accelerates release velocity. That's because it allows multiple developers to work on the same codebase. They can commit and merge code without conflicts.

## Distinguish between Git and Team Foundation Version Control

The major difference with branching between Git and TFVC is that TFVC makes copies of the parent from which it branched while Git branches are just pointers to a commit. On the other hand, Git simply creates a new stream of commits for a branch. Thereby keeping only deltas between commits and branches.

Team Foundation Version Control (TFVC) is a centralized version control system provided by Microsoft as part of Azure DevOps Services, Azure DevOps Server and Team Foundation Server.

## What is a pull request? Discuss.

A pull request is an event in Git where a contributor asks a maintainer of a Git repository to review code they want to merge into a project.

An example of a pull request is where you as a project manager wants to merge a certain branch like “Back-End SQL Connection Branch” to the main Branch, The Project Manager will open a Pull Request to pull the new content of the Back-End Branch to merge it into the Main branch for public use.

## Describe a pipeline is Azure DevOps.

Azure Pipelines automatically builds and tests code projects to make them available to others. It works with just about any language or project type. Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target constantly and consistently.

## Describe the benefits of continuous integration in DevOps.

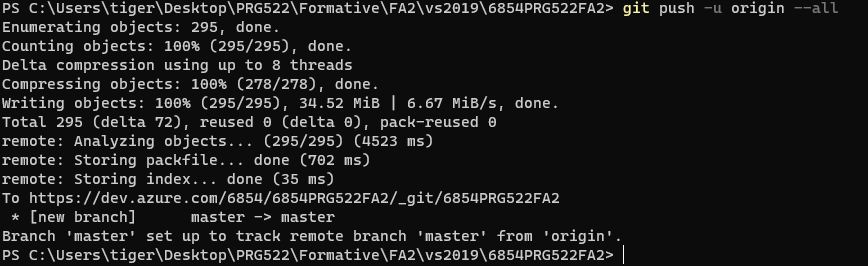
**Continuous integration** is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

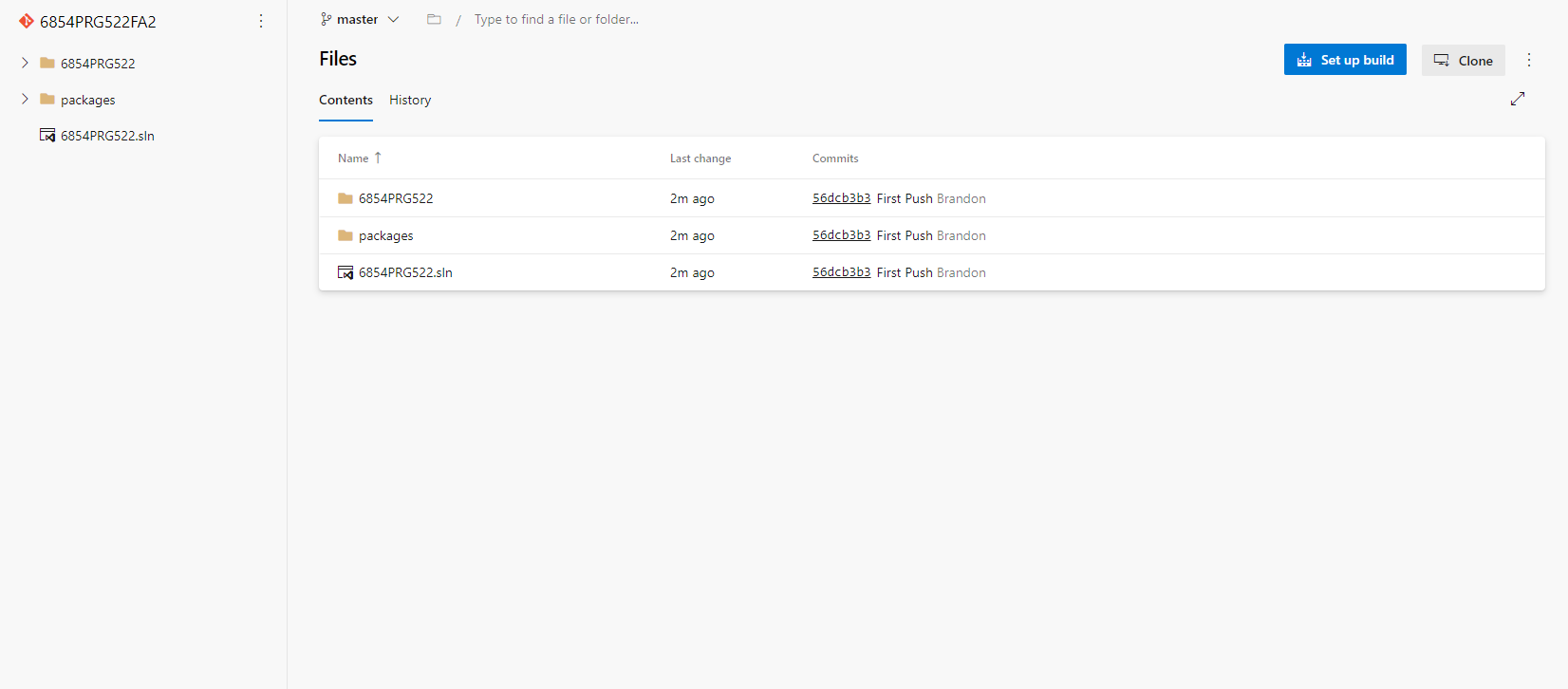
**Benefits of continuous integration.**

* Smaller code changes are simpler (more atomic) and have fewer unintended consequences.
* Fault isolation is simpler and quicker.
* Mean time to resolution (MTTR) is shorter because of the smaller code changes and quicker fault isolation.
* More Test Reliability Using CI/CD, test reliability improves due to the bite-size and specific changes introduced to the system, allowing for more accurate positive and negative tests to be conducted.
* Faster Release Rate Failures are detected faster and as such, can be repaired faster, leading to increasing release rates.
* Smaller Backlog Incorporating CI/CD into your organization’s development process reduces the number of non-critical defects in your backlog.

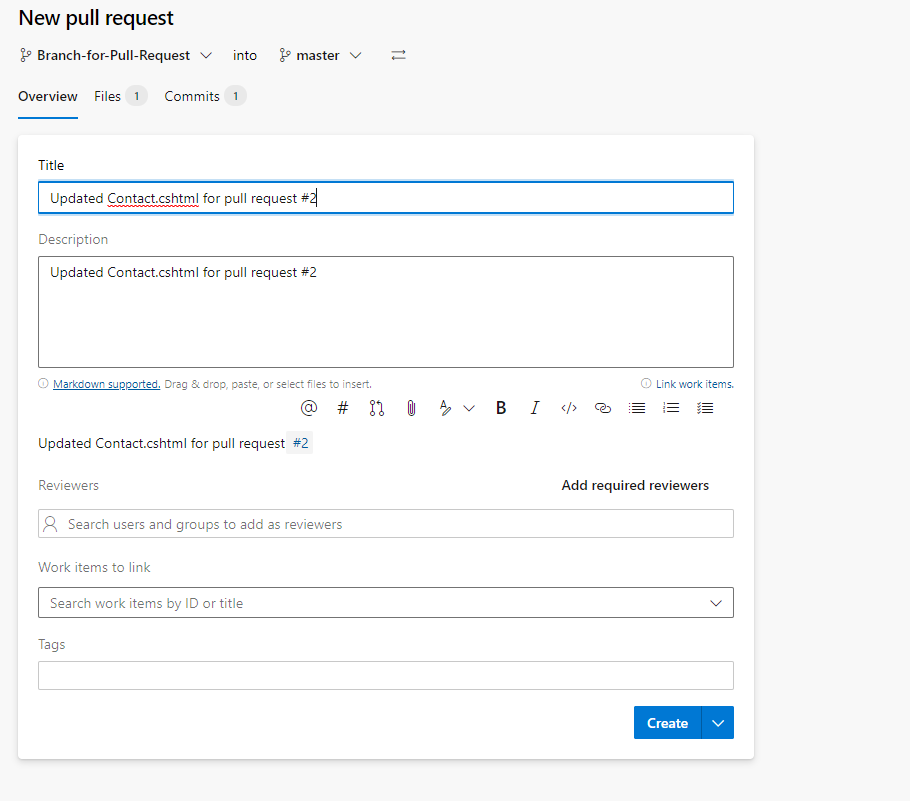
# Question 2:

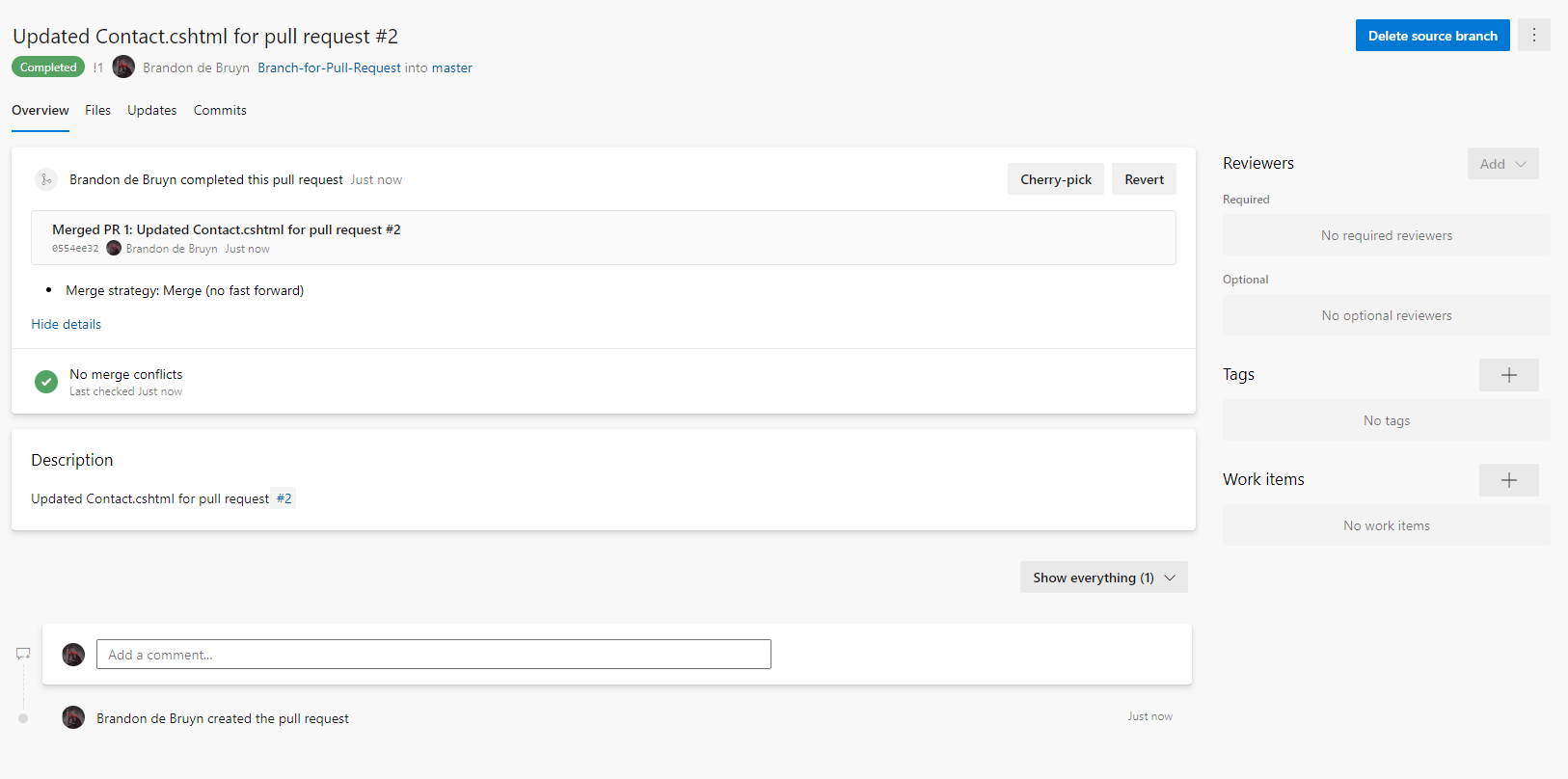
## 2.1



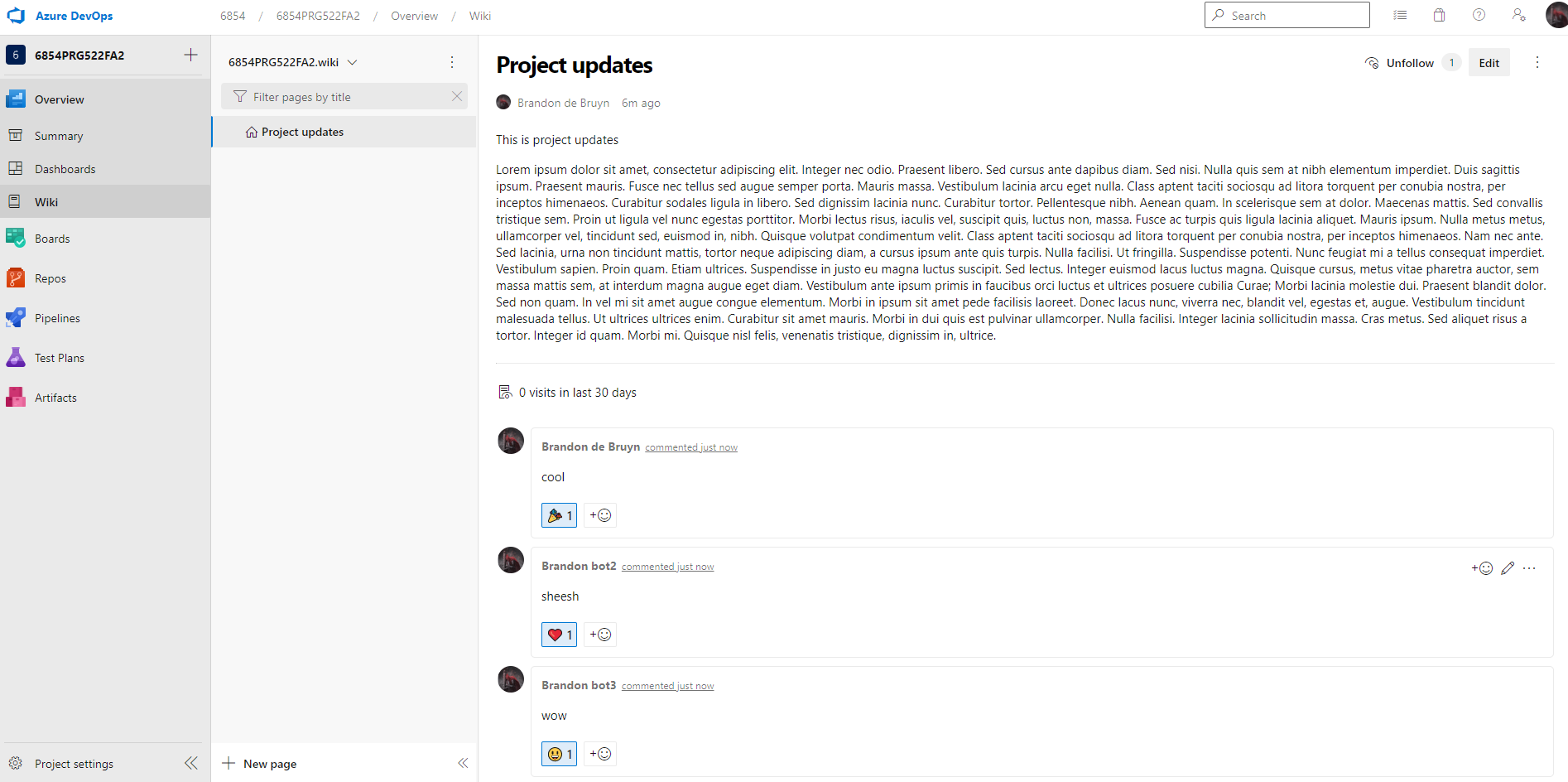


## 2.2

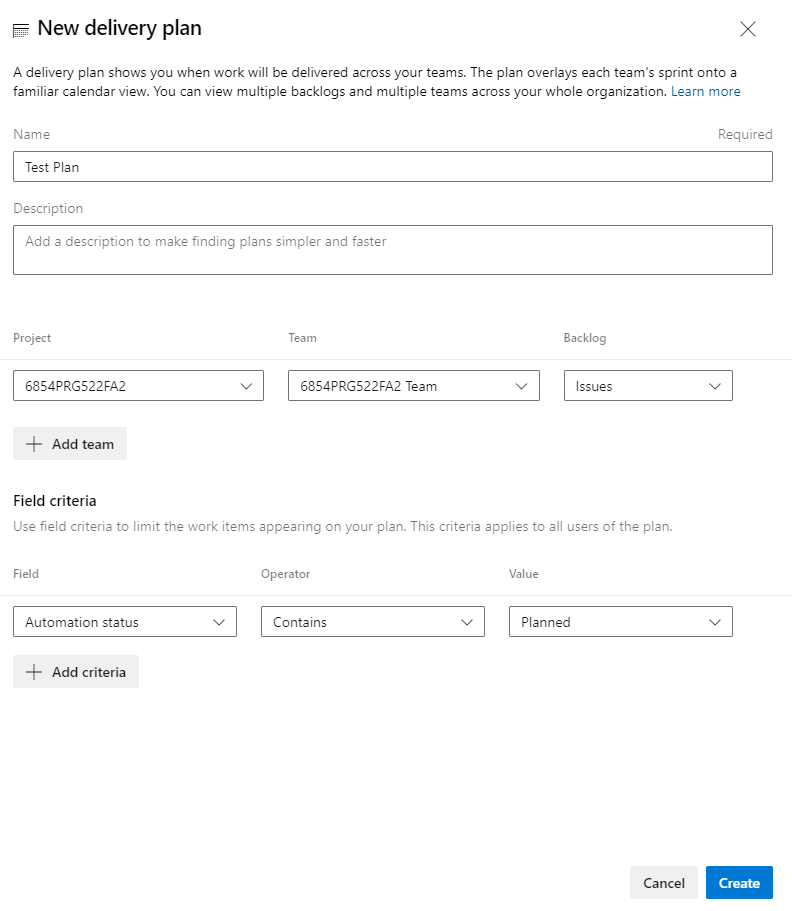


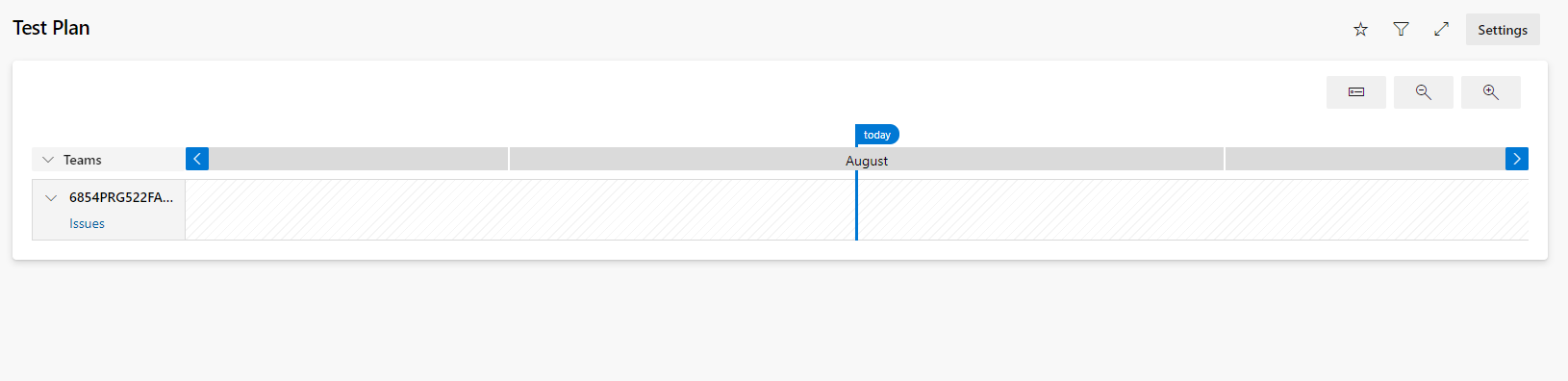


## 2.3



## 2.4





## 2.5

