

Software Testing

OMP

University of Antwerp (2025)

Who am I?

Tars Joris

- **1997-2001: Master in Computer Science: University of Antwerp**
- **2001-2006: Developer at eVision: Software for the furniture industry**
- **2006-2023: Inventive Designers / Unifiedpost**
 - 2006-2008: Developer at Inventive Designers
 - 2008-2019: Development Manager at Inventive Designers
 - 2020-2022: Team Lead at Unifiedpost
 - 2022-2023: Delivery Manager at Unifiedpost
- **2023-now: Team Manager at OMP**

Supply Chain Planning

... is the forward-looking process of coordinating assets to optimize the delivery of goods, services and information from supplier to customer, balancing supply and demand.

- 💡 Make sure enough raw materials are available
- 💡 Reorder production runs to reduce setup times
- 💡 Move products in time to avoid expensive transport means



Industry solutions on a generic framework



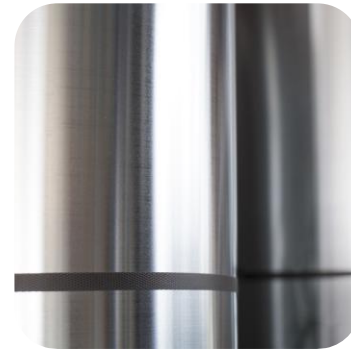
Chemicals



Consumer goods



Life sciences



Metals



Paper, film
& packaging



Plastics
Glass
Building products
Tires

Customer references by industry

Paper & Plastics



Consumer Goods



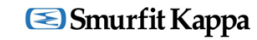
Life sciences



Metals



Packaging



Chemicals



Tires

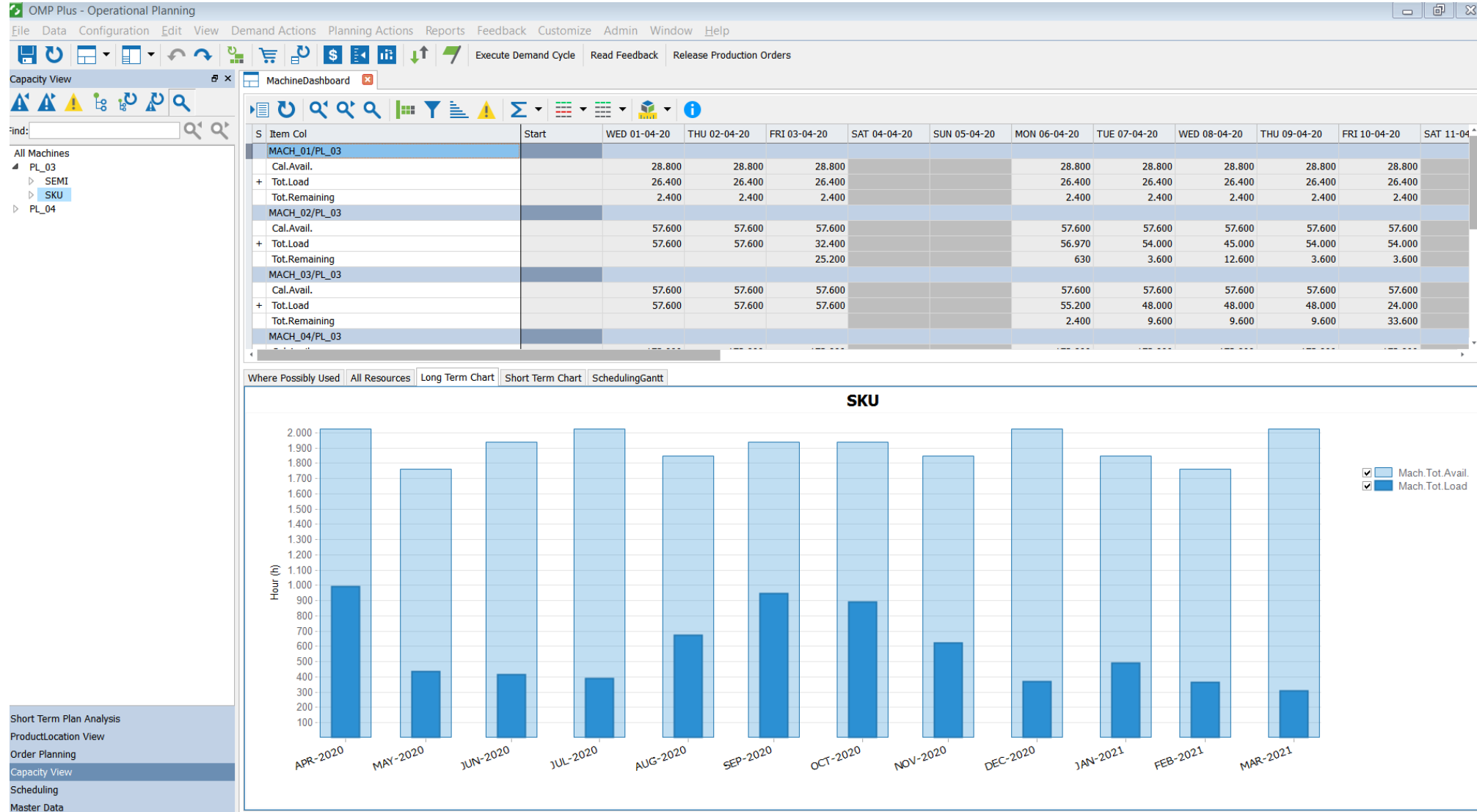


Building products



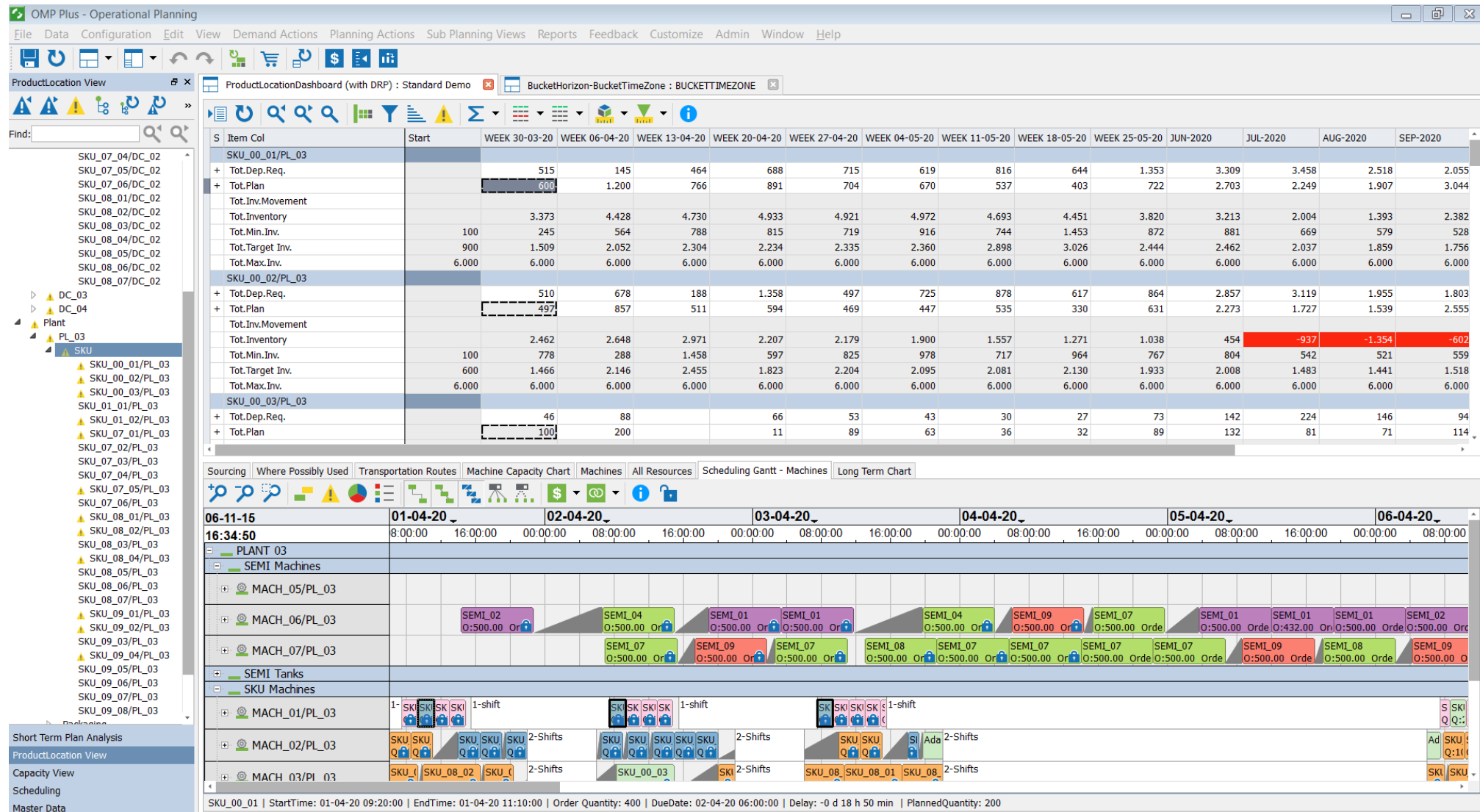
OMP desktop application

C++ / Qt



OMP desktop application

C++ / Qt



OMP Web application

React / Typescript – Azure cloud

Commercial Demand Management

Customer

KeyUser_Sales - 101 - EU - ALL

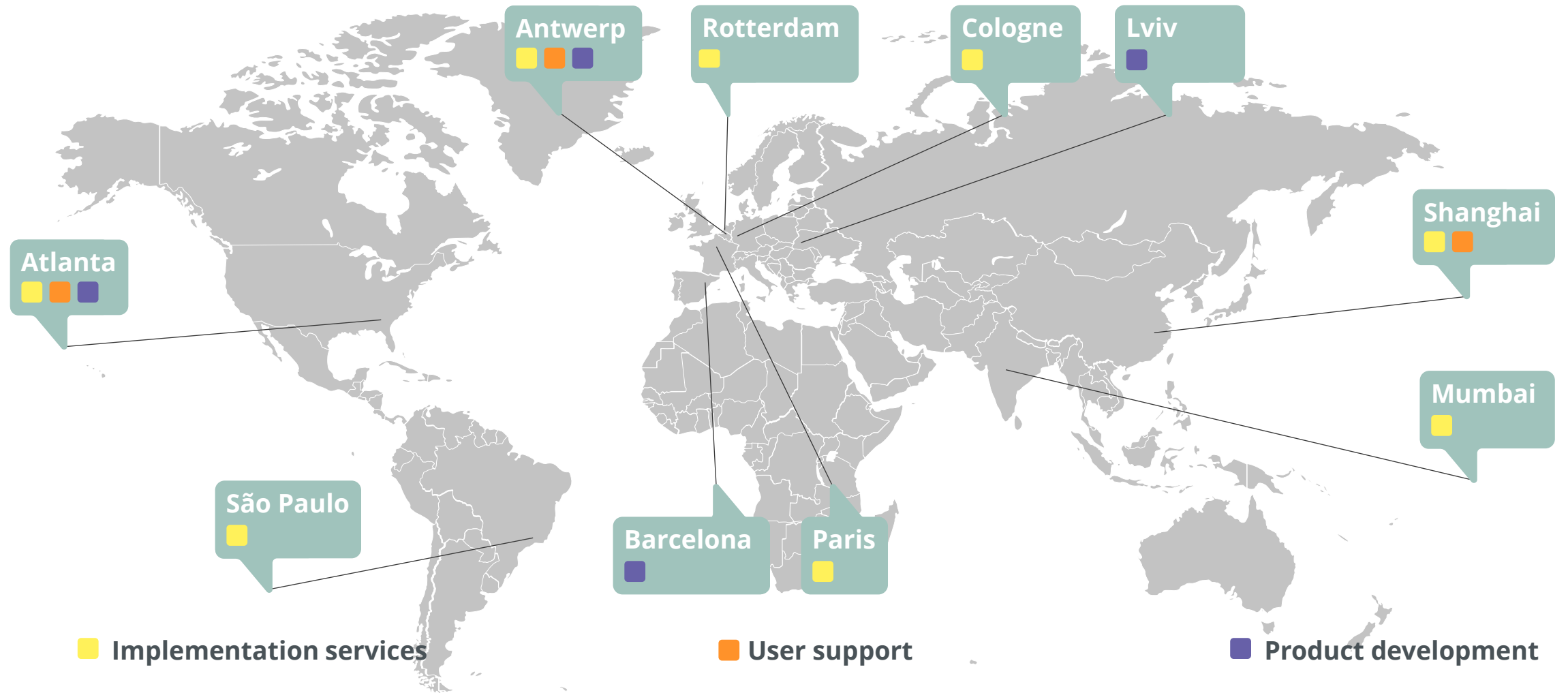
Sales forecast entry

Country group: All Product Group 9: All

Customer name: All Product Group 9: All Product description: All Summary line: Sales Base Forecast (KG)

	SEP-2023	OCT-2023	NOV-2023	DEC-2023	JAN-2024	FEB-2024
Total						
Sales History Year-1	23,854,168	22,414,826	23,864,505	20,089,369	23,857,627	23,538,050
Sales Base Forecast (KG)	<u>26,544,951</u>	<u>6,656,098</u>	<u>22,912,275</u>	<u>25,739,594</u>	<u>25,383,807</u>	<u>24,538,548</u>
Statistical Forecast	3,737,207	26,227,655	24,020,289	23,277,303	24,357,097	23,376,710
Customer names						
>	0	714	4,500	38,301	26,652	17,680
>	<u>5,610</u>	<u>1,068</u>	<u>1,762</u>	<u>5,739</u>	<u>5,753</u>	<u>5,396</u>
>	<u>54,000</u>	<u>12,298</u>	<u>4,859</u>	<u>55,440</u>	<u>58,342</u>	<u>51,863</u>
>	0	1	—	—	—	—
>	677	193	442	862	880	842
>	9	1	10	10	2	10
>	<u>12,000</u>	<u>14,752</u>	<u>40,714</u>	<u>114,900</u>	<u>52,105</u>	<u>49,513</u>
>	0	0	—	—	—	—
>	0	0	—	—	—	—
>	<u>5,623</u>	<u>1,796</u>	<u>5,623</u>	<u>5,811</u>	<u>5,811</u>	<u>5,436</u>
>	<u>23,237</u>	<u>15,000</u>	<u>24,840</u>	<u>17,762</u>	<u>21,253</u>	<u>25,720</u>
>	0	0	—	—	—	—
>	74	21	74	76	76	71
>	<u>1,755</u>	0	<u>1,723</u>	<u>1,780</u>	<u>1,780</u>	<u>1,665</u>
>	2	15	32	15	3	2

Scaling close to our customers





Unit Tests



Unit Tests (white box)

C# XUnit

```
[Theory]
[InlineData(@"a\%param2%\b", @"a\\b")]
[InlineData(@"a\%param1~%\b", @"a\\b")]
0 | 0 references | Tars Joris, 46 days ago | 1 author, 1 change
public void ResolveVariables_WhenVariableIsUnkown_ShouldReplaceWithEmptyString(string input, string expectedOutput)
{
    // Arrange
    var EnvironmentVariableDictionary? dictionary = new EnvironmentVariableDictionary
    {
        ["param1"] = "val1"
    };

    // Act
    var string? result = VariableResolver.ResolveVariables(input, dictionary);

    // Assert
    Assert.Equal(expectedOutput, result);
}
```

Unit Tests (white box)

TypeScript/React: Jest/Enzyme libraries

```
it('shows the task history when invoking the onShowTaskHistory context callback', () => {
  const task = defaultProps.tasks.get('task1');
  const wrapper = shallow(<ResourcePromotion {...defaultProps} />);
  wrapper.find('ContextProvider').prop<any>('value').onShowTaskHistory(task.task.id);
  expect(wrapper.find('TaskHistoryFetcher').prop('task')).toBe(task);
});

it('hides the task history after pressing the sidebar close button', () => {
  const wrapper = shallow(<ResourcePromotion {...defaultProps} />);
  wrapper.setState({ ...wrapper.state, shownHistoryItemId: 'task1' });
  wrapper.find('Sidebar').find('Button').simulate('click');
  expect(wrapper.find('Sidebar').prop('visible')).toBeFalsy();
  expect(wrapper.find('TaskHistoryFetcher').exists()).toBeFalsy();
});

it('hides the task history if the task no longer exists in the list of tasks', () => {
  const wrapper = shallow(<ResourcePromotion {...defaultProps} />);
  wrapper.setState({ ...wrapper.state, shownHistoryItemId: 'task1' });
  wrapper.update();
  wrapper.setProps({ ...defaultProps, tasks: defaultProps.tasks.filter(t => t.task.id !== 'task1') });
  expect(wrapper.find('Sidebar').prop('visible')).toBeFalsy();
  expect(wrapper.find('TaskHistoryFetcher').exists()).toBeFalsy();
});
```

Unit Tests (white box)

TypeScript/React: Jest library snapshot functionality

```
2. node
FAIL __tests__/Link.react-test.js
  ● renders correctly

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot 1.

    - Snapshot
    + Received

    <a
      className="normal"
    - href="http://www.facebook.com"
    + href="http://www.instagram.com"
      onMouseEnter={ [Function] }
      onMouseLeave={ [Function] }>
    - Facebook
    + Instagram
    </a>

    at Object.<anonymous> (__tests__/Link.react-test.js:14:16)

    x renders correctly (10ms)

Snapshot Summary
  > 1 snapshot test failed in 1 test suite. Inspect your code changes or press `u` to update
```



Contract Tests



Contract Tests

Web API testing with Postman

The screenshot shows the Postman interface for a collection named 'CAT Pentest 2025 April'. The left sidebar lists the collection's structure, including folders like 'Restore environment', 'Install build', and 'Create plan', and individual requests like 'GET Get items', 'GET Get instances', 'POST Start instance', 'GET Wait until instance started', and 'POST Create plan'. The 'GET Get items' request is selected and expanded in the main panel.

The request details are as follows:

- Method:** GET
- URL:** `{{STAGING_API_BASEURL}}/items?user={{testuser}}`
- Params:** (None visible)
- Authorization:** (None visible)
- Headers (7):** (None visible)
- Body:** (None visible)
- Scripts:** (Visible, showing the Post-response script)
- Settings:** (None visible)

The **Post-response** script contains the following JavaScript code:

```
1 pm.test("Check response code", function() {
2   pm.expect(pm.response.code).to.eql(200);
3 });
4 pm.test("Check response body", function() {
5   var jsonBody = pm.response.json();
6   pm.expect(jsonBody).to.have.property("value");
7   var value = jsonBody.value;
8   pm.expect(value).to.be.an("array");
9   pm.expect(value).length.above(1);
10 });
```

Contract Tests

Web API testing with Postman

CAT Pentest 2025 April - Run results ERROR Run Again Automate Run + New Run Export Results

Ran today at 10:59:04 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	CAT Sandbox	1	3m 36s	49	2292 ms

All Tests Passed (45) Failed (4) Skipped (0) [View Summary](#)

GET Get install messages

https://app-cat-api-stagingservice1.azurewebsites.net/api/ca...

200

PASS

 Check response code

PASS

 Check response body

GET Get items

https://app-cat-api-stagingservice1.azurewebsites.net/api/ca...

200

PASS

 Check response code

PASS

 Check response body

GET Get instances

https://app-cat-api-stagingservice1.azurewebsites.net/api/ca...

200

1

GET CAT Pentest 2025 April / Create plan / Get items

Response

Headers

Request

200 · 1458 ms · 2.49 KB

Pretty

```
1 {
2   "value": [
3     {
4       "type": "menu",
5       "label": "Advanced Download Configurator",
6       "parentMenuItems": [
7         "Utilities",
8         "Configuration",
9         "Solution Template For SAP"
10      ],
11      "id": "Utilities|Configuration|Solution"
```




End-to-End Tests



End-to-End Tests (black box)

Selenium: Web UI test framework

- Simulate actual usage of the application
- Starts the browser and interacts with the web pages
- Depend on external systems
- Build OO model of your pages
 - Easier to reuse code
 - Leads to more readable tests

```
@Test
public void testSaveCopy() throws Exception
{
    final EditBlockPage editBlockPage = new EditBlockPage();
    editBlockPage.open("blocks/Formal%20salutation.slb");
    EditBlockPage.SettingsTab blockSettingsTab = editBlockPage.selectTabSettings();

    blockSettingsTab.waitForPath("blocks/Formal salutation.slb");

    // change name and save as a new block
    blockSettingsTab.setName(1, "another");
    editBlockPage.saveAs();
    blockSettingsTab.waitForPath("blocks/another.slb");

    // make sure the original block still exists
    editBlockPage.open("blocks/Formal%20salutation.slb");
    blockSettingsTab = editBlockPage.selectTabSettings();
    blockSettingsTab.waitForPath("blocks/Formal salutation.slb");
    blockSettingsTab.assertName(1, "Formal salutation");

    // see if the new block can be found
    editBlockPage.open("blocks/another.slb");
    blockSettingsTab = editBlockPage.selectTabSettings();
    blockSettingsTab.waitForPath("blocks/another.slb");
    blockSettingsTab.assertName(1, "another");
}
```

End-to-End Tests (black box)

Browser Stack: Browsers as a Service to run Selenium tests

The screenshot displays the Browser Stack interface, which is used for running Selenium tests in a cloud environment. The interface is divided into two main sections: a left sidebar for project management and a main area for test execution details.

Left Sidebar (Project Management):

- Username and Access Keys:** A section for managing user credentials, with a "Show +" link.
- Parallel threads:** A section for managing the number of parallel threads, with a "Add more" link.
- Builds:** A list of builds with columns for "Completed", "Timeout", and "Errors". The builds are sorted by "Search Build or Session name".
- Builds List:**
 - Build: Build 865 - origin/master (11h ago)
 - Build: Build 4272 - origin/build/greenkeeper (9m ago)
 - Build: Build 4270 - origin/issue-OPS-1286-add-starter-pack-to-nexus (10h ago)
 - Build: Build 4269 - origin/build/greenkeeper (10h ago)
 - Build: Build 4268 - origin/issue-WBD-1767-fix-module-loading-errors-in-lambda (10h ago)

Main Area (Test Execution Details):

- Test Log:** A table showing the execution of Selenium tests. The columns are "Time", "Status", "Command", and "Element".
- Test Steps:**
 - 00:33 0 Find multiple elements `css selector=#property-dst.dropdown.active` ⇒ ELEMENT 27
 - 00:33 0 Send a sequence of key strokes to the active element. `s,i,r,m,p,l,e`
 - 00:33 1 Send a sequence of key strokes to the active element. `ENTER`
- Test Results:** A section showing the results of the test execution, including a screenshot of the web application and a video of the test execution.

End-to-End Tests (black box)

Qt

- **Qt: library that supports and enables cross-platform software development for embedded, mobile and desktop**
 - Rapid prototyping
 - Mainly focusing on UI, but a lot more than that (e.g. XML, threading, communication, ...)
 - We use it mainly for the UI in our fat client
- **Both available as licensed and free software**



End-to-End Tests (black box)

Squish - General



Squish

■ General overview

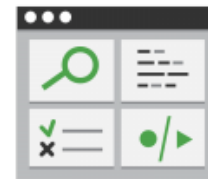
- Froglogic founded in 2003
- Squish: automated GUI (Graphical User Interface) testing tool
 - a SquishQt link component that hooks into the application to capture (UI) Qt calls and create the mapping
- Squish can be easily integrated in a CI tool (e.g. Jenkins)
- Closely linked to Qt, so upgrades need to be done for both tools together...



**Recording and
Playback**



**Object Map & Object
Identification Tools**



**Powerful and Intuitive
Test Creation
Environment**



**Test Verification &
Validation**

End-to-End Tests (black box)

Squish

Test Management - E:\Development\quality\Squish\ScrumTesting\FCT\last\EngineConfiguration\suite_ExpertSystem\tst_MethodCheck\test.js - Squish IDE 6.3.0

File Edit Source Refactor Navigate Search Run Window Help

Test Suites

suite_ExpertSystem

Test Cases

- ☒ tst_createfileenvironment
- ☒ tst_ExpertGlobalVsNode
- ☒ tst_ExpertPyramid
- ☒ tst_BaseFctChanges
- ☒ **tst_MethodCheck**
- ☒ tst_Pyramid
- ☒ tst_GlobalDialogChecks
- ☒ tst_NodeExpertMatrixCheck

Test Case Resources

Scripts Test Data VPs Gestures

- VP_Initial
- VP_Old

Test Suite Resources

Scripts Test Data VPs Gestures

tst_MethodCheck(suite_ExpertSystem)

```

37  mouseClicked(waitForObject(":OMFCTUI_SelectForecastConfigDialog.m_fctConfigComboBox_QComboBox"), 61, 10, 0, Qt.
38  mouseClicked(waitForObjectItem(":OMFCTUI_SelectForecastConfigDialog.m_fctConfigComboBox_QComboBox", "FctItemPro
39  clickButton(waitForObject(":OMFCTUI_SelectForecastConfigDialog.OK_QPushButton"));
40
41  clickButton(waitForObject(":MessageViewer.qt_dockwidget_closebutton_QDockWidgetTitleButton"));
42  mouseDrag(waitForObject(":MainWindow.qt_splithandle_DashboardControlsFrame_QSplitterHandle"), 618, 2, 0, -200,
43
44  clickButton(waitForObject(":DashboardControlsFrame.Run_Expert_System_QPushButton"));
45  snooze(3);
46  test.vp("VP_Initial");
47
48  mouseClicked(waitForObject(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QComboBo
49  mouseClicked(waitForObjectItem(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QCom
50  clickButton(waitForObject(":DashboardControlsFrame.Run_Expert_System_QPushButton"));
51  snooze(3);
52  test.vp("VP_Initial"); //Chart stays the same in this case
53
54  mouseClicked(waitForObject(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QComboBo
55  mouseClicked(waitForObjectItem(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QCom
56  clickButton(waitForObject(":DashboardControlsFrame.Run_Expert_System_QPushButton"));
57  snooze(3);
58  test.vp("VP_Old"); //Here chart changes
59
60  //Check of the rest of methods (without VPs)
61  mouseClicked(waitForObject(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QComboBo
62  mouseClicked(waitForObjectItem(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QCom
63  clickButton(waitForObject(":DashboardControlsFrame.Run_Expert_System_QPushButton"));
64  mouseClicked(waitForObject(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QComboBo
65  mouseClicked(waitForObjectItem(":OMFCTUI_NodeCategoryMethodSelectionWidget.m_methodSchemeSelectionComboBox_QCom

```

Test Results

Result	Message	Time
Log	Uploaded for object -ConflictDefProperty- : File E:\Development\quality\Squish\ScrumTesting\FCT\last\En...	Feb 4, 2019 10:22:45 AM
Log	Exiting application.	Feb 4, 2019 10:22:46 AM
Pass	Exit dialog appeared when performing exit.	Feb 4, 2019 10:22:46 AM
Log	Saving changes at exit.	Feb 4, 2019 10:22:46 AM
Pass	Application exited within 5 seconds.	Feb 4, 2019 10:22:48 AM
Log	--- Started to close possible crash dialogs...	Feb 4, 2019 10:22:48 AM
Log	Doing nothing - no window with ompplus_last* title found..	Feb 4, 2019 10:22:48 AM
Log	--- Done with closing possible crash dialogs.	Feb 4, 2019 10:22:48 AM

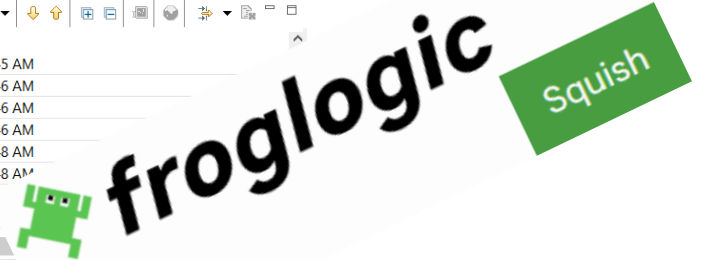
Recent Test Results

Test Management Test Debugging Spy

Global Scripts

Last (E:\Development\quality\Squish\ScrumTesting\FCT\last\EngineConfiguration\suite_ExpertSystem\tst_MethodCheck\test.js)

Last_1 (E:\Development\quality\Squish\ScrumTesting\FCT\last_1\EngineConfiguration\suite_ExpertSystem\tst_MethodCheck\test.js)



End-to-End Tests (black box)

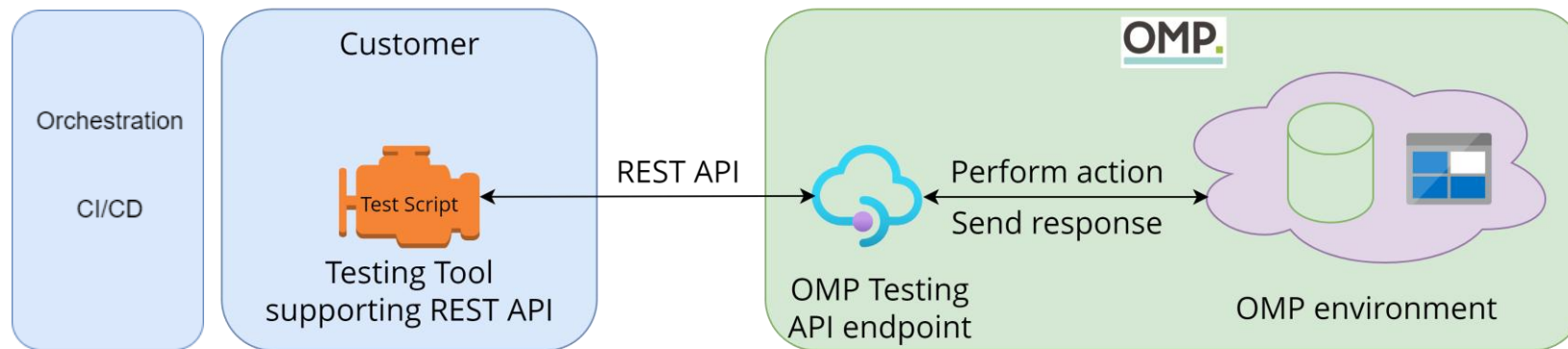
Squish learning points

- **Squish is very easy and intuitive to start with...**
 - ... but building maintainable and robust tests requires advanced knowledge
 - Scripting language
 - Complex interaction between Squish and our applications (e.g. synchronization issues)
- **Proper ICT infrastructure to deploy and maintain the tool in a multi-user environment is key to successful adoption**
 - Standardized machines (e.g. resolution, locale)
 - Automate upgrades as much as possible
- **Strong communication between developers and testers helps to reduce maintenance and debugging costs**
- **Screenshots require more effort to maintain**
 - Try to test specific things, with checks that are less brittle

End-to-End Tests (black box)

Customer application testing

- Solution to make **desktop application testable without** using **GUI testing tools** (like Squish)
 - Operates on the layer just below the User Interface
 - The actual application is started in the background
 - Application switches to **REST API** request listening mode
 - Instruct application to perform certain actions through REST APIs
 - Response allows verification in testing tool

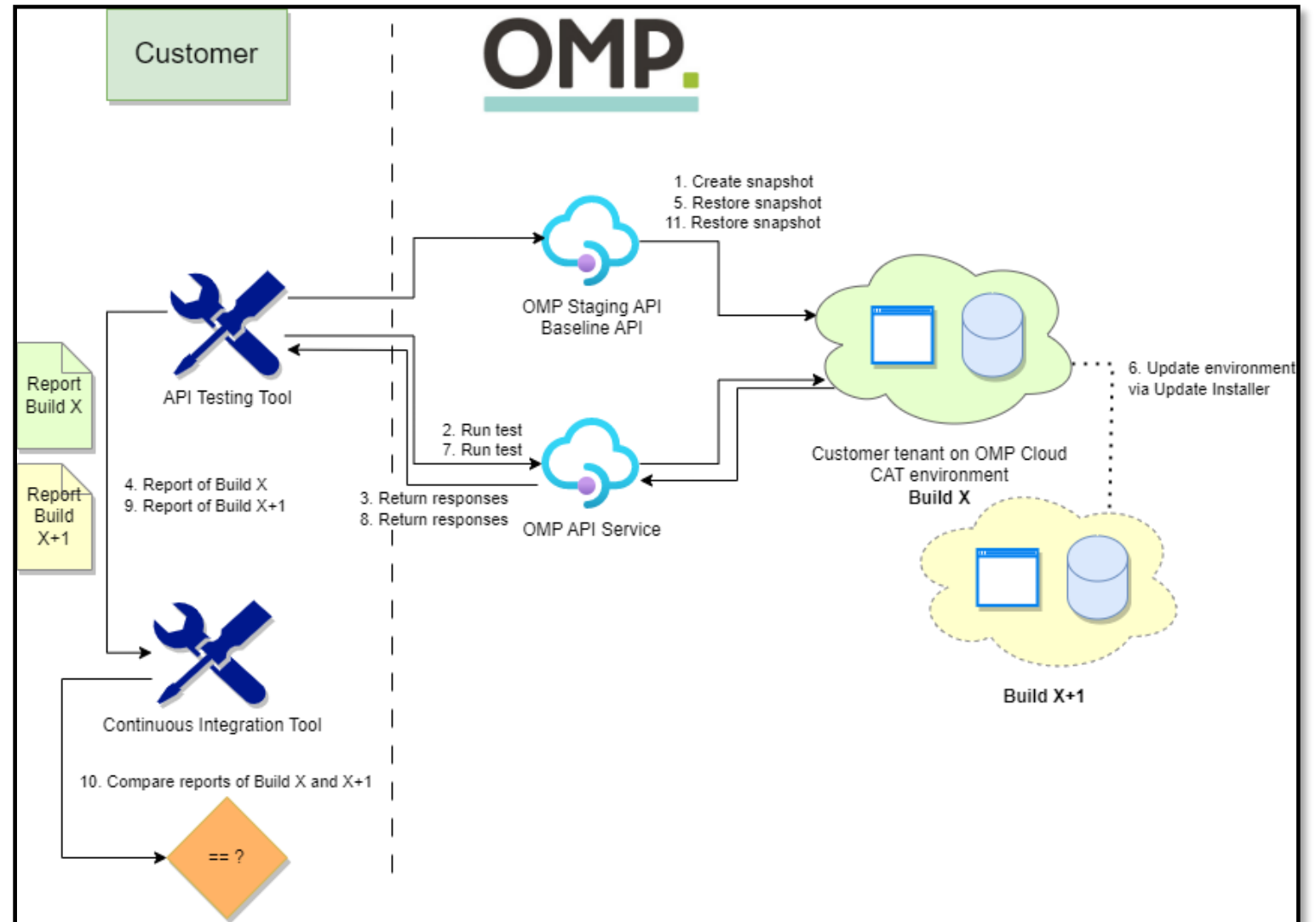


End-to-End Tests (black box)

Customer application testing

Set up :

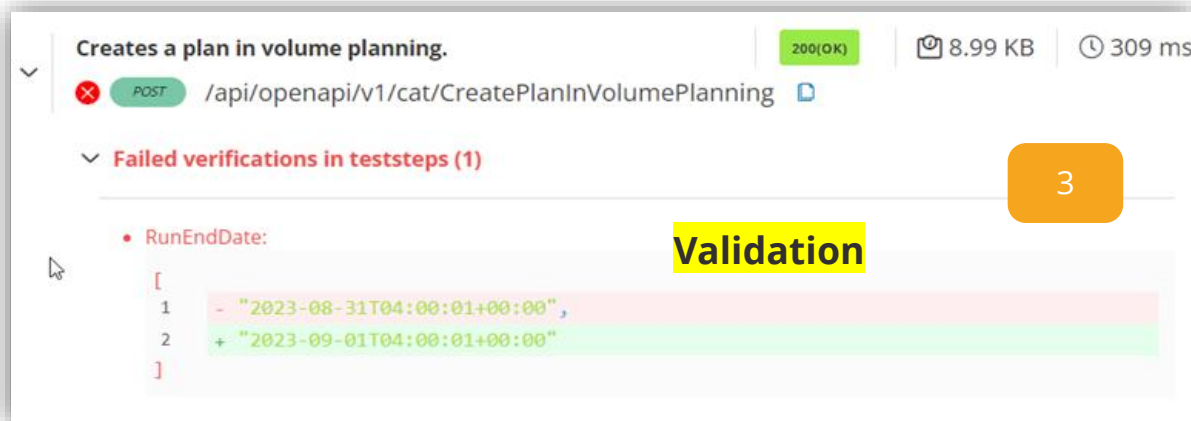
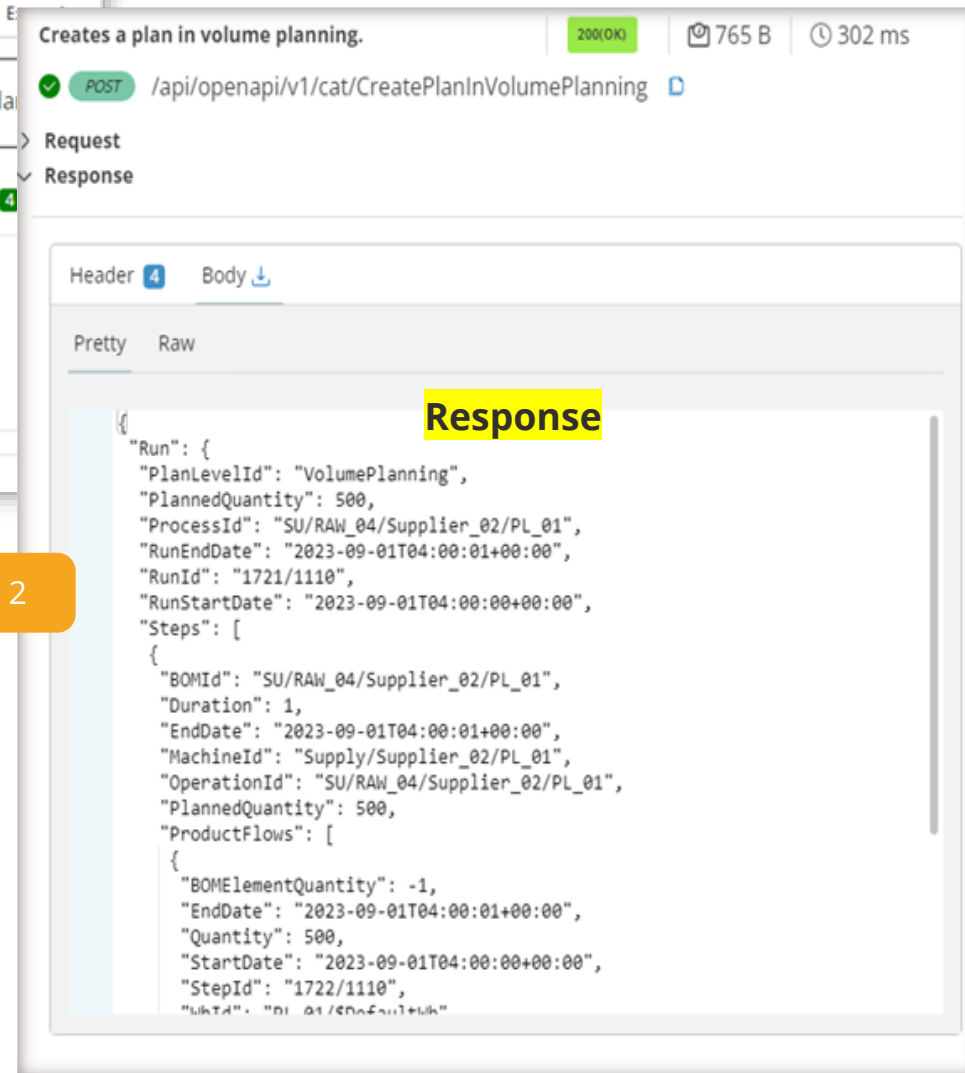
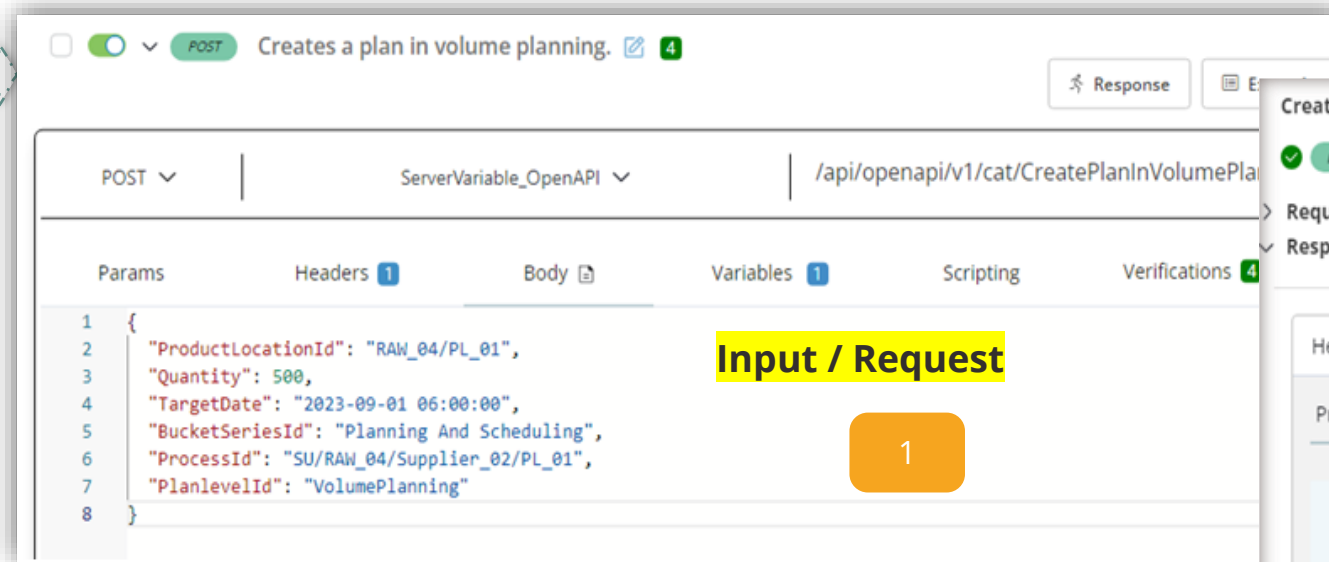
A solution for **regression testing** customer specific installation with every new build



End-to-End Tests (black box)

Customer application testing

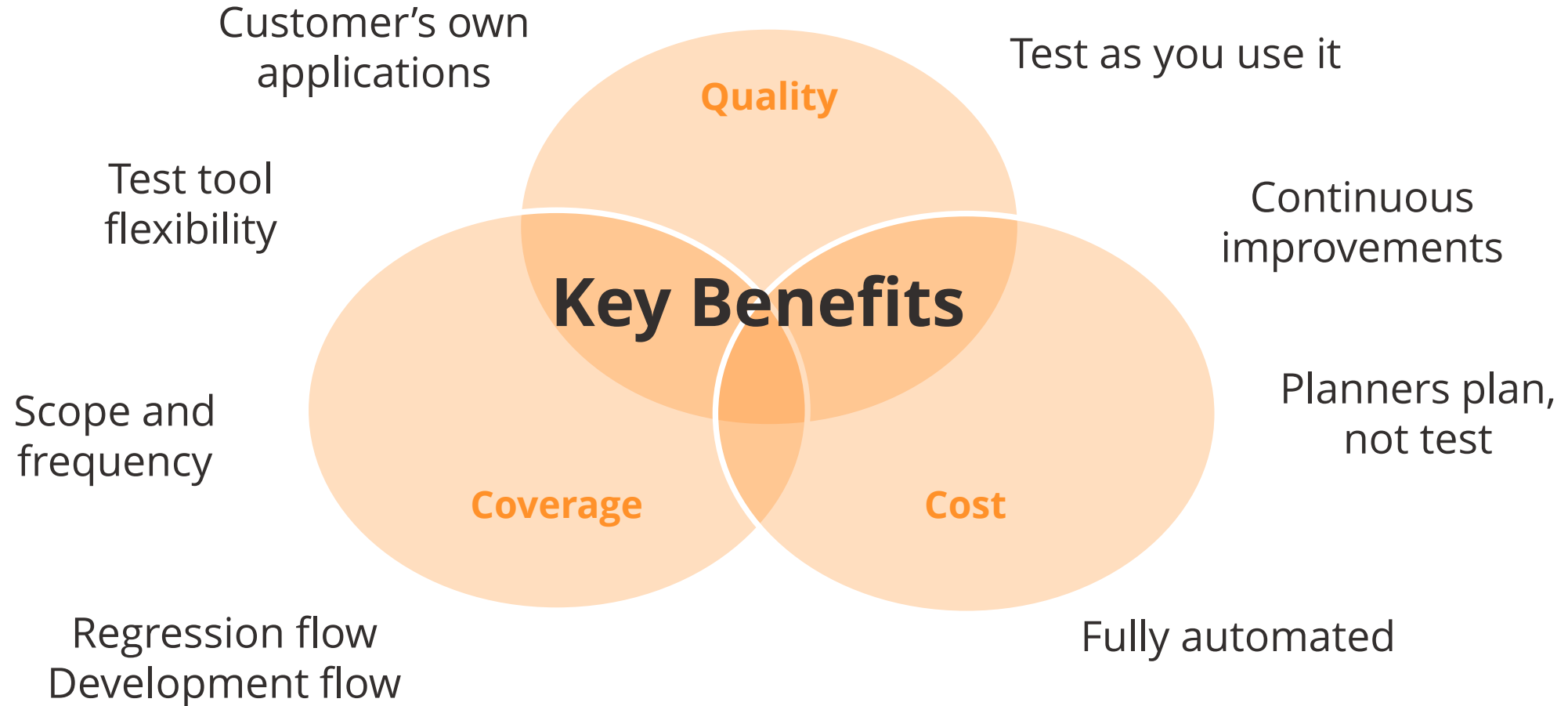
Test
Scenario



Any API Testing tool

End-to-End Tests (black box)

Customer application testing






Quality Gates



Quality Gates

Manual tests before merge / Separate automatic deploy for every Pull Request

 **Jenkins**

Dashboard

Compose

design-studio

Pull requests (52)

PR-1091

Up

Status

Changes

Build Now

View Configuration

Full Stage View

Bitbucket

Pipeline Syntax

Build History

trend

find

#14 Apr 28, 2021 1:07 PM CEST

#13 Apr 27, 2021 1:07 PM CEST

#12 Apr 26, 2021 1:07 PM CEST

#11 Apr 25, 2021 1:07 PM CEST

#10 Apr 24, 2021 1:07 PM CEST

#9 Apr 23, 2021 1:08 PM CEST

#8 Apr 22, 2021 1:07 PM CEST

#7 Apr 21, 2021 1:07 PM CEST

Pull Request PR-1091

Full project name: Compose/design-studio/PR-1091

Last Successful Artifacts

docker-log.txt2.34 KBview

id-designer.json205 Bview

wdio-0-0-junit-reporter.log18.04 KBview

Recent Changes

Stage View

Average stage times:
(Average full run time: ~47min
55s)

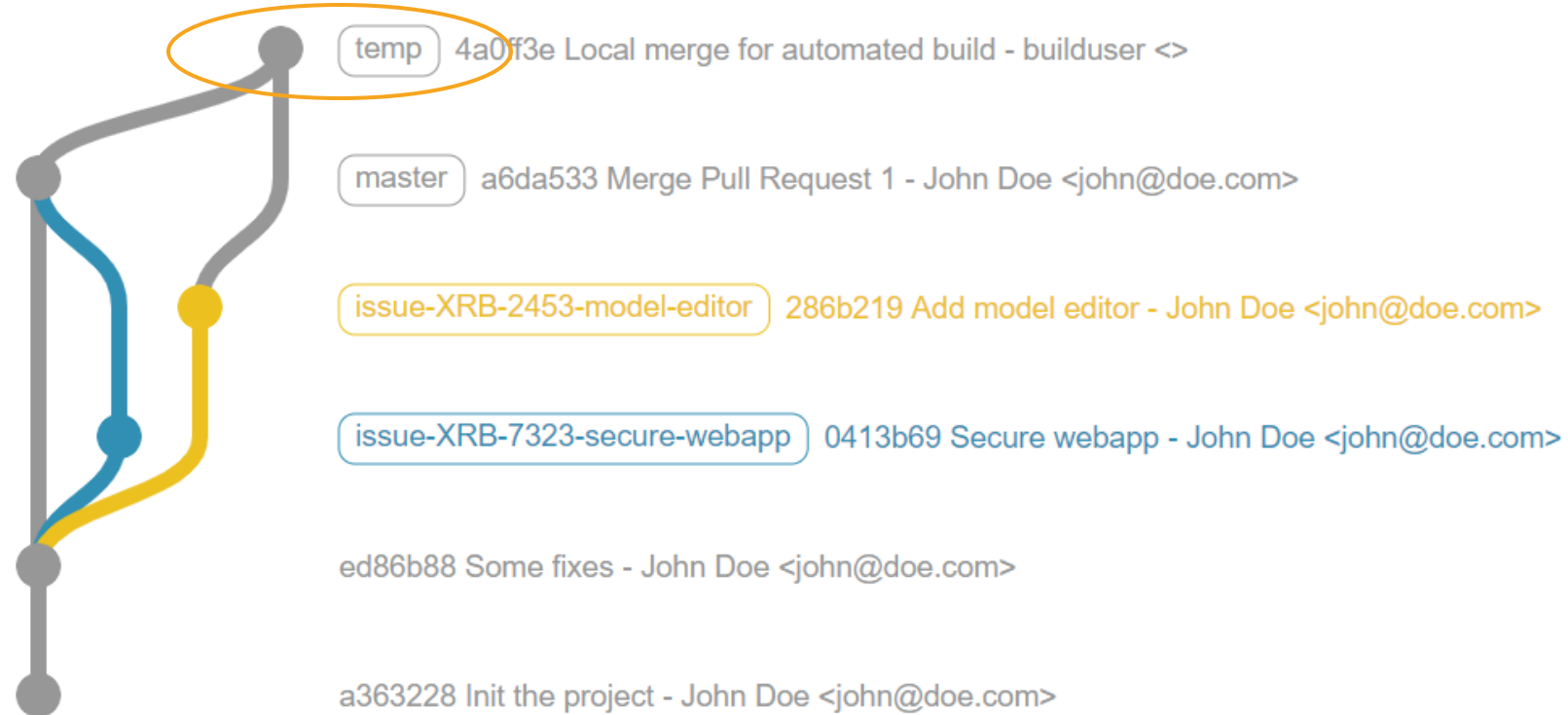
	Declarative: Checkout SCM	Stop running builds	Install	Build pr	Build master	Test	Integration	Deploy static	Deploy for QA	Selenium
#14 Apr 28 13:07 17 commits	25s	158ms	5min 47s (paused for 5ms)	6min 49s		26s	32s		8min 27s	7min 59s (paused for 4ms)
#13 Apr 27 13:07 18 commits	26s	195ms	5min 23s (paused for 3ms)	8min 37s		26s	32s		8min 10s	10min 0s (paused for 4ms)
#12 Apr 26 13:07 9 commits	27s	330ms	7min 4s (paused for 4ms)	8min 55s		15min 44s	37s		8min 25s	7min 12s (paused for 3ms)

© OMP Proprietary

Quality Gates

QA deploy integrates changes from main branch

- You know the new code will behave well after it is merged to master



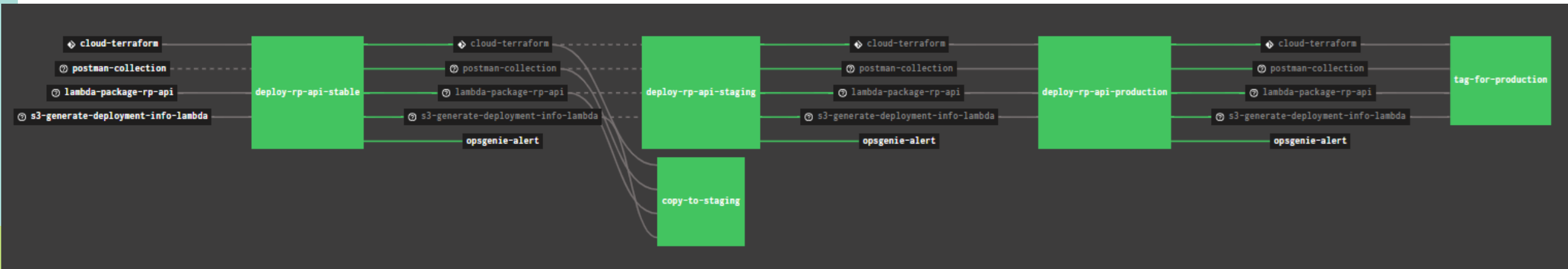
Quality Gates

Concourse / Continuous Deployment Tool

Deploys software in different environments (staging)

Don't deploy directly in production

Runs automated tests to assert that deploy was OK



4 Environments

Development: Contains bleeding edge code

- Used for developing features and experimenting

Stable: contains recently finished code

- Used for manual tests

Staging: mimics production to allow debugging without affecting production

Production: Live environment, used by customers



Quality of Tests

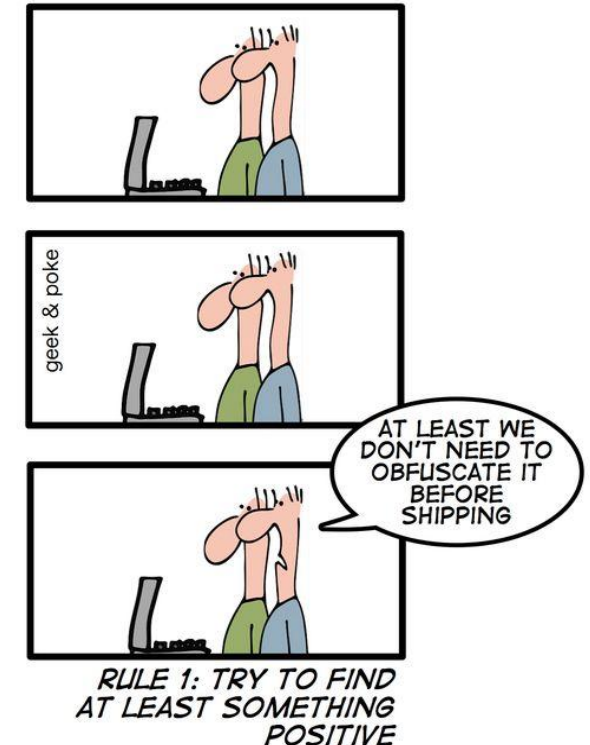


Quality of tests

Code reviews

- **Test code is also reviewed**
- **Good tests act as documentation of the software**
 - How is the unit supposed to behave?
 - Which functionalities / responsibilities does it have?
 - Tests should be readable
 - Test-name: <Function>_When<condition>_Should<expectation> (e.g., BookOrder_WhenOrderIdExists_ShouldRejectRequest)
 - Split code in blocks **arrange/act/assert**
- Test should be maintainable
 - DRY (don't repeat yourself), ...

HOW TO MAKE A GOOD CODE REVIEW



Quality of tests

Metrics for Customer Application Testing

- **Unit tests**
 - 34.112 lines of test-code (of a total 59.705 lines of code) = **57% of code is test code**
 - Line coverage: 88%
 - Branch coverage: 79%
 - 1523 tests
- **Contract tests (Web API tests)**
 - 40 (~ 50-100 API requests per test)
- **End-to-End / Journeys (Web API tests)**
 - 6 (~ 50-100 API requests per test)

Quality of tests

When there is a bug:

Analysis: why wasn't this caught by automatic tests?

Add test to prevent this bug from happening again

TDD works very well when
fixing bugs





Lessons learned



Profile of a good tester

- **“The best testers think like users... .”**
 - Get functional understanding of the supply chain context
 - Be creative (as the end user 😊), while being organized and rigorous as well
- **But they also need good technical skills**
 - Be able to cope with a complex system
 - Focus on automation
- **Interaction with other people (developers, functional, ICT...)**
 - Be aware on status/changes of development tasks
 - Discuss requirements

No one strategy fits all

You need Unit tests and Integration tests

- **Unit tests: run fast / higher coverage / easier to run locally**
- **Integration tests: Test wiring of all the components / are contracts aligned?/ does the software work?**

Choose the right test type for the job

- **Unit test / Integration test / End-to-end test**



No one strategy fits all



Find balance between code coverage and cost

- **Depends on industry and the part of the software !**
- **Generally, full coverage doesn't pay off: gain of caught bugs is smaller than cost of adding/maintaining tests**



1. Run tests on Pull Request build

- Test results are validated before code is merged into main

2. Do manual tests on Pull Request deploys

- Manual check before code is merged into main

3. Write code that is easy to unit test

- Dependency Inversion Principle
- Functional programming

4. TDD

- Helps while fixing bug
- You already have an automatic regression test for the future

main build green (most of the time)
main should always be deployable

