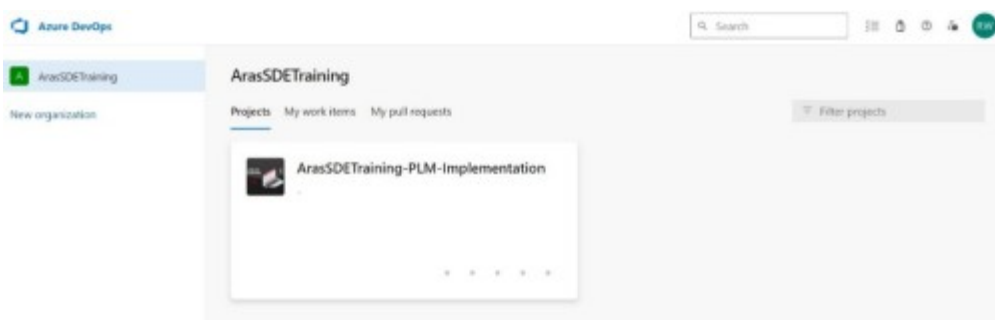


# SDE Navigation

The URL to the SDE environment is accessible only after the invitation process, which requires setting up multi-factor authentication, is completed.

When connecting to the SDE URL, the following screens appear:

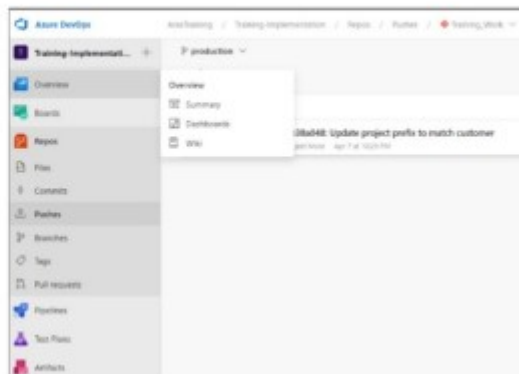
- **Azure DevOps landing page:** The visible implementation project represents where the project team customizes Aras Innovator for subscribers.



- **Azure DevOps Structure:** Displays required information.

## Azure DevOps Structure

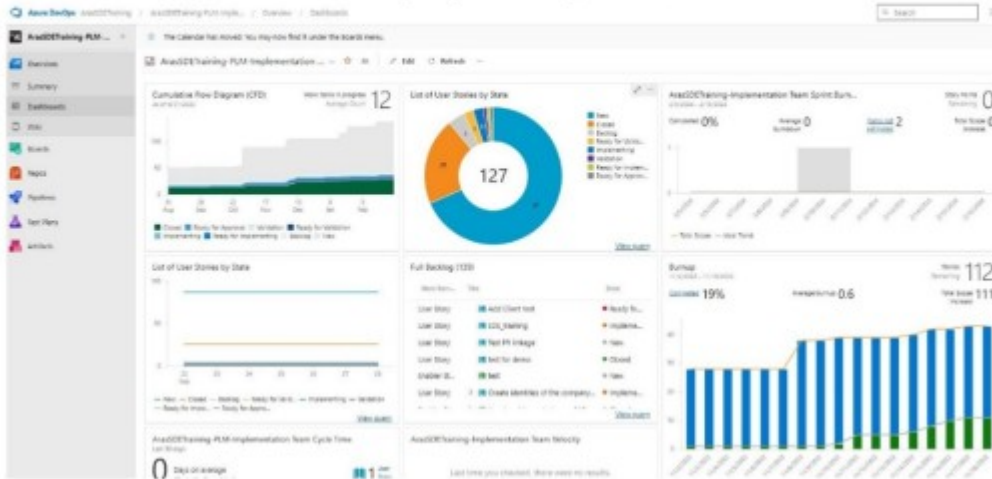
- Overview – dashboards (scrum Masters)
- Boards – backlog, queries, etc.
- Repos – team vs. Individual
- Pipelines – ContinuousIntegration, DeployToSIT, etc.
- Test Plans – Test Plans, Runs
- Artifacts – Baselines, Feeds



- **Dashboards:** Scrum masters use these to track project progress for team members' Views. A link is provided for traceability between change management and implementation.

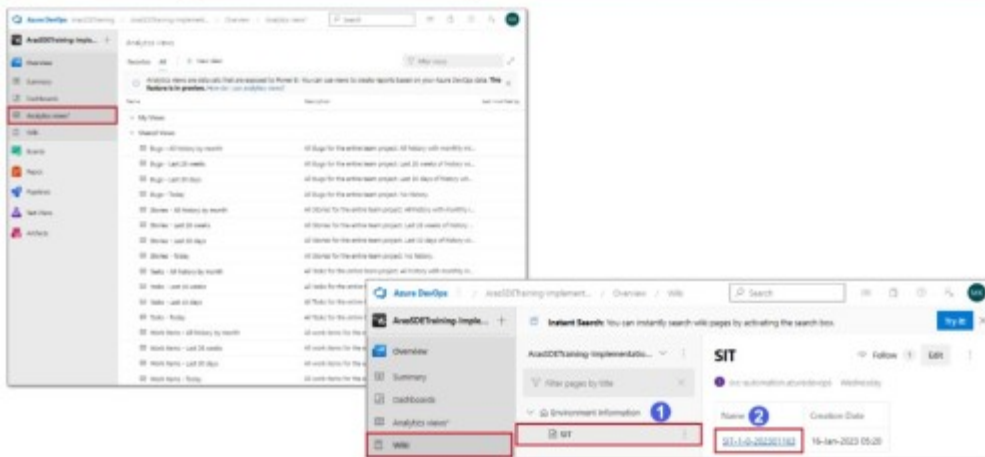
## Azure DevOps Overview – Dashboards

Customizable, highly configurable dashboards provide you and your teams with the flexibility to share information, monitor progress and trends, and improve your workflow processes.



- **Analytics Views and Wiki Page:** The team project **Wiki** is a platform for sharing information with other team members. Provisioning a **Wiki** from scratch initiates a new Git Repository that stores Markdown files, images, attachments, and a sequence of pages. The **Wiki** can support collaborative editing of both its content and structure. Aras uses **Wiki** to share the URLs of SIT test instances.

## Azure DevOps Overview – Analytics views & Wiki



- **Work Items:** These Items are required for approved changes. This may encompass other tasks necessitated by the project or the tracking of requests made to Aras.

## Azure DevOps Boards – Work Items

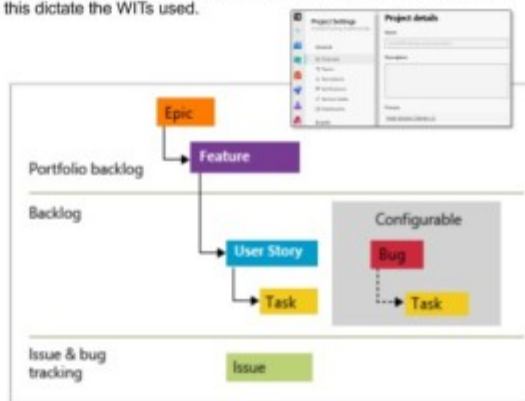
- Work item types (WITs) represent the core of the Azure DevOps tracking system and can be a bug, a requirement, a general to-do, etc.
- Each work item has a unique ID to keep track of its references from its creation to its implementation as a piece of executable software.

| ID  | Title                         | Assigned To | State  | Area Path               | Tags | Comments | Activity Date        |
|-----|-------------------------------|-------------|--------|-------------------------|------|----------|----------------------|
| 360 | Add Design Request P05        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:00:00 PM |
| 360 | Design Request Management P05 | Unassigned  | Future | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:00 PM |
| 360 | Change Management P05         | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:47 PM |
| 360 | Add Design Request C03        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 1:01:04 PM |
| 367 | Add Design Request C04        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 1:01:10 PM |
| 368 | Add Design Request C08        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 1:03:45 PM |
| 368 | Add Design Request C10        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 1:07:39 PM |
| 369 | Design Request Management C10 | Unassigned  | Future | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:10 PM |
| 372 | Change management C04         | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:57 PM |
| 378 | Design request management C06 | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:40 PM |
| 378 | Design Request Management C12 | Unassigned  | Future | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:42:30 PM |
| 383 | Change Management C10         | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:40:28 PM |
| 398 | Install package C18           | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:50:44 PM |
| 399 | Change management C10         | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:50:27 PM |
| 399 | Add Design Request C16        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:57:09 PM |
| 399 | Add Design Request C07        | Unassigned  | New    | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:57:46 PM |
| 397 | Design Request Management C16 | Unassigned  | Future | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:57:11 PM |
| 328 | Design Request Management C07 | Unassigned  | Future | ArasDevOps-PCM-imple... |      |          | 2/16/2024 2:57:02 PM |

- **Work Item Structure:** Azure DevOps manages the process according to the definition set by Aras Global Cloud Services (GCS). GCS continually updates the process to address subscriber input and feedback. The current Agile Solution Delivery version is 4.1.

## Azure DevOps Boards – Agile Process Model

- Supports Agile planning methods, including Scrum, and tracks development and test activities separately. This process works great if you want to track user stories and (optionally) bugs on the Kanban board, or track bugs and tasks on the taskboard. The process type is based on **Agile Solution Delivery v3**, and this dictate the WITs used.
- Hierarchy
  - Epics
  - Features
  - Stories: User Story and Enabler Story, with Acceptance Criteria
  - Areas: Functional, capability attached/assigned to teams
  - Sprints (Iterations): Used for refinement (grooming)
- Set definitions of Ready and Done
- Epics and features reflect the business focus, user stories and tasks are related to the development
- Example:
  - Area: Certification
  - Epic: Change Management
  - Feature: Design Request Management
  - User Story: Add Design Request

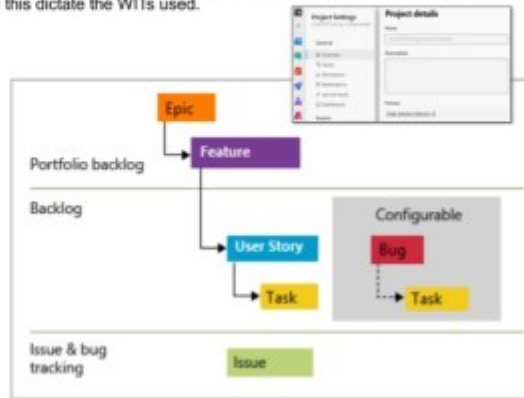


- **Boards:** The **Boards** can track various **Work Items**, including features, user stories, tasks, bugs, etc. They support both Scrum and Kanban methodologies. Additionally, the **Boards**

include capabilities for planning **Sprint**, managing **Backlogs**, and generating reports on work progress.

## Azure DevOps Boards – Agile Process Model

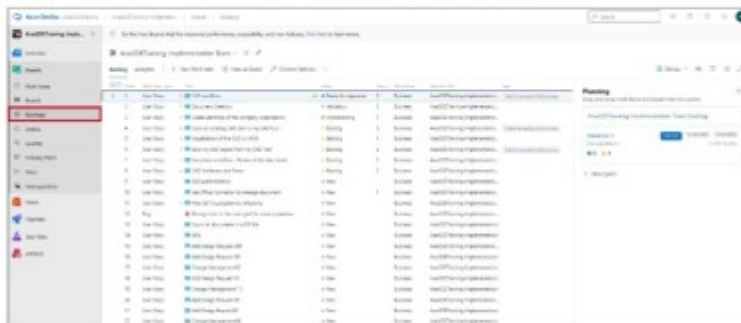
- Supports Agile planning methods, including Scrum, and tracks development and test activities separately. This process works great if you want to track user stories and (optionally) bugs on the Kanban board, or track bugs and tasks on the taskboard. The process type is based on **Agile Solution Delivery v3**, and this dictate the WITs used.
- Hierarchy
  - Epics
  - Features
  - Stories: User Story and Enabler Story, with Acceptance Criteria
  - Areas: Functional, capability attached/assigned to teams
  - Sprints (Iterations): Used for refinement /grooming
- Set definitions of Ready and Done
- Epics and features reflect the business focus, user stories and tasks are related to the development
- Example:
  - Area: Certification
  - Epic: Change Management
  - Feature: Design Request Management
  - User Story: Add Design Request



- **Backlogs:** **Backlogs** in Azure DevOps are used to manage and prioritize **Work Items** in a queue. They provide an ordered list of **Work Items**, such as user stories, features, or bugs that the team needs to work on.

## Azure DevOps Boards – Backlogs

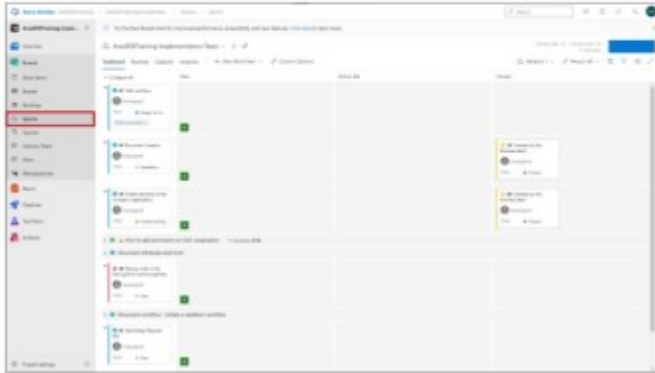
- Backlogs display work items as a list and boards display them as cards. With list backlogs, you can quickly develop your project plan, group and prioritize work, and make bulk updates on selected work items.
- There are three classes of backlogs
  - **Portfolio backlogs** typically track high-level features, scenarios, or epics; they help you organize your product backlog into a hierarchy of elements.
  - The **product backlog** contains a prioritized list of user stories, deliverables, or works you plan to build or fix.
  - **Sprint backlogs** contain just those items that each team is working on during a scheduled sprint or iteration period.



- **Sprints:** **Sprints** in Azure DevOps represent time-boxed iterations where a set amount of work is completed.

## Azure DevOps Boards – Sprints

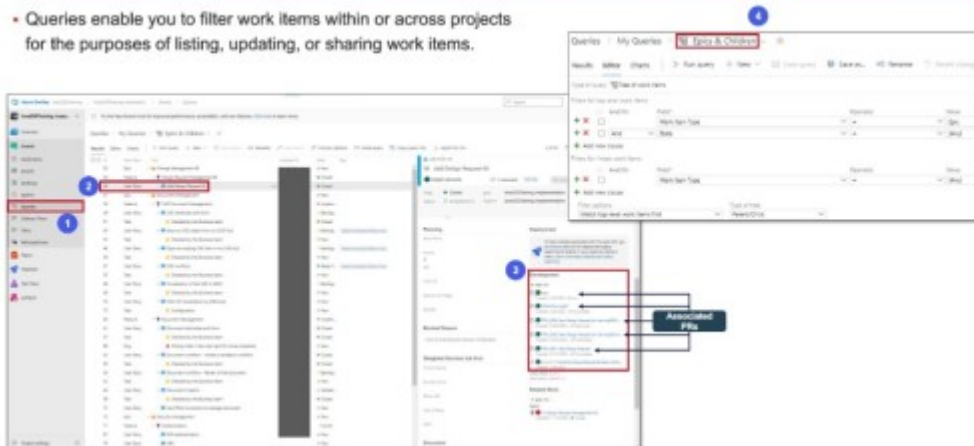
- Sprints, specified by iteration Paths, are defined for a project and then selected by teams. A sprint cadence can vary between one week to four weeks or longer.
  - You assign work to sprints that teams commit to deliver at the end of the sprint.
  - Azure Boards tools rely on sprint assignments to a team Sprint backlogs, Taskboard, and Delivery plans.



- **Queries:** Queries enable users to filter **Work Items** within or across projects for listing, updating, or sharing **Work Items**.

## Azure DevOps Boards – Queries

- Queries enable you to filter work items within or across projects for the purposes of listing, updating, or sharing work items.

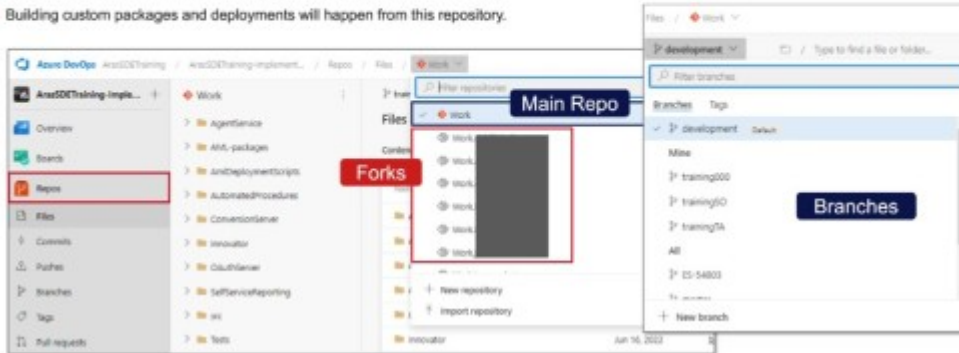


- **Repos:** There is only one main configuration **Repo** per project for the Standard Development Environment. Notice the orange and white backgrounds of the Git logo. The white background repositories are Forks.



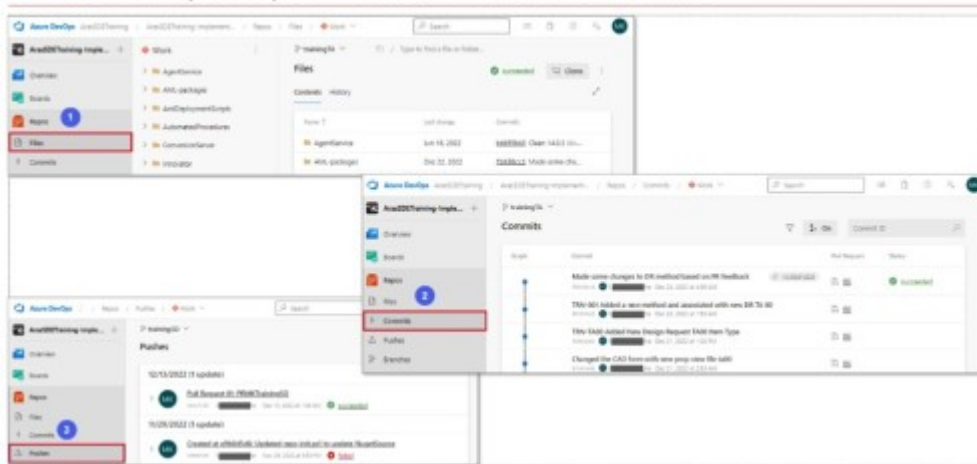
## Azure DevOps Repos – Repos

- Azure Repos is a set of version control tools that you can use to manage your code.
- Before we start the development or implementation, based on Innovator version required, an Azure DevOps repository (CRT) will be created by GCS and access will be given to customer project team
- This Azure DevOps Repo will be based on CRT, and all customizations and configurations should be committed to this Azure DevOps repository.
- Building custom packages and deployments will happen from this repository.



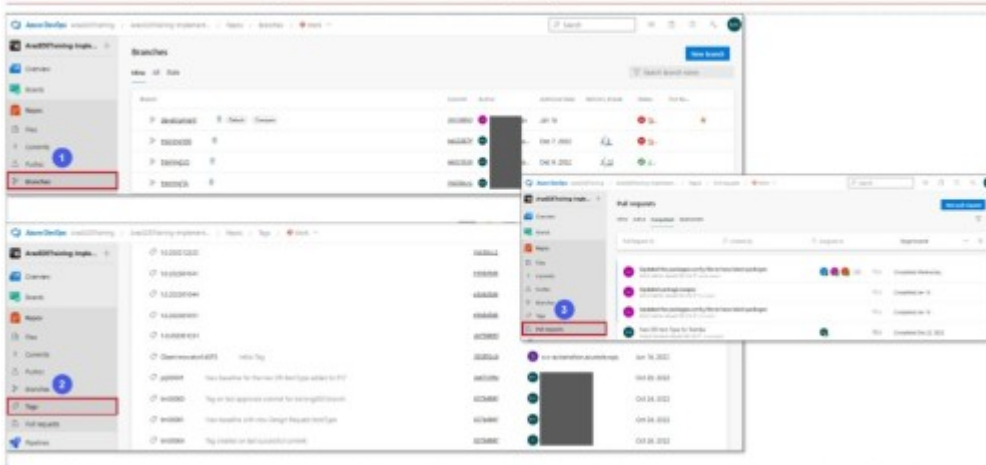
- **Files, Commits, and Pushes:** In Azure DevOps, **Files** are the individual project components, **Commits** are snapshots of changes made to those files, and **Pushes** are the action of uploading these **Commits** to a Remote Repository for team access and collaboration.

## Azure DevOps Repos – Files, Commits & Pushes



- **Branches, Tags, and Pull Request:** In Azure DevOps, **Branches** are separate versions of the codebase for isolated development, **Tags** are reference points to specific versions of the code, and a **Pull request** is a mechanism for developers to propose, review, and merge changes from one **Branch** to another.

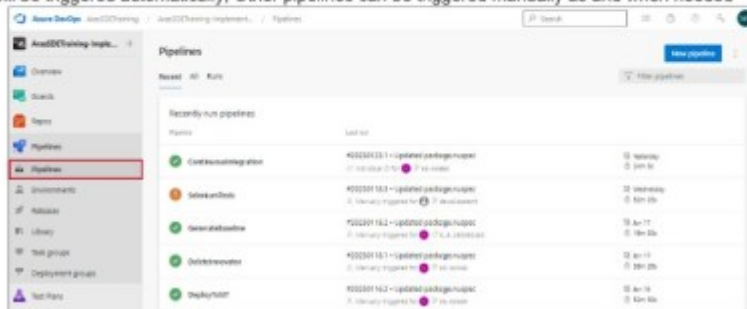
## Azure DevOps Repos – Branches, Tags & Pull Requests



- **Pipelines:** Pipelines in Azure DevOps are automated workflows for Continuous Integration and Delivery, enabling code build, test, and deployment processes.

## Azure DevOps Pipelines – Pipelines

- Azure Pipelines automatically builds and tests code projects to make them available to others.
- Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target (SIT, and other environments).
- These pipelines are configured for your repository based on the subscription model
- CI Pipeline will be triggered automatically; Other pipelines can be triggered manually as and when needed



- **Test Plans:** Test Plans in Azure DevOps provide a structured approach for defining, tracking, and managing testing activities to ensure software quality.

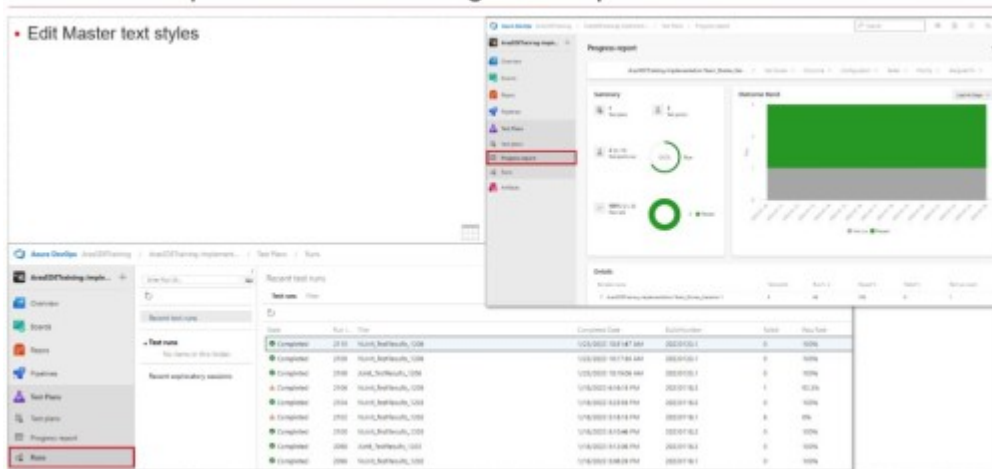
## Azure DevOps Test Plans – Test Plans

- Azure Test Plans is an easy-to-use, browser-based test management solution that provides all the capabilities required for planned manual testing, user acceptance testing, exploratory testing, and gathering feedback from stakeholders.
- This is an addition to the standard SaaS environment (additional cost).



- **Progress Reports and Runs:** **Progress Reports** in Azure DevOps provide insights into the development lifecycle and project milestones, while **Runs** represent individual executions of tests, builds, or deployments.

## Azure DevOps Test Plans – Progress Report & Runs



- **Baselines and Artifacts:** **Baselines** in Azure DevOps represent specific versions of the project for comparison or recovery, while **Artifacts** are the output files generated from build and release **Pipelines**.



## Azure DevOps Artifacts – Baselines & Artifacts

- With Azure Artifacts, you can create and share NuGet and npm packages from private and public sources with teams in Azure DevOps. These packages can be used in source code and can be made available to the CI/CD pipelines. With Azure Artifacts, you can create multiple feeds that you can use to organize and control access to the packages.
- Feeds
  - Baselines
    - Provides the GCS initial baselines that can be used by customers
    - Home for future baselines created during the customer development cycles
  - Artifacts (non-SaaS customers)
    - Provides diverse CRT and System related packages

