



Fourth

ECLIPSE TRACTUS-X COMMUNITY DAYS

May 22–23, 2025

STUTTGART



SUPPORTED BY:



Possehl Digital Services





Testing in Tractus-X



 Tractus X

May 22nd, 2025

Harald Zimmer (doubleSlash)
Test Management for Catena-X e.V.





Agenda

Together we will walk you through some insights of testing in Tractus-X



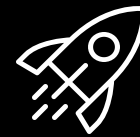
1. Introduction

- Tractus-X release and testing cycle
- Testing principles



2. Status and Challenges

- Test scope involving many services
- News about the test management tool stack
- Challenges encountered in a year of releases



3. Outlook

- Goals and planned improvements



4. Get Involved

- Good examples and lessons learned from the community
- Questions and Answers



No prerequisites, you don't need to know about the functionality of a particular Tractus-X project or how it is tested

Our Data Space Expertise



Catena-X Association & Tractus-X

Establishing a test management framework for Catena-X reference implementations and business applications. Active Eclipse Tractus-X contributions to multiple projects.



Former Catena-X Consortium

Development of Trace-X FOSS App, Item Relationship Service (IRS), and EDC Connector according to Catena-X standards.



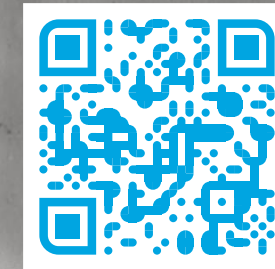
Operating Company

Maturity assessment of existing Catena-X reference implementations. Development and operationalization of the Trace-X App.



Data Space Onboarding

Enabling companies for data spaces – from onboarding and component setup to realizing impactful use cases.



Follow
doubleSlash
on LinkedIn





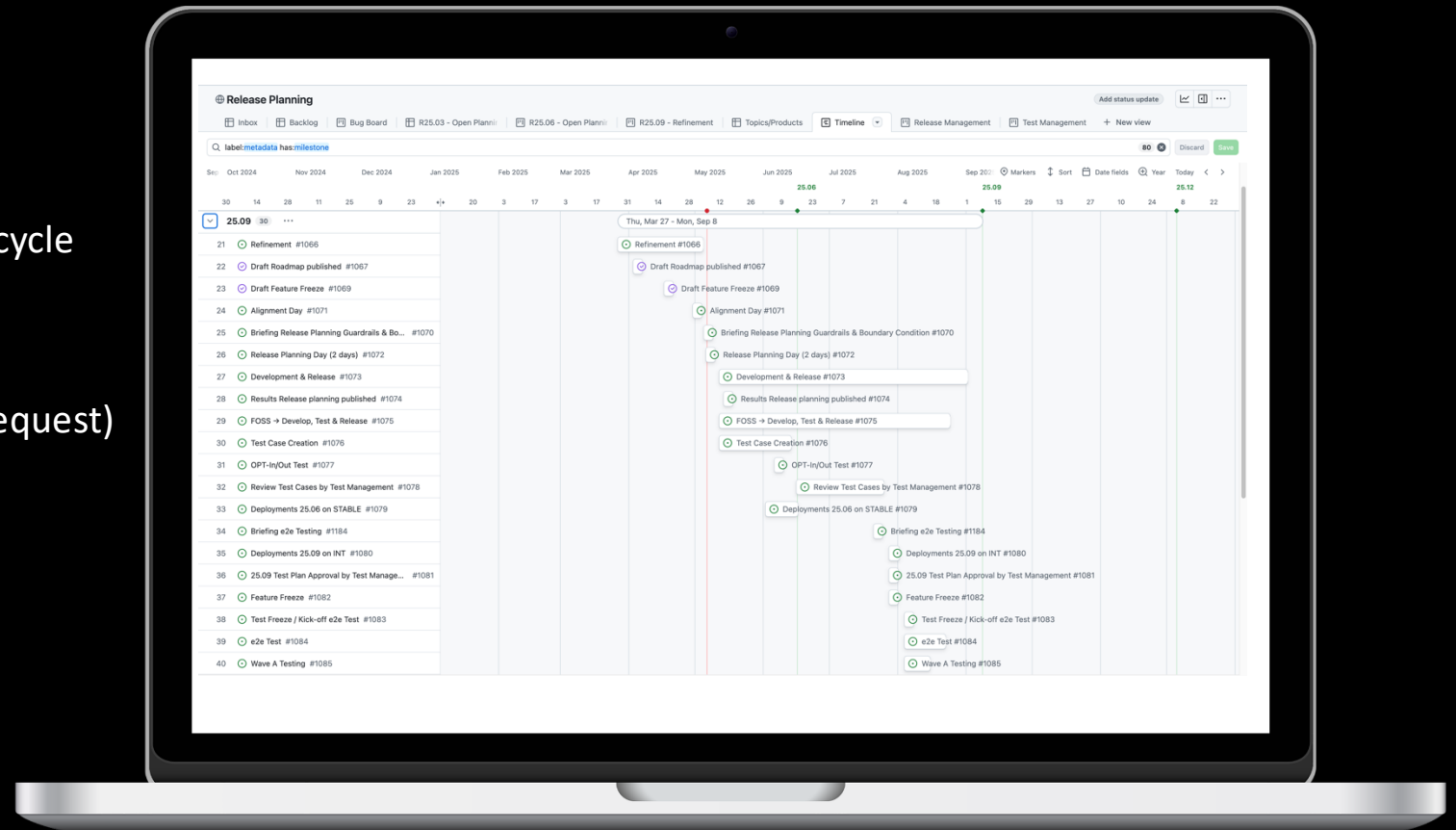
1. Introduction to Tractus-X Testing





Release Phases Timeline

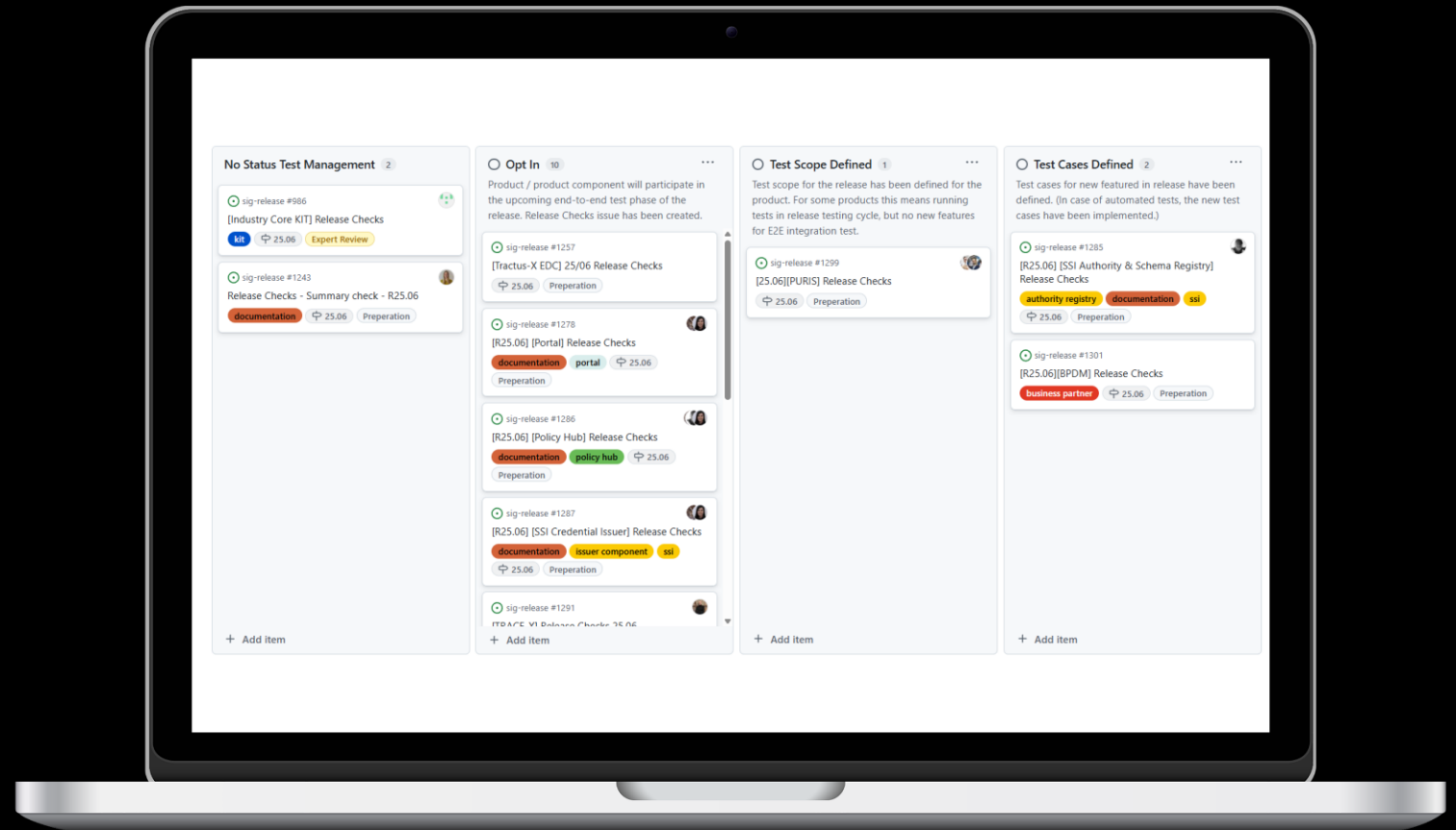
- ❧ Ca. 12 weeks of development per release, in increments of 2 weeks
- ❧ Individual product tests during development cycle with scope depending on team
- ❧ Mandatory automated scans (daily, per pull request) for example to identify vulnerabilities
- ❧ Ca. 3 weeks of end-to-end integration testing





Product Release Check Issues

- ✂ Tracking issues per product and per release
 - ✂ Versions that are tested in release
 - ✂ Mapping of new features and tests
 - ✂ Other quality checks
- ✂ This is also used as basis for the Change Log maintained by Release Management





High level approach: Testing integrated in the Process



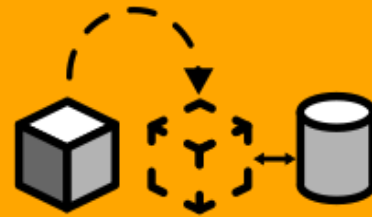
Build



Test
in Pipeline



Provide
Infrastructure



Deploy

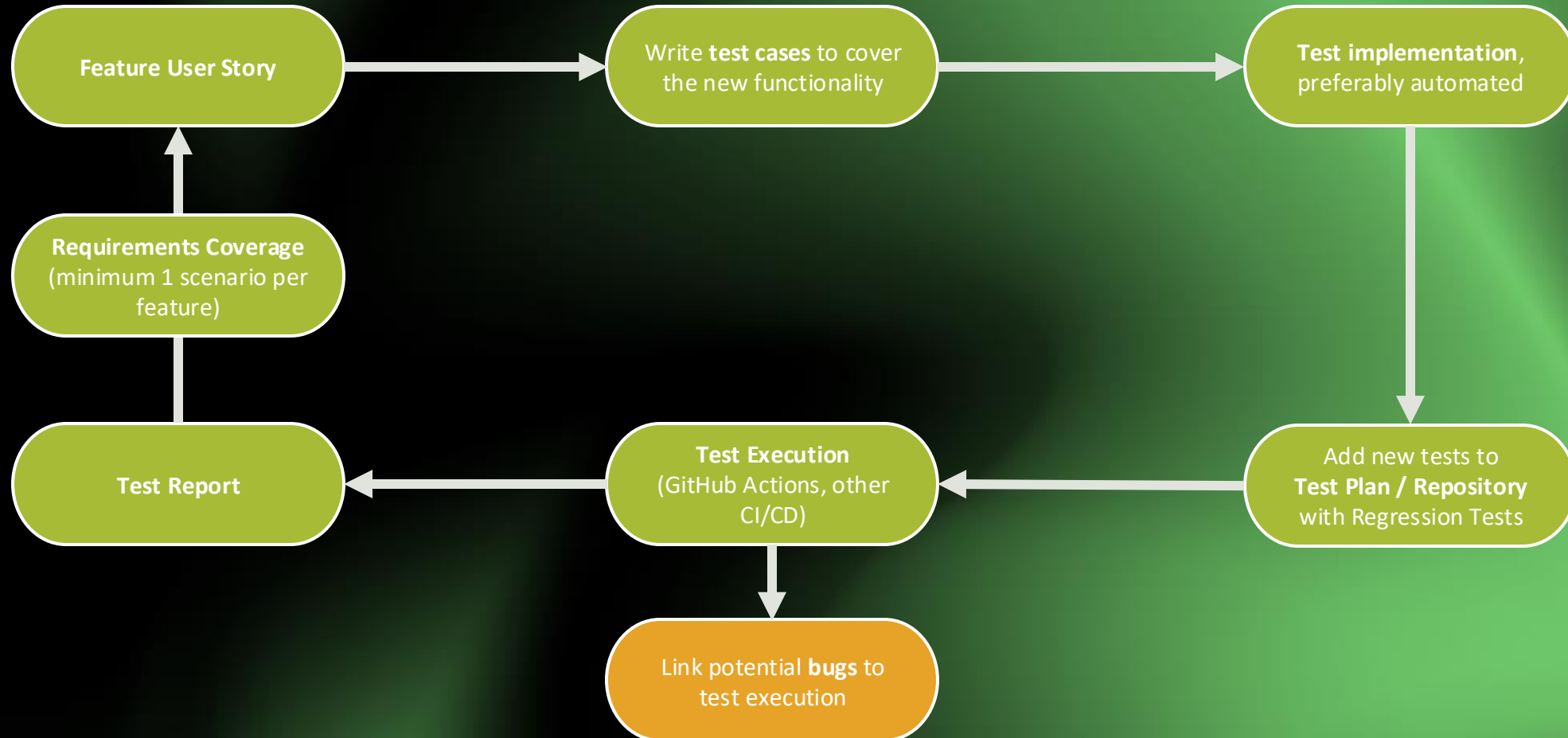


Test
Integration





Creating a closed test cycle





2. Status and Challenges





Test Scope: A complex environment of many services

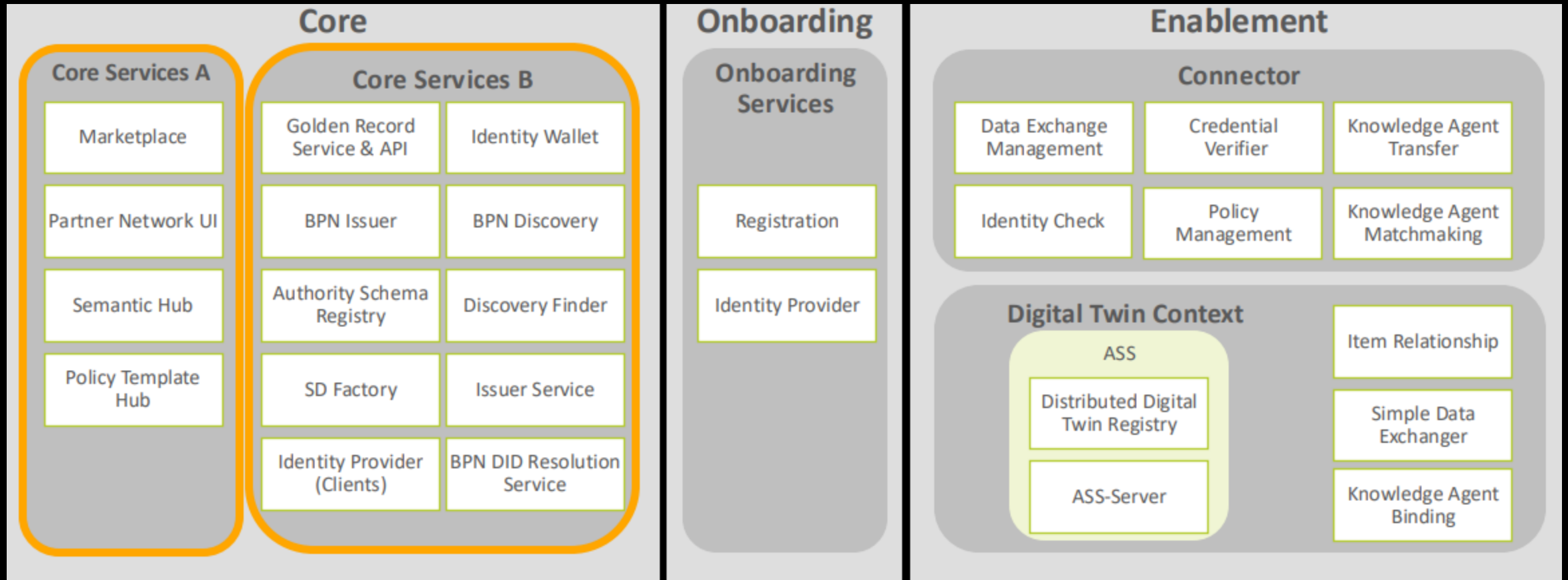


Image: Service Map from Catena-X Operating Model. Some services now archived, while others not yet show.



Test Methods and Outcomes

20+ components
updated and tested

200+ E2E tests executed

At least 68 bugs found and
resolved in last release 25.03

Manual and Automated Tests

Security & Vulnerability Scans

Dependency Checks,
Licensing Checks, others





Test tool stack with some proprietary applications - but accessible to everyone now

⌘ Really good news: Last week an **open-source license** was granted by Atlassian for Jira + Xray! A big **thank you** to the Eclipse Foundation and everyone involved!

⌘ This will help us to get a better shared view and tool support for test planning, test execution, and test reporting together





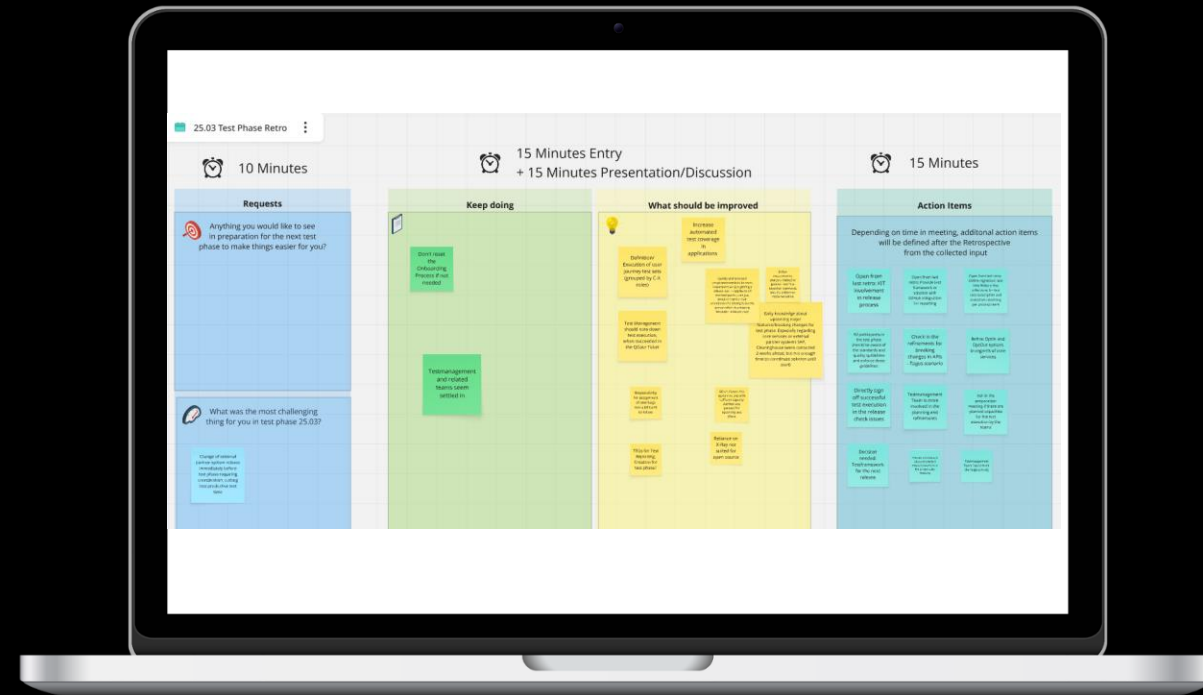
Challenges (1/2)

❌ Complex test scope:

Unlike testing a single application, this is an environment of core & enablement services and business applications having to interact with each other, developed by different teams and contributors

❌ Dependencies:

- ❌ Reliance on some key shared components that are also in active development in the same release cycle
- ❌ As well as to external shared services
- ❌ Potentially differing challenges from release to release if a key system has a breaking change that was not anticipated soon enough
- ❌ This can pose issues for deployments, service configuration on environment, and the testing itself





Challenges (2/2)

❌ Test case development:

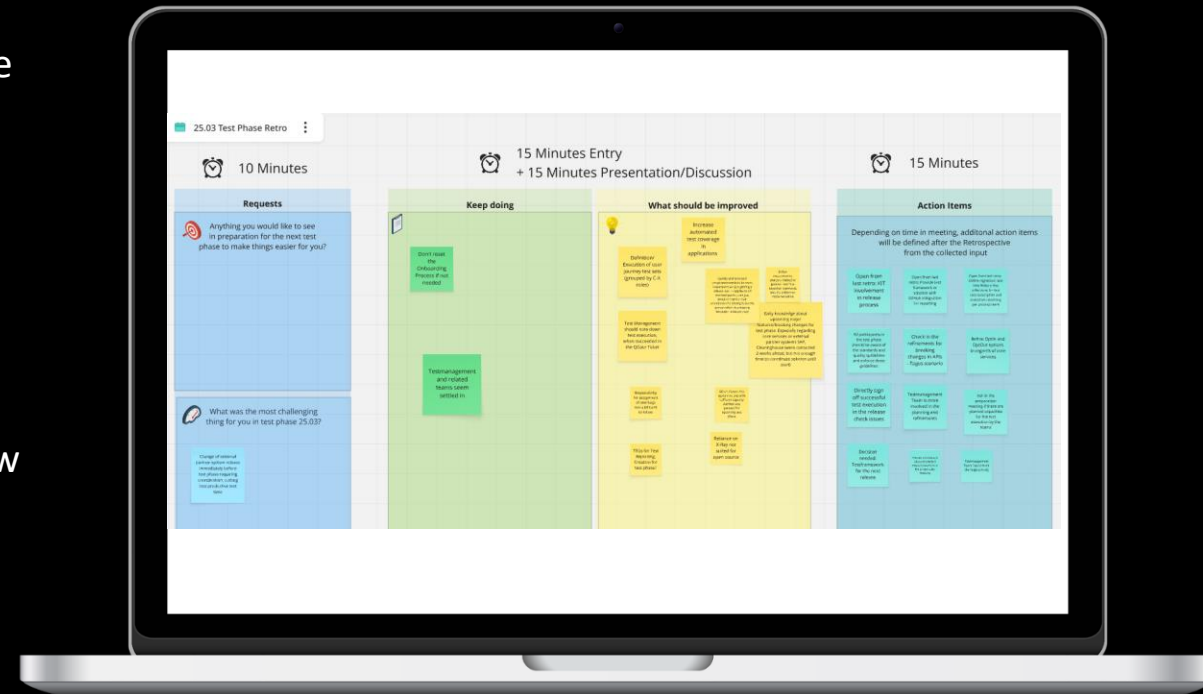
- ❌ Significant effort needed to create and maintain tests, while also developing the feature itself
- ❌ Functional and technical expertise required to define and implement test cases

❌ Test time:

- ❌ A mix of automated and manual integration testing
- ❌ Thus, testing is still a relatively slow process to cover all new features and conduct full regression tests

❌ Traceability:

Not all new functionality issues can always be fully anticipated by planned tests (standard/happy path, alternative paths, negative scenarios, ...)





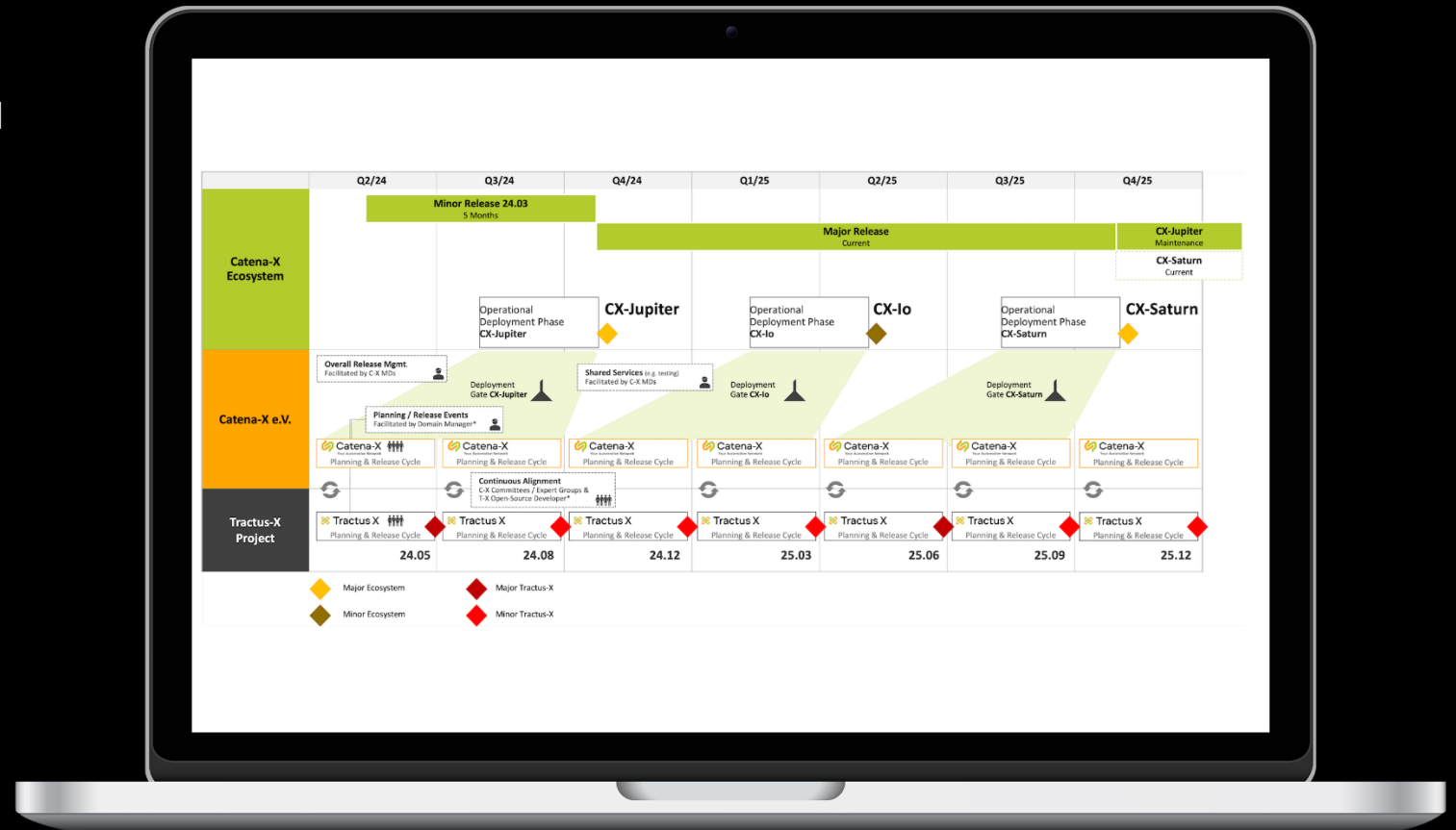
⌘ Tractus X

3. Outlook:
Goals and steps to improve



Goal 1: Ensuring Compatibility between Release Versions

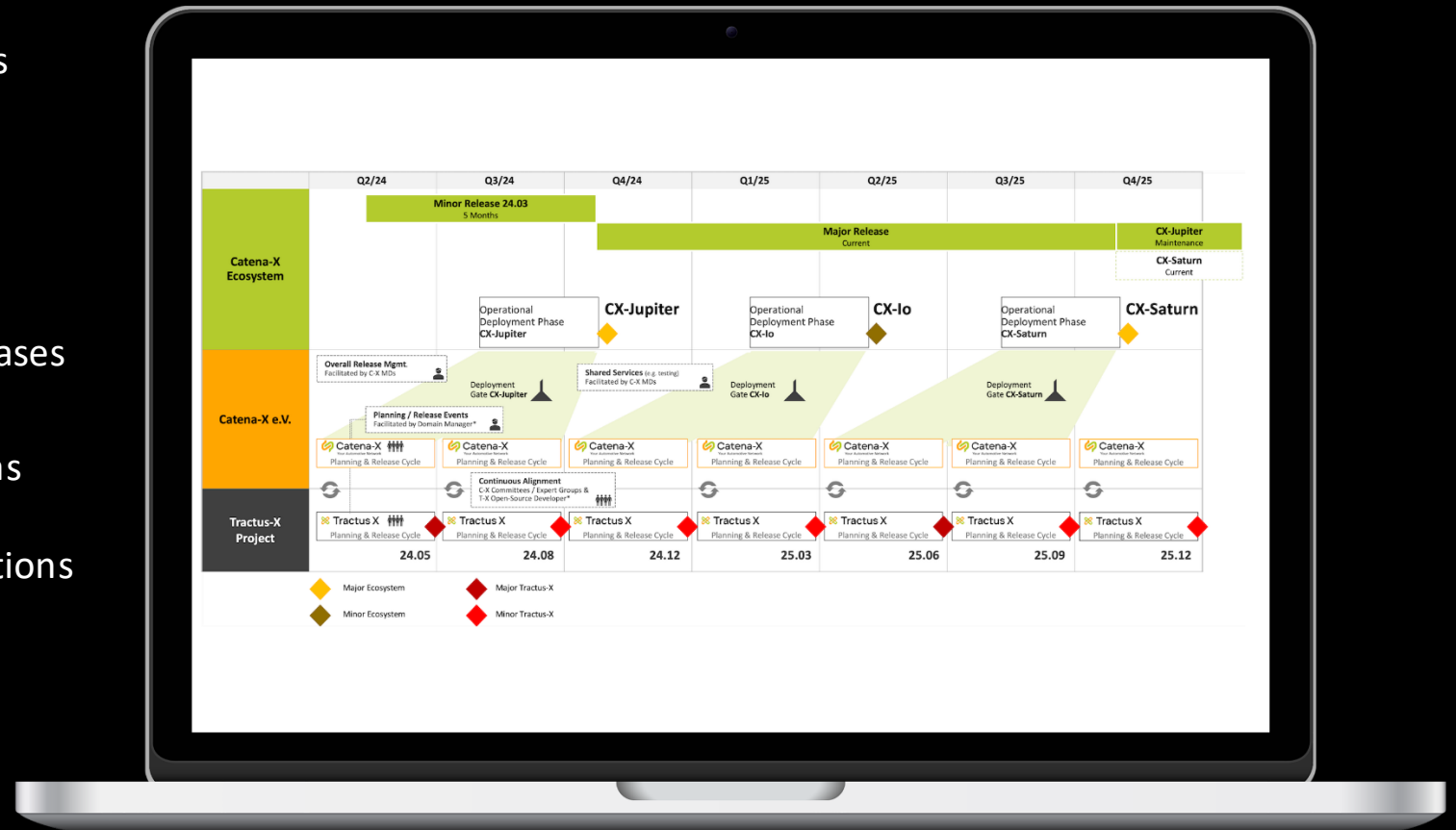
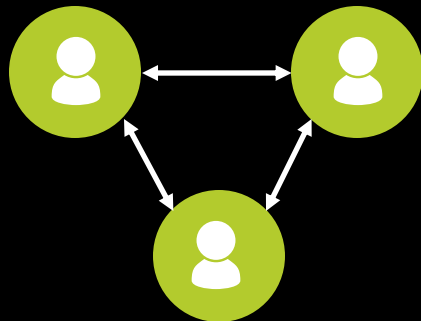
- ⌘ **Compatibility tests** so that multiple ecosystem releases can be operated in parallel
- ⌘ Ensuring future releases have a high quality also in different configurations, for example **Clearinghouse** or **Identity Wallet** providers depending on the market, to support the growth and internationalization of the ecosystem





Goal 2: Further Collaboration between Upstream – Downstream Testing

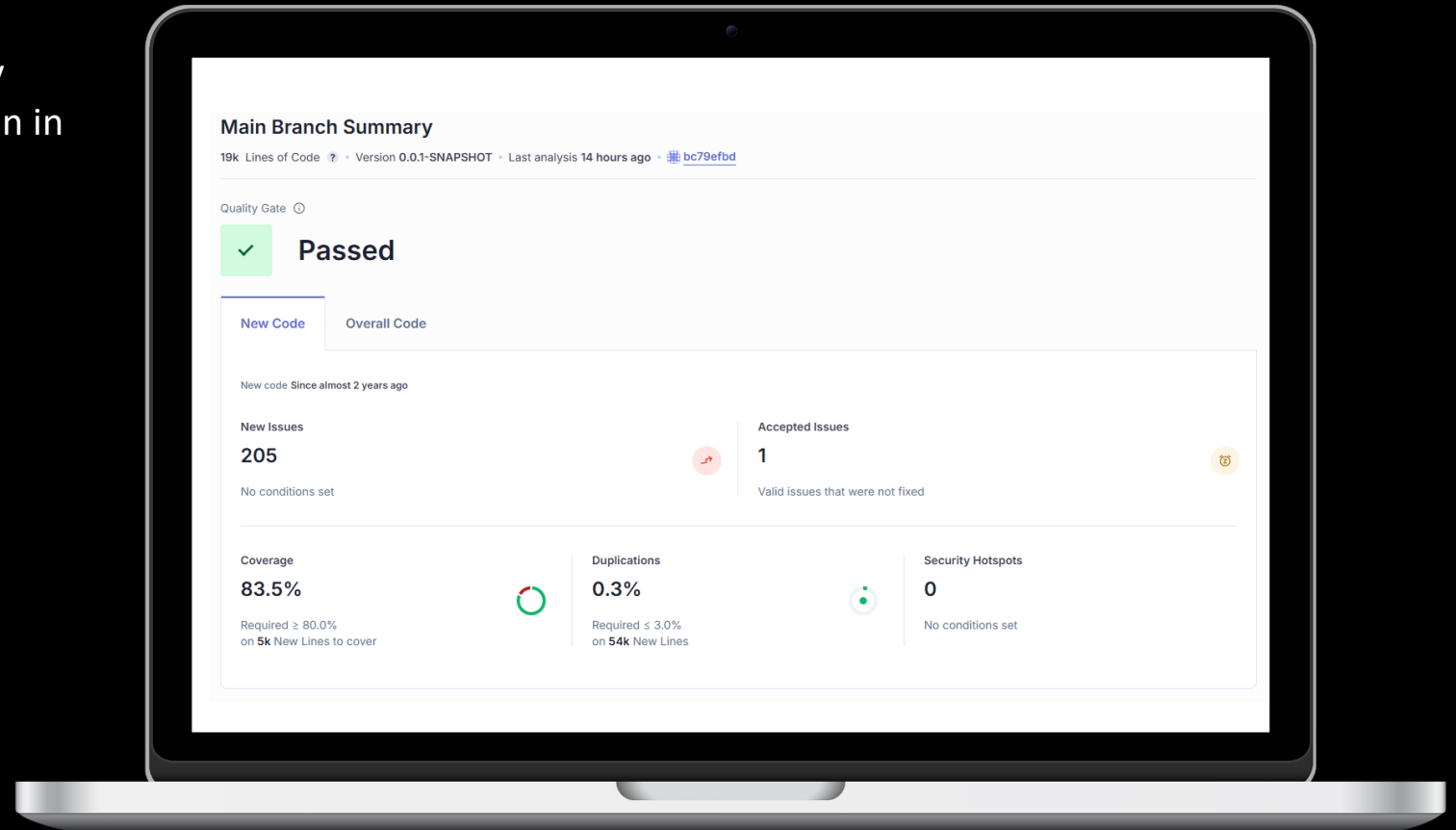
- ✂ In multiple formats we are fostering exchanges to improve testing outcomes involving:
 - ✂ Tractus-X community of developers and other experts
 - ✂ Catena-X e.V. as the standard provider
 - ✂ Cofinity-X as operator of productive releases
- ✂ Objective: To identify and prevent / fix issues as early as possible in the release cycles, while observing legal limits of the different organizations





Goal 3: Utilizing Testing Metrics for Test Reporting

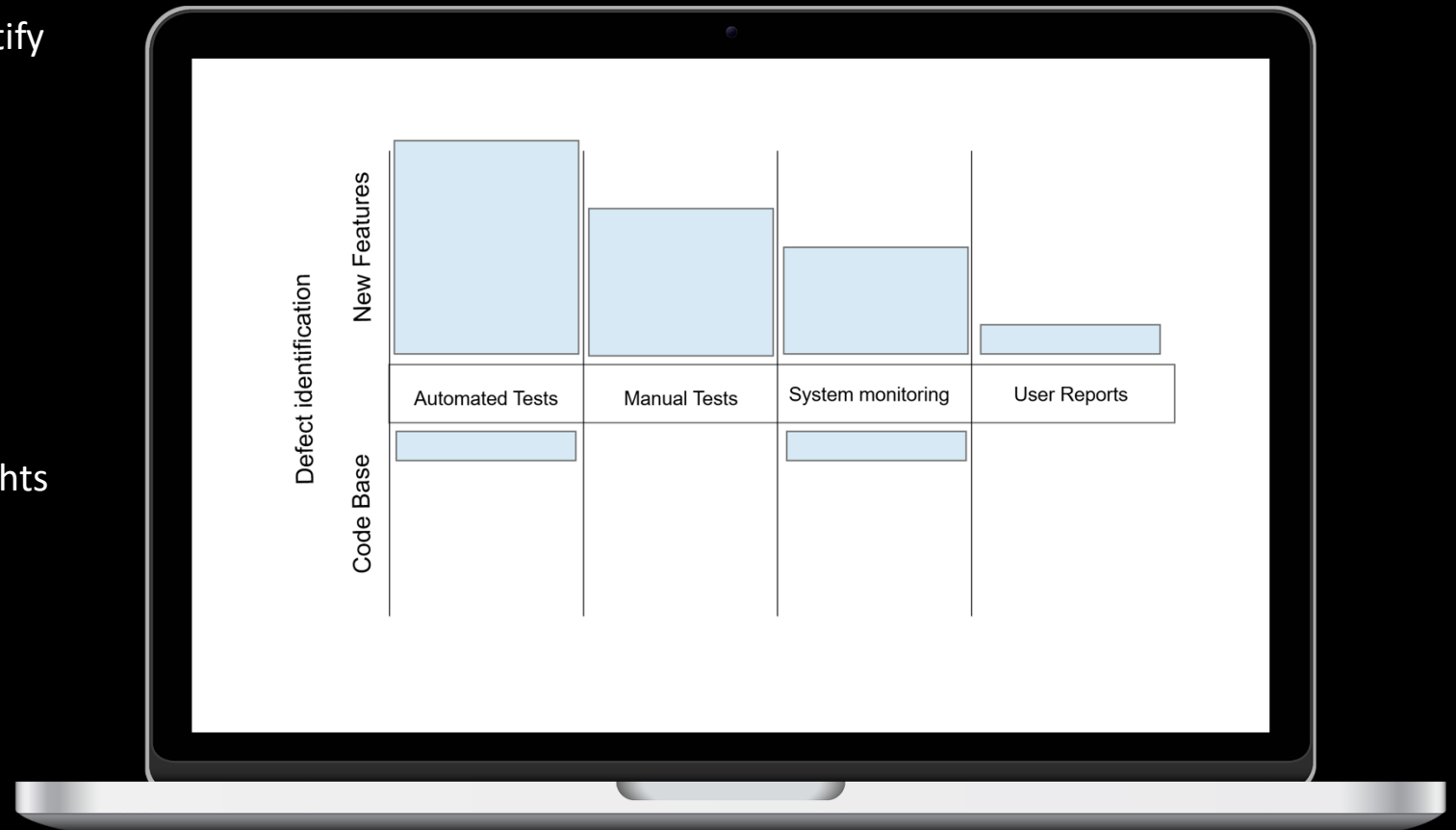
✂ Example: Code Coverage, Duplications, Security Hotspots from an application's main branch scan in Eclipse Tractus-X SonarQube





Testing Vision: Defect Funnel

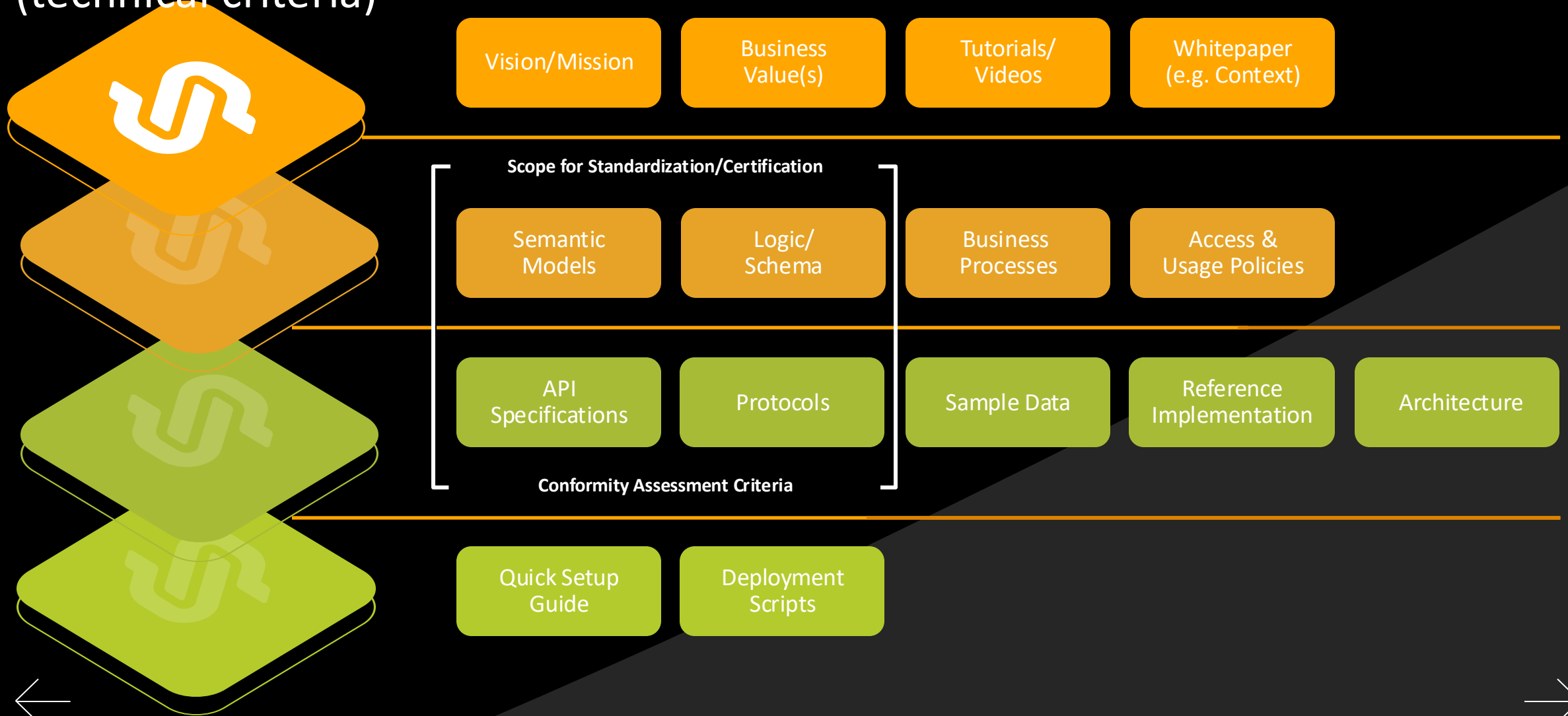
- ⌘ **Automated tests** (Test-Driven Design) to identify issues in new features as early as possible
- ⌘ **Manual tests** to uncover issues that were not anticipated prior to implementation of features and tests
- ⌘ **System monitoring** additions to maintain technical stability and generate business insights
- ⌘ Only a few remaining issues identified e.g. via User Acceptance Tests



Idealized illustration, adapted from “Fail Fast, Move On”, Michael Küsters, 2020



Exploration: Automated testing of standards for Conformity Assessment (technical criteria)





Outlook: Incremental improvements that increase quality



Automation

- ✂ Reduce manual efforts especially for repetitive regression testing to shorten test cycles
- ✂ Improve lead time for changes
- ✂ Then more focus can be given to (new) functionality that creates customer value



Security + Quality

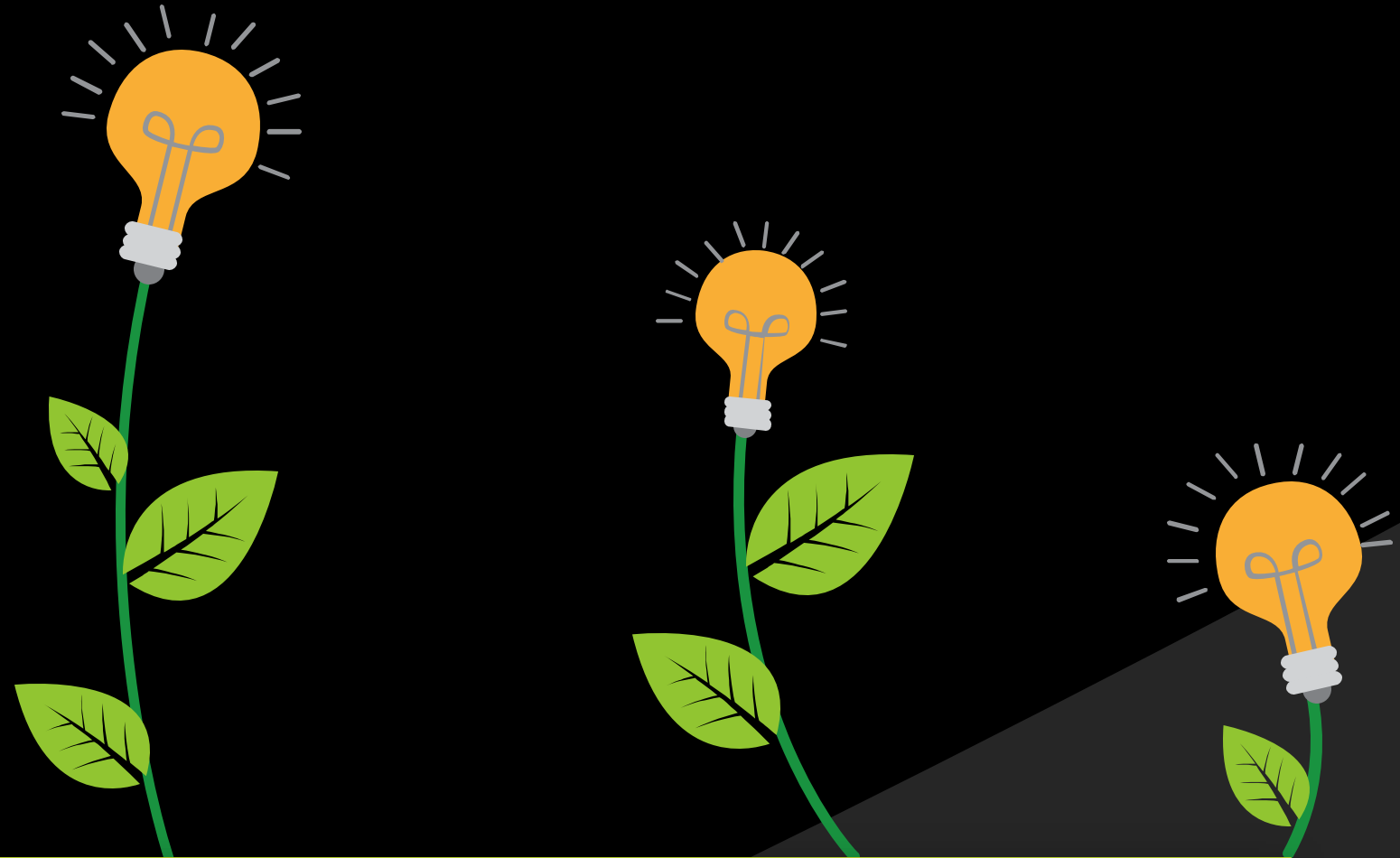
- ✂ Building on the existing Actions for automated scans, as defined in the Tractus-X Release Guidelines
- ✂ User journey testing to ensure key functionality for all user roles in the ecosystem



Sustainable for the long run

- ✂ Involvement of the whole Eclipse Tractus-X Community and definition of clear responsibilities, to keep quality high every day.
- ✂ Testers and Test Management involved early in the development cycle of upcoming features





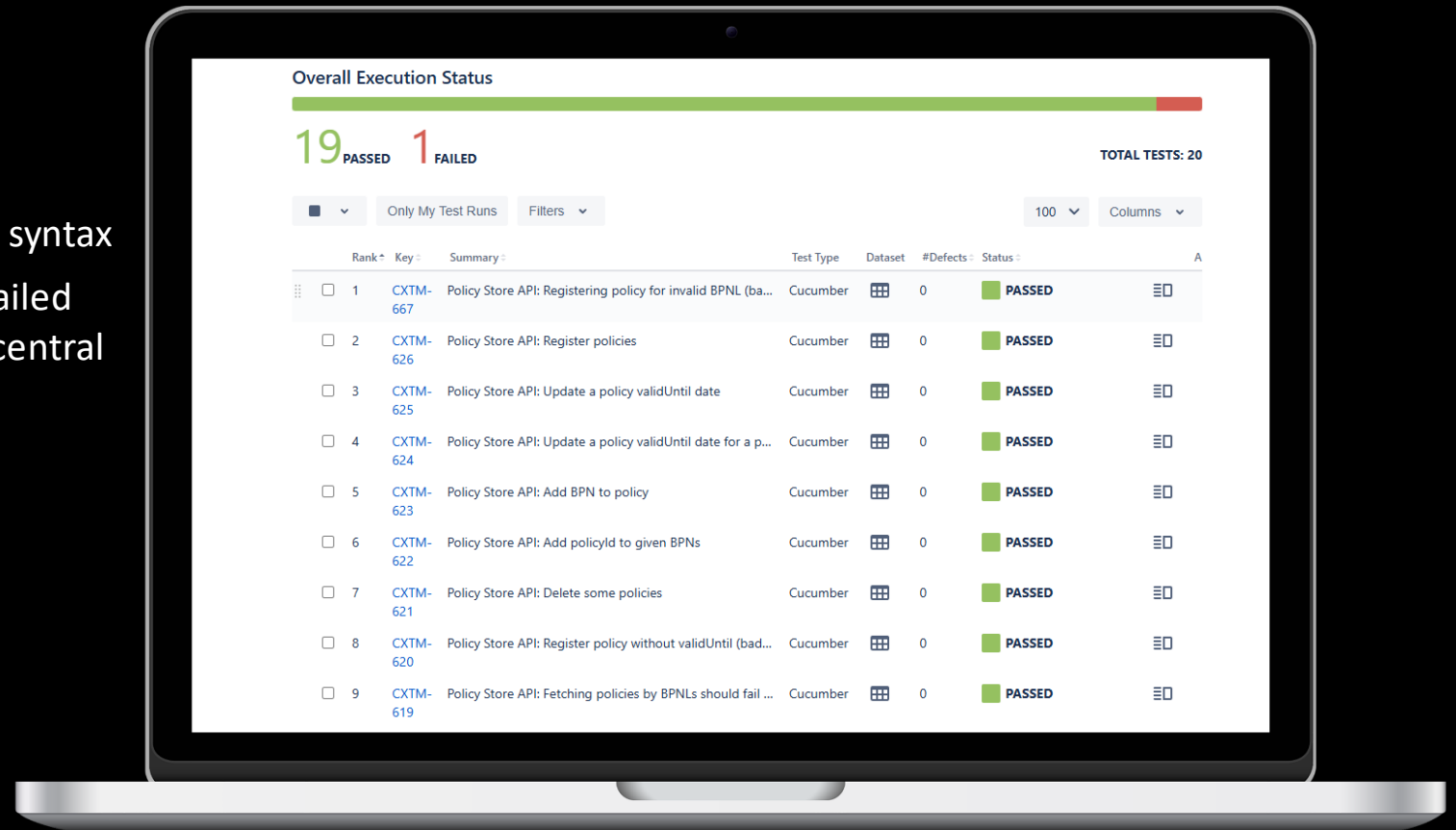
4. Get involved: Real-life examples from the community





Example 1: Automated API Testing

- ❧ Teams for Trace-X and IRS have implemented a number of automated tests for their regression
 - ❧ Test cases described in Cucumber Gherkin syntax
 - ❧ GitHub Actions Runner, and import of detailed test results (down to each step) back in a central test management system
 - ❧ Running nightly, or on request
- ❧ Similar API tests using various test tools and frameworks have been implemented by other Tractus-X projects





Example 2: Automated UI E2E Testing for Portal

- ✂ Ongoing Frontend UI test automation initiative, lead by contributions from Cofinity-X
 - ✂ Using the Cypress framework
 - ✂ Cross-browser testing possible, which is difficult to do when only testing functionality manually
 - ✂ Built-in retry functionality for flaky tests
 - ✂ Enabling parallelization of test runs
- ✂ Additional automated API tests for Portal Backend using other frameworks





Example 3: Demand and Capacity Management KIT

- Expert group defined test sets and applicable user journeys for the upcoming 25.06 release
- Companies implementing this KIT to build a DCM application now have a good template for their testing



Testing a DCM application

Having a well organized and documented testing of a DCM application is advantageous not only for the whole development process, but especially because test-runs can be used as proof for required certification of the application.

It is recommended to compartmentalize tests into test-sets from which different user journeys can be built, depending on which specific capabilities are supposed to be tested by the user journey. Below you will find tests, test-sets and user journeys. Please be aware that they do not provide full coverage of the standard and depending on how you build your application you will need to expand on them.

List of Test-Sets

Test-Set	Description	Tests in test-set
Customer: Prepare yourself	Prepares customer for the user journey, by setting up EDC, data assets, wallet, certificates etc.	<ul style="list-style-type: none">- Setup EDC- Register APIs as assets- Check wallet for certificates- Prepare variables for other tests- User journey specific preparation
Supplier: Prepare yourself	Prepares supplier for the user journey, by setting up EDC, data assets, wallet, certificates etc.	<ul style="list-style-type: none">- Setup EDC- Register APIs as assets- Check wallet for certificates- Prepare variables for other tests- User journey specific preparation
Customer: Create WeekBasedMaterialDemand	Tests the implementation of the aspect model	<ul style="list-style-type: none">- Create valid aspect model- Create invalid aspect model- Create user journey specific aspect model



General lesson learned: Keep tests focused on use cases

We would like to encourage new users to buy in our shop. Therefore we offer 10% discount for their first order.

```
public void CalculateDiscount(Order order)
{
    if (order.Customer.IsNew)
        order.FinalAmount =
            Math.Round(order.Total * 9/10);
}
```

Register as "bart_bookworm"
Go to "/catalog/search"
Enter "ISBN-0955683610"
Click "Search"
Click "Add to Cart"
Click "View Cart"
Verify "Subtotal" is "\$33.75"

We would like to encourage new users to buy in our shop. Therefore we offer 10% discount for their first order.

Given the user has not ordered yet

When the user adds a book with the price of EUR 37.5 into the shopping cart

Then the shopping cart sub-total is EUR 33.75.

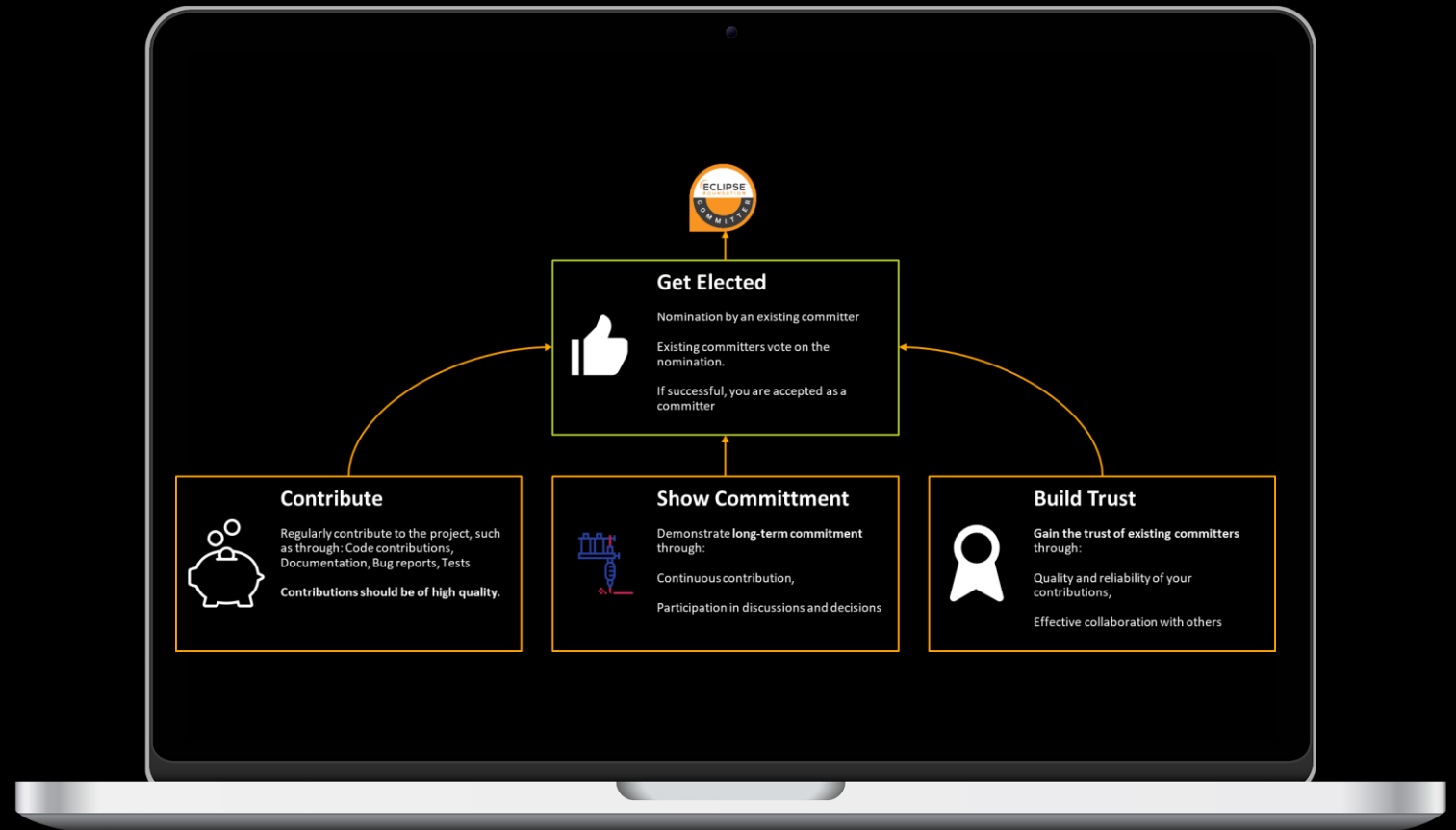
- ✂ Write test scenarios rather in a **plain non-technical language** to create a common basis
 - ✂ All stakeholders from business, development and testing can understand the test case and participate in its definition.
 - ✂ The test implementation then deals with the actual technology behind a functionality or system interface.





Become a Contributor, become a Committer, and please also become a Tester 😊

- ⌘ Bug reports, tests and documentation improvements are valuable contributions, and they help on the path of becoming an Eclipse Tractus-X committer!
- ⌘ Contribute to test case creation and maintenance either directly via a project, or for example via an expert group that helps all future projects implementing a KIT / use case
- ⌘ Participate in developing and testing of your desired features in an upcoming release cycle, to ensure they are implemented in high quality





Thank you!

A computer monitor displaying the Tractus X logo and text. The logo is a colorful infinity symbol, and the text 'Tractus X' is in white. The background of the screen is dark with a grid pattern. The monitor is on a desk with an orange keyboard and mouse.

 Tractus X

Contacts:

monika.jacobsen@catena-x.net
(Test Manager for Catena-X)

harald.zimmer@doubleslash.de

[Test Management Channel in the Eclipse Tractus-X Matrix Chat](#)