



# Azure DevOps

Azure Artifacts and Azure Test Plans



# Agenda

**01**

**What is Azure Artifacts**

**02**

**Working with Packages**

**03**

**Feeds, views and upstream sources**

**04**

**Connecting to Azure Pipelines**

**05**

**What are Azure Test Plans**

**06**

**Exploratory and Manual Testing**

**07**

**Test from Kanban Board**

**08**

**Creating Test Plans**

# What is Azure Artifacts?

# What is Azure Artifacts?

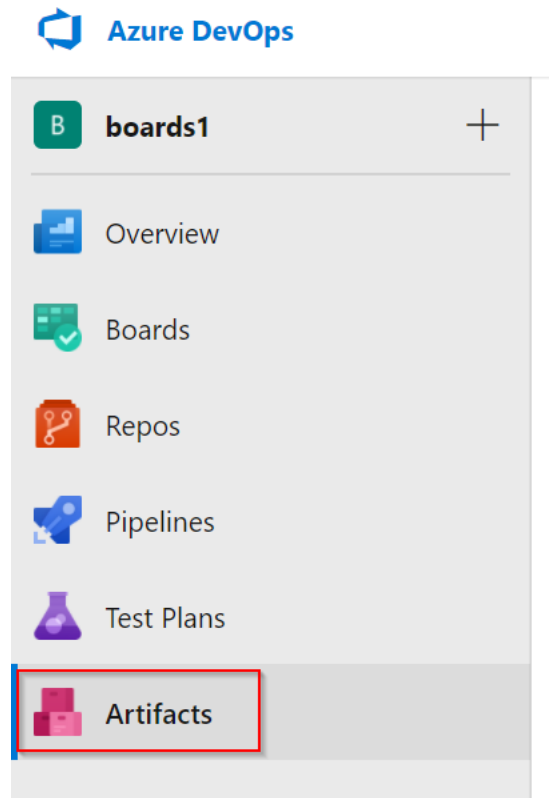
With Azure Artifacts you can create and share Maven, npm, and NuGet package feeds from public and private sources with teams of any size.



You can add fully integrated package management to your continuous integration/continuous delivery (CI/CD) pipelines with a single click.

# What is Azure Artifacts?

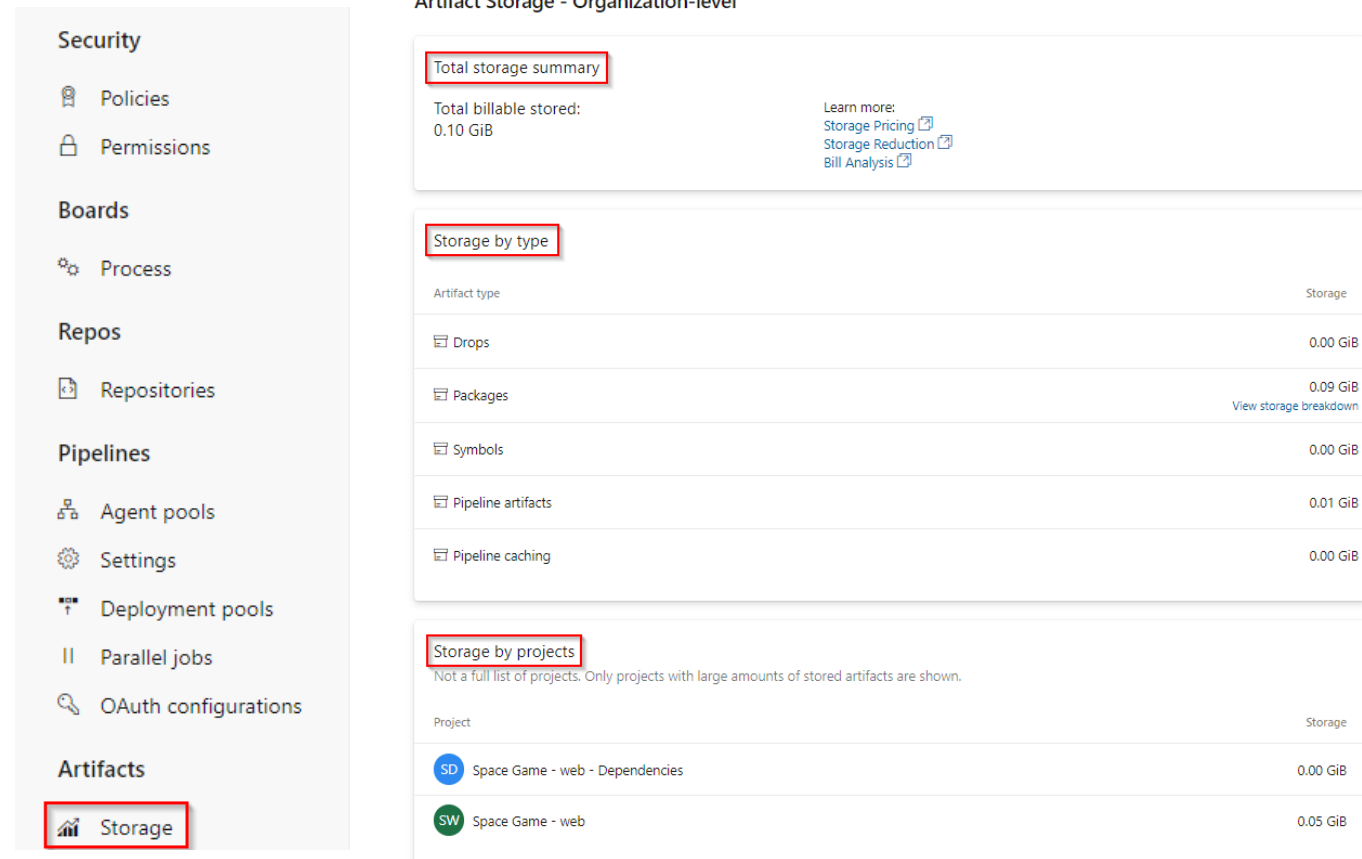
The new update has introduced a new, top-level area that is the home of Package Management in Azure DevOps Services. This area is known as Azure Artifacts.



# What is Azure Artifacts?

Azure Artifacts offers a free-tier plan that includes 2 gigabytes (GiB) of free storage for different types of packages.

When you reach your maximum storage limit, you can either upgrade to a paid subscription or delete some of your existing artifacts.



**Artifact Storage - Organization-level**

**Total storage summary**

Total billable stored: 0.10 GiB

Learn more:  
[Storage Pricing](#)  
[Storage Reduction](#)  
[Bill Analysis](#)

**Storage by type**

Artifact type	Storage
Drops	0.00 GiB
Packages	0.09 GiB <a href="#">View storage breakdown</a>
Symbols	0.00 GiB
Pipeline artifacts	0.01 GiB
Pipeline caching	0.00 GiB

**Storage by projects**

Not a full list of projects. Only projects with large amounts of stored artifacts are shown.

Project	Storage
SD Space Game - web - Dependencies	0.00 GiB
SW Space Game - web	0.05 GiB

# What is Azure Artifacts?

## Best Practices for using Azure Artifacts

- Each repository should only reference one feed.
- On package creation, automatically publish packages back to the feed.
- Enable retention policies to automatically cleanup old package versions.
- A feed is a container for packages, The only package source for each repository should be that single unique feed.

# Working with Packages



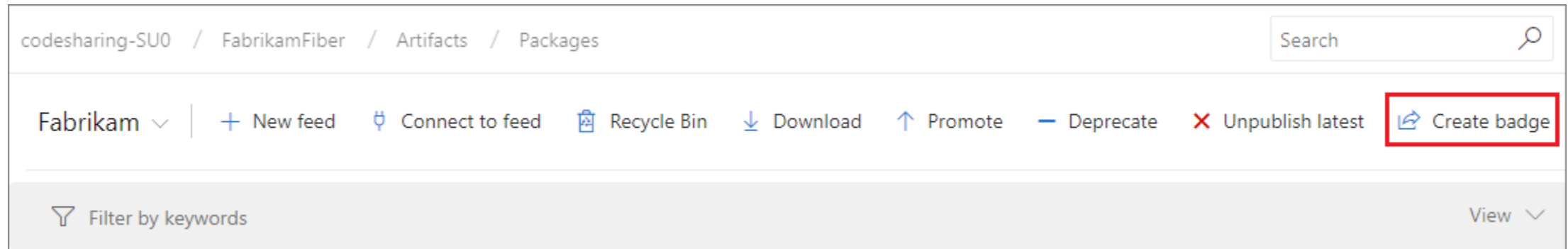
# Working with Packages

A package is container file which contains scripts, files and a manifest which can be used as an external dependency to a project.



# Working with Packages

You can share your packages anywhere you can share an image with badges. You can put a badge directly into your project's home page in Azure DevOps Services or in any Markdown/README file so readers can easily discover and consume your package.



Example Badge:



Azure Artifacts is a highly-scalable artifact service. However, the service does have some architectural limits and also some limits imposed by the client tools (e.g. nuget.exe) it integrates with.

## Size limits

- NuGet packages are limited to 500 MB.
- npm packages are limited to 500 MB.
- Maven packages are limited to 500 MB per file.
- Python packages are limited to 500 MB per file.
- Universal Packages have been tested up to 1 TB and are recommended for managing large binary content.

# Feeds, views and upstream sources

# Hands-on: Starting off with Azure Artifacts

# Feeds, views and upstream sources

**Artifacts Feeds** are organizational constructs that allow you to store, manage, and group your packages and control who to share it with. Feeds are not package-type dependent.

## Project-scoped

- Project-scoped feeds will always use the visibility of the project. If a project is public, the feed is also public and vice versa.
- Creating a new feed through the create feed web UI will create a project-scoped feed.
- Project-scoped feed:  
[https://feeds.dev.azure.com/contoso/projectId/\\_apis/Packaging/Feeds](https://feeds.dev.azure.com/contoso/projectId/_apis/Packaging/Feeds)

## Organization-scoped

- Organization-scoped feeds will always remain private.
- All organization-scoped feeds will show up in the feed list of the Artifacts feed UI.
- Organization-scoped feed:  
[https://feeds.dev.azure.com/contoso/\\_apis/Packaging/Feeds](https://feeds.dev.azure.com/contoso/_apis/Packaging/Feeds)

# Feeds, views and upstream sources

**Feed views** enable you to share subsets of package-versions with your consumers. A common use of feed views is to share package versions that have been tested and validated but hold back on packages that are still under development.

**All Artifacts feeds come with three views:**

@local

@prerelease

@release

The latter two are suggested views that you can rename or delete as desired. **@local** is the **default view** that's commonly used in upstream sources.

## Using feed views to release packages

When creating packages in continuous integration and delivery scenarios, it's important to convey three pieces of information: the nature of the change, the risk of the change, and the quality of the change.

1.2.3-beta2



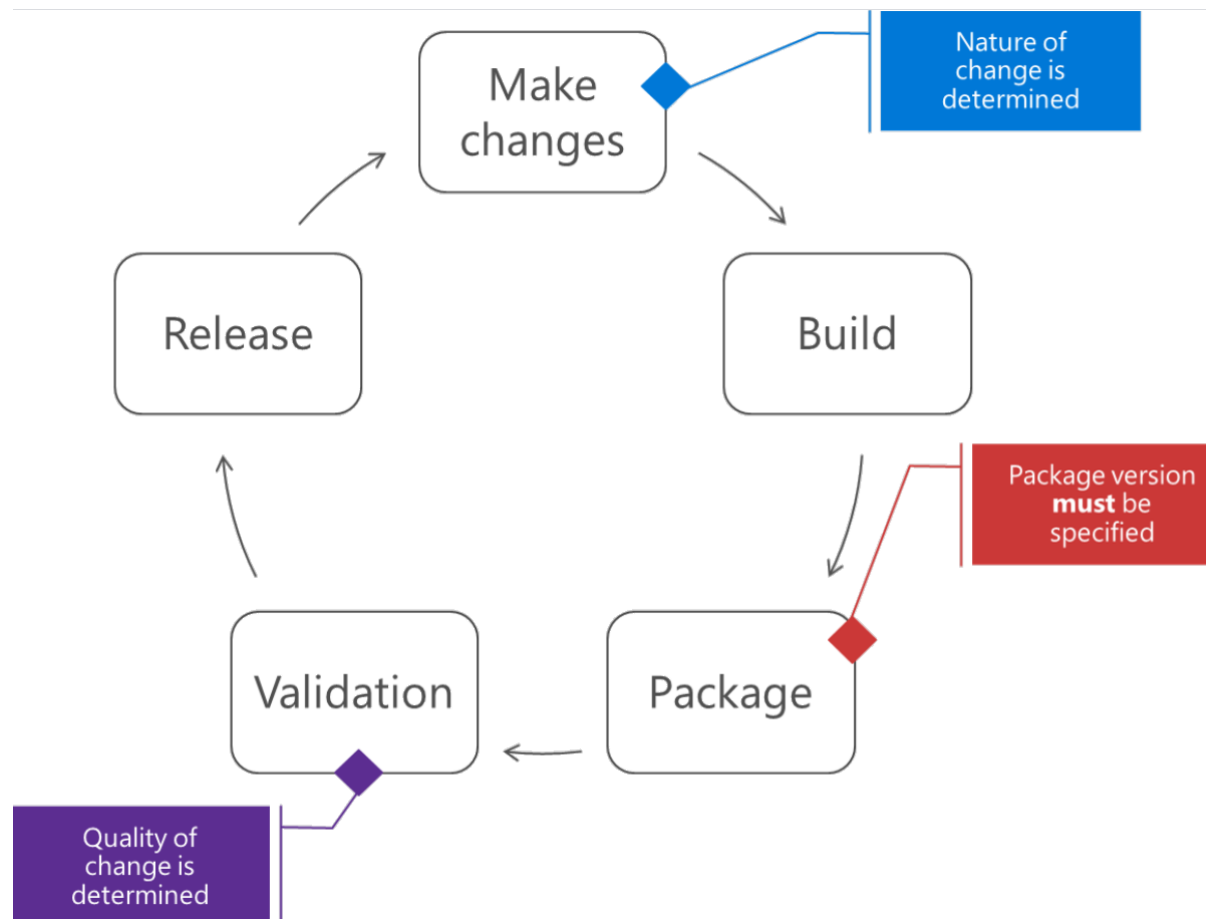
nature of  
change

quality of  
change



# Feeds, views and upstream sources

Determine and communicate quality of the change



# Feeds, views and upstream sources

Promote a package to prerelease or release view

Azure DevOps

codesharing-SU0 / FabrikamFiber / Artifacts / Packages

FabrikamFiber +

Overview Boards Repos Pipelines Test Plans Artifacts

Fabrikam > accepts 1.3.4

Overview Versions | Connect to feed Download **Promote** Deprecate Unpublish Create badge Follow

Get this package

Connect to feed then `npm install accepts@1.3.4`

Already connected to another feed? Copy this feed locator and add an *upstream source* to your feed, then run the install command above. [Here's how](#)

`https://registry.npmjs.org/`

Description

Higher-level content negotiation

# Feeds, views and upstream sources

Promote a package to prerelease or release view

Promote this package

View

FabrikamFiber@Prerelease



FabrikamFiber@Prerelease

FabrikamFiber@Release

# Feeds, views and upstream sources

**Upstream sources** enable you to use a single feed to store both the packages you produce and the packages you consume from "remote feeds": both public package managers (npmjs.com, NuGet.org, Maven Central, and PyPI) and Artifacts feeds.

FabrikamFiber > Feed settings

Feed details

Permissions

Views

**Upstream sources**

+ Add upstream source

🗑 Delete

↑ Move up

↓ Move down

Source	Location	Source type	Last synced
 NuGet Gallery	https://api.nuget.org/v3/index.json	Public source	✅ Today at 2:27 PM

Once you've enabled an upstream source, any user connected to your feed can install a package from the remote feed, and your feed will save a copy.

# Connecting to Azure Pipelines

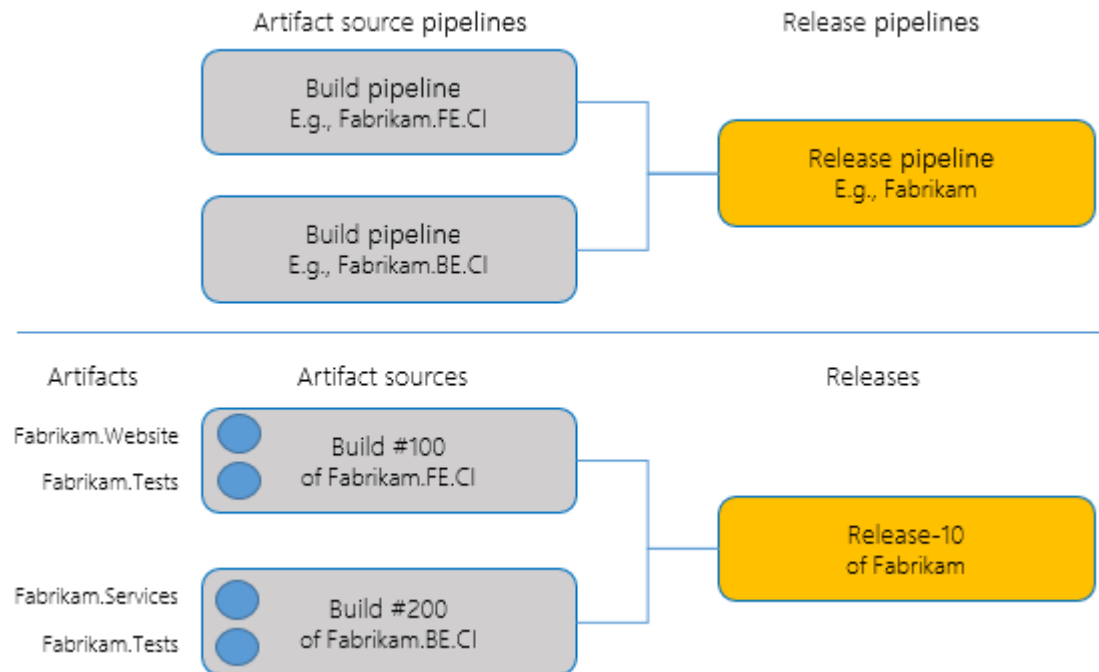
# Connecting to Azure Pipelines

## Supported artifact types in Azure Pipelines

Supported Artifact Types	Description
Build Artifacts	Build artifacts are the files that your build produce. E.g. .dll, .exe
Pipeline Artifacts	Help store build outputs and move intermediate files between jobs in your pipeline.
Maven	You can publish Maven artifacts to Azure Artifacts feeds or Maven repository.
npm	You can publish npm packages to Azure Artifacts feeds or npm registry.
NuGet	You can publish NuGet packages to Azure Artifacts feeds or NuGet repository.
Universal Packages	Universal Packages store one or more files together in a single unit that has a name and version.

# Connecting to Azure Pipelines

A **release** is a collection of artifacts in your DevOps CI/CD processes. An artifact is a deployable component of your application.



When **authoring a release pipeline**, you link the appropriate artifact sources to your release pipeline.

When **creating a release**, you specify the exact version of these artifact sources. A single release pipeline can be linked to multiple artifact sources, of which one is the primary source.

# What are Azure Test Plans?



# What are Azure Test Plans?

Azure Test Plans provides a browser-based test management solution for exploratory, planned manual, and user acceptance testing. For automated testing as part of your CI/CD workflow, consider leveraging Azure Pipelines.



## Capture rich data

Capture rich scenario data as you execute tests to make discovered defects actionable.



## Get end-to-end traceability

Take advantage of end-to-end traceability and quality for your stories and features.



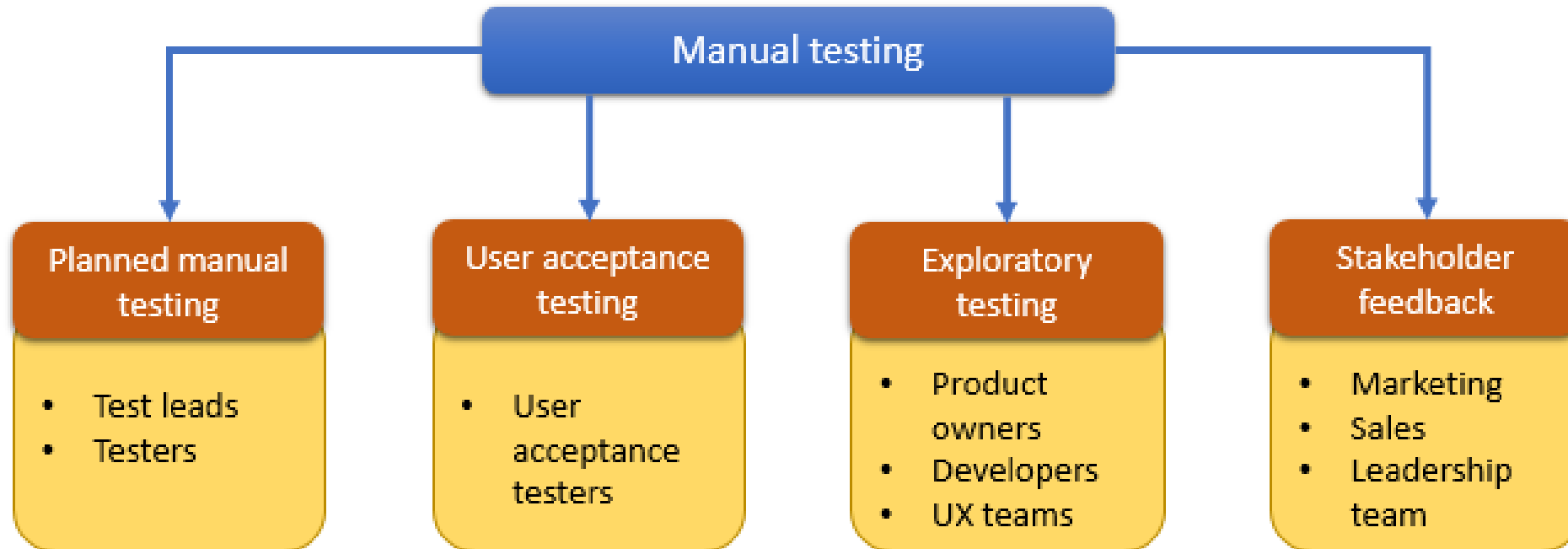
## Test across web and desktop

Test your application by executing tests across desktop or web apps.

# Exploratory and Manual Testing

# Exploratory and Manual Testing

Quality is a vital aspect of software systems, and manual testing and exploratory testing continue to be an important techniques for maximizing this.



# Exploratory and Manual Testing

## Planned manual testing



1 In any browser navigate to homepage and log in



2 Navigate to a product and add it to cart



3 Check if cart is populated with Item

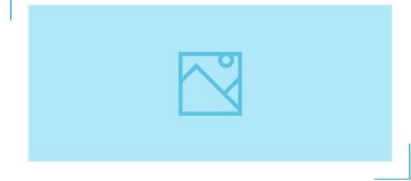


This results in a 404 once you try to navigate to the shopping cart

## Exploratory testing



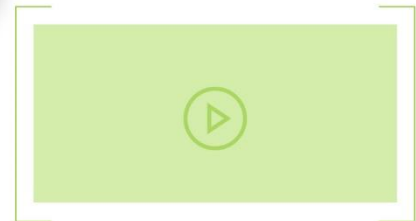
Captured annotated screenshot



Captured page load data



Captured video



Created bug 12526

# Exploratory and Manual Testing

## Test planning

Team aware

Create test plans and test suites

Filter test plans

Accessible from the web

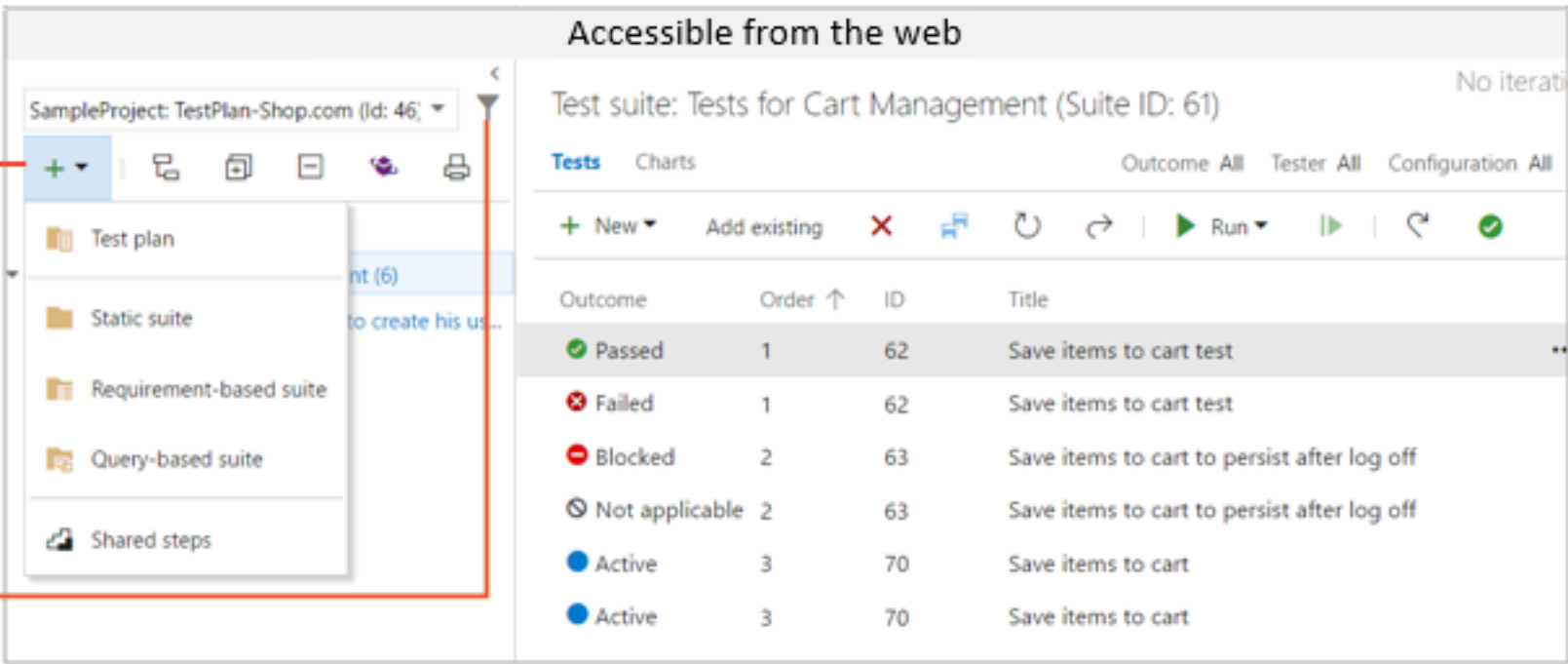
SampleProject: TestPlan-Shop.com (Id: 46)

Test suite: Tests for Cart Management (Suite ID: 61)

Tests Charts Outcome All Tester All Configuration All

+ New Add existing X Run Run Run

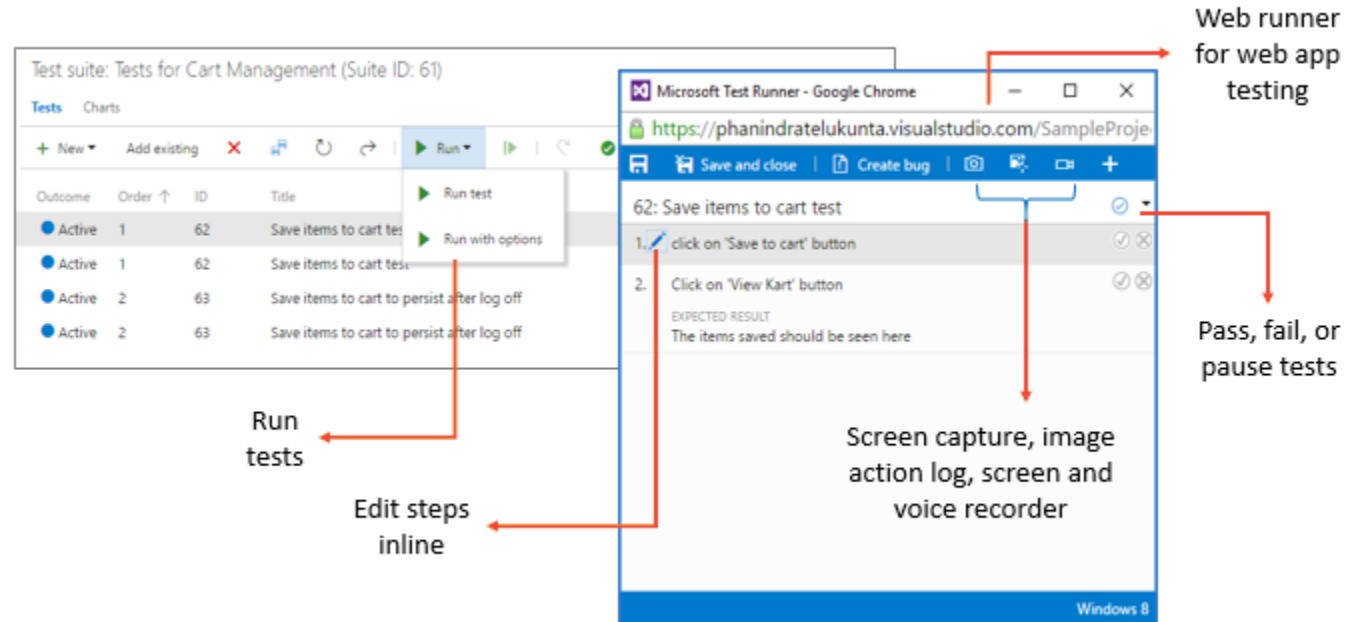
Outcome	Order ↑	ID	Title
Passed	1	62	Save items to cart test
Failed	1	62	Save items to cart test
Blocked	2	63	Save items to cart to persist after log off
Not applicable	2	63	Save items to cart to persist after log off
Active	3	70	Save items to cart
Active	3	70	Save items to cart



# Exploratory and Manual Testing

## Testing applications

Azure Test Plans provides test runners to run tests for your web and desktop applications.



The screenshot displays the Azure Test Plans interface. On the left, a table lists tests within a suite titled 'Test suite: Tests for Cart Management (Suite ID: 61)'. The table has columns for Outcome, Order, ID, and Title. A 'Run' button is visible in the toolbar. On the right, a 'Microsoft Test Runner - Google Chrome' window is open, showing a test titled '62: Save items to cart test'. The test steps are listed, and a 'Pass, fail, or pause tests' button is visible. A 'Screen capture, image action log, screen and voice recorder' window is also open, showing a list of actions. Red arrows point from text labels to specific UI elements: 'Run tests' points to the 'Run' button; 'Edit steps inline' points to the test steps; 'Web runner for web app testing' points to the Chrome window; and 'Pass, fail, or pause tests' points to the corresponding button in the runner window.

Outcome	Order	ID	Title
Active	1	62	Save items to cart test
Active	1	62	Save items to cart test
Active	2	63	Save items to cart to persist after log off
Active	2	63	Save items to cart to persist after log off

Test suite: Tests for Cart Management (Suite ID: 61)

Microsoft Test Runner - Google Chrome

https://phanindratelukunta.visualstudio.com/SampleProje

62: Save items to cart test

1. click on 'Save to cart' button
2. Click on 'View Kart' button

EXPECTED RESULT  
The items saved should be seen here

Windows 8

Web runner for web app testing

Pass, fail, or pause tests

Screen capture, image action log, screen and voice recorder

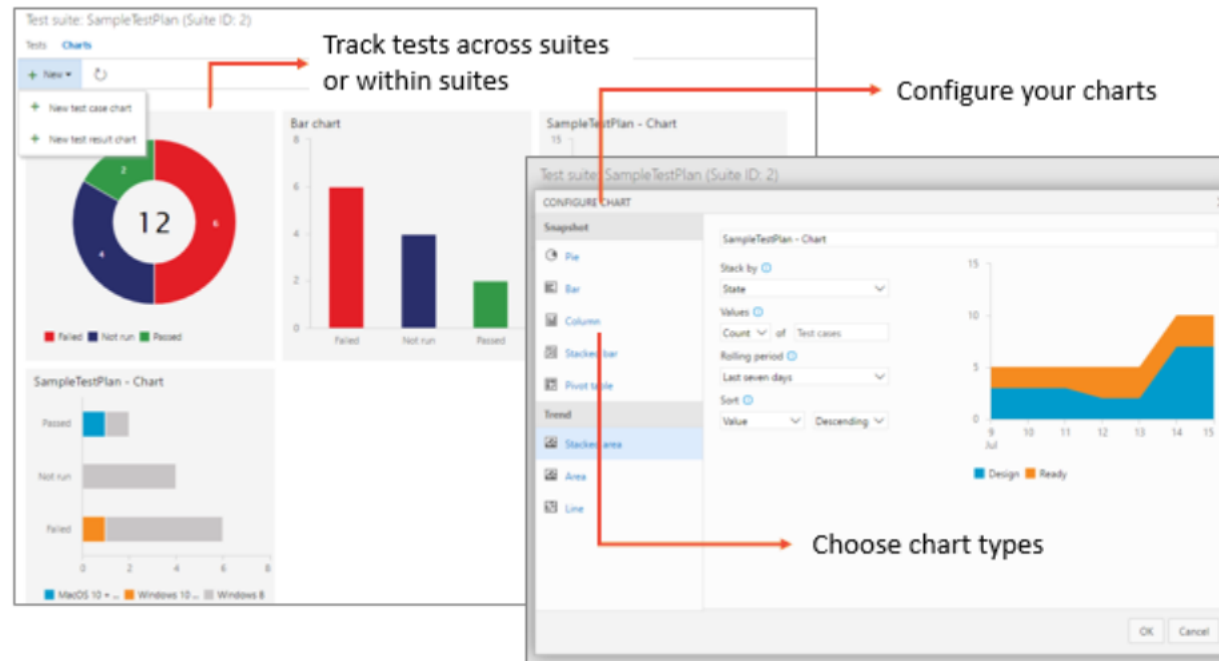
Edit steps inline

Run tests

# Exploratory and Manual Testing

## Test tracking

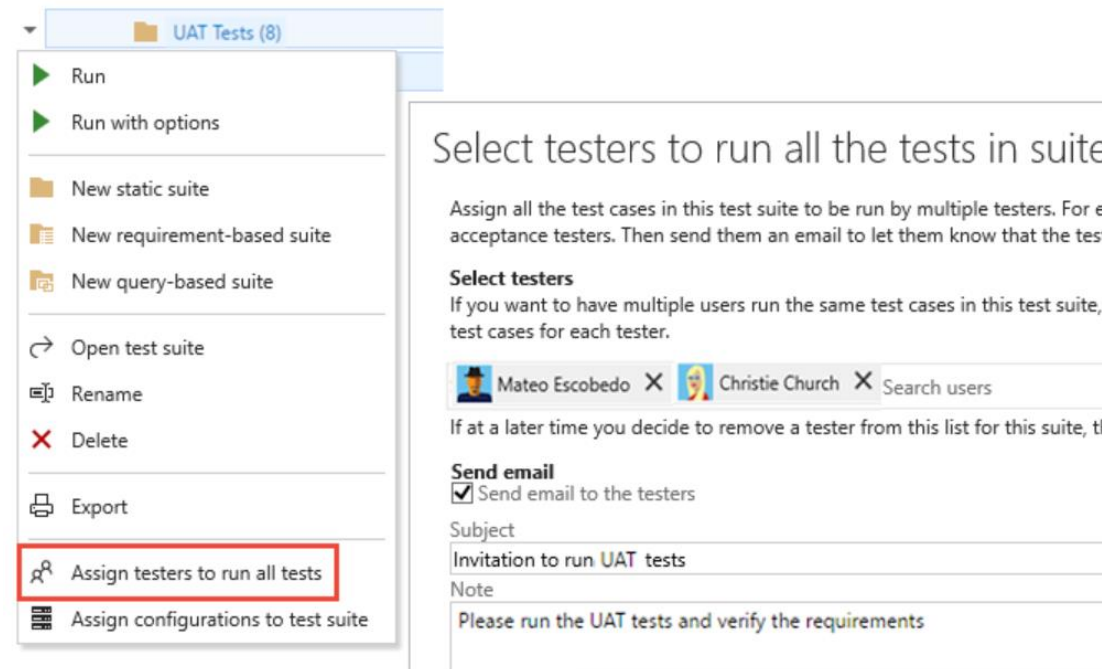
Quickly configure lightweight charts to track your manual test results using the chart types of your choice, and pin the charts to your dashboard to easily analyze these results.



# Exploratory and Manual Testing

## User acceptance testing

UAT is a key factor in software development that ensures the value requested by customers is being delivered by the engineering team.

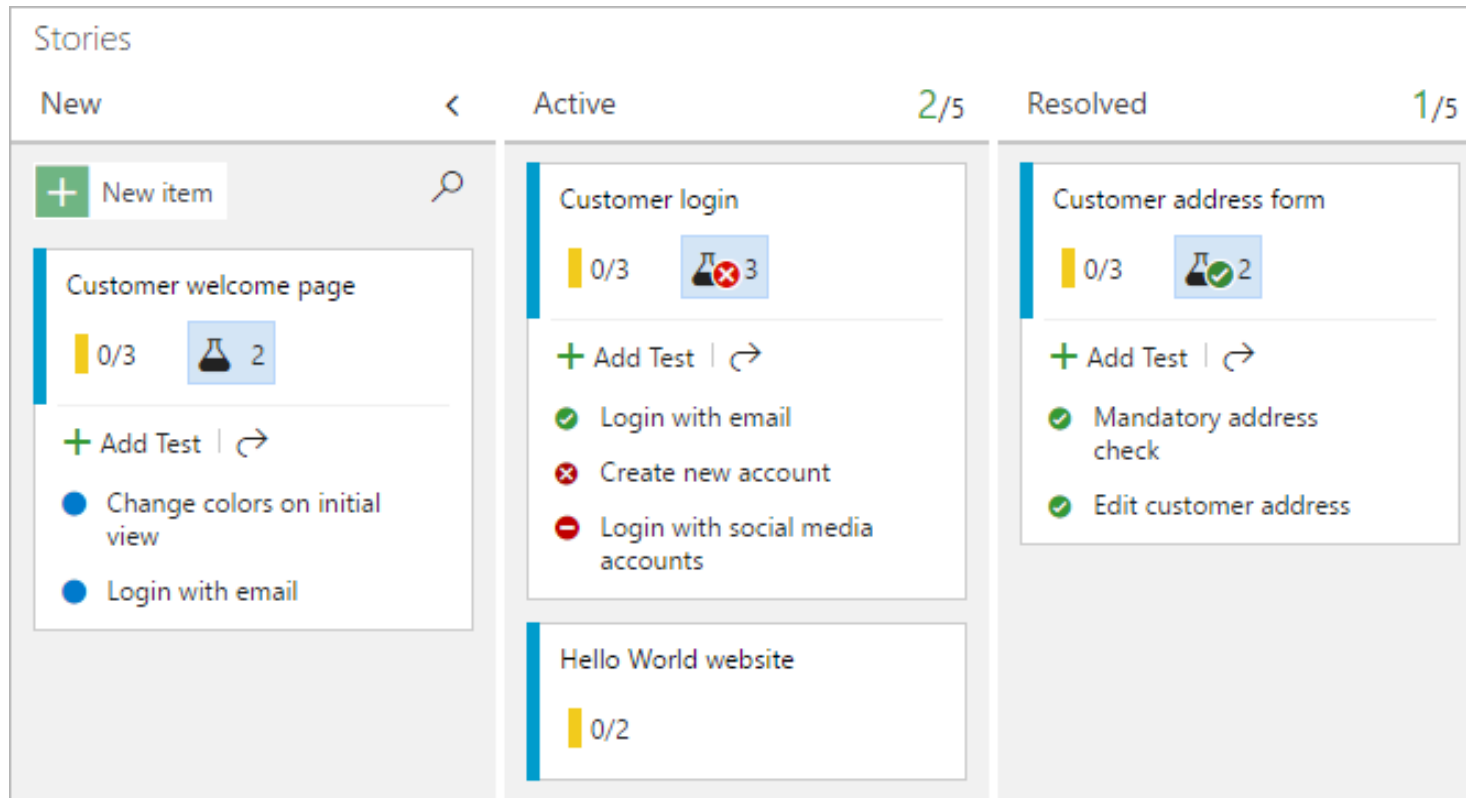




# Test from Kanban Boards

# Test from Kanban Boards

Add, view, and interact with test cases directly from the cards on the Kanban board, and then progressively monitor status directly from the card.



The screenshot displays a Kanban board with three columns: New, Active, and Resolved. Each column contains cards for user stories, each with a progress bar, a test case icon, and a list of test cases.

- New Column:**
  - Card: Customer welcome page (0/3 tests, 2 tests shown)
    - + Add Test | ↗
    - Change colors on initial view
    - Login with email
- Active Column (2/5 items):**
  - Card: Customer login (0/3 tests, 3 tests shown)
    - + Add Test | ↗
    - ✓ Login with email
    - ✗ Create new account
    - ✗ Login with social media accounts
  - Card: Hello World website (0/2 tests)
- Resolved Column (1/5 items):**
  - Card: Customer address form (0/3 tests, 2 tests shown)
    - + Add Test | ↗
    - ✓ Mandatory address check
    - ✓ Edit customer address

# Hands-on: Tests from Kanban Boards

# Hands-on: Manual Testing

# Hands-on: Exploratory Testing



**India: +91-7022374614**

**US: 1-800-216-8930 (TOLL FREE)**



**[support@intellipaat.com](mailto:support@intellipaat.com)**



**24/7 Chat with Our Course Advisor**