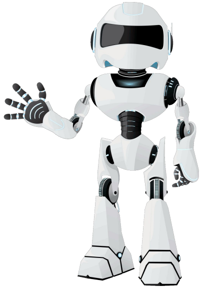


Testing Autonomous Mobile Robots

Warehouse Automation Systems

Mesut Durukal

in mesutdurukal

▶ mesutdurukal

X DurukalMesut

🌐 mesutdurukal.com

AGENDA

01

System Under Test

- Description
- Components

02

Test Activities

- Test Environment
- Automation

03

Challenges

- Problems
- Solutions

04

Summary

Copyright © Moyan Brenn - Flickr.com

Warehouse Automation

Autonomous Mobile Robots

Testing & Automation

System Under Test

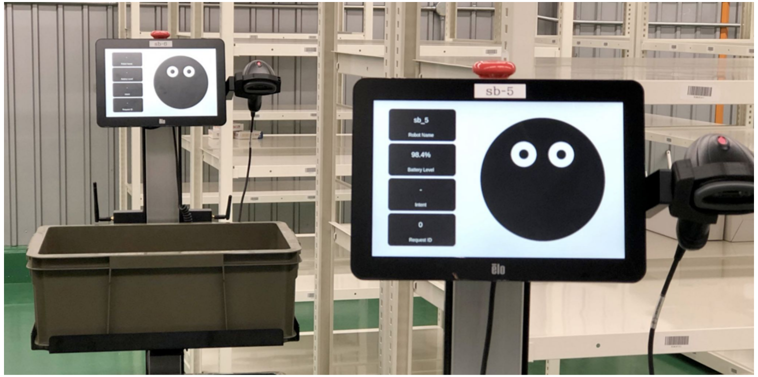
Warehouse Automation

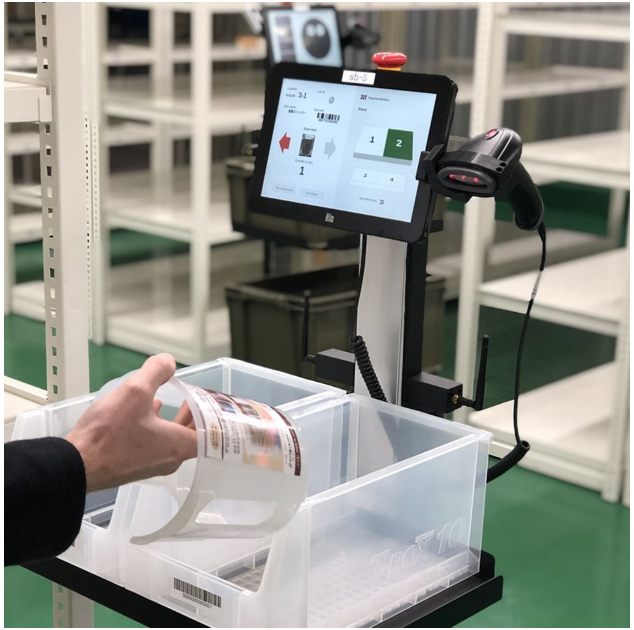
Components

AMR (Autonomous Mobile Robots)

Components

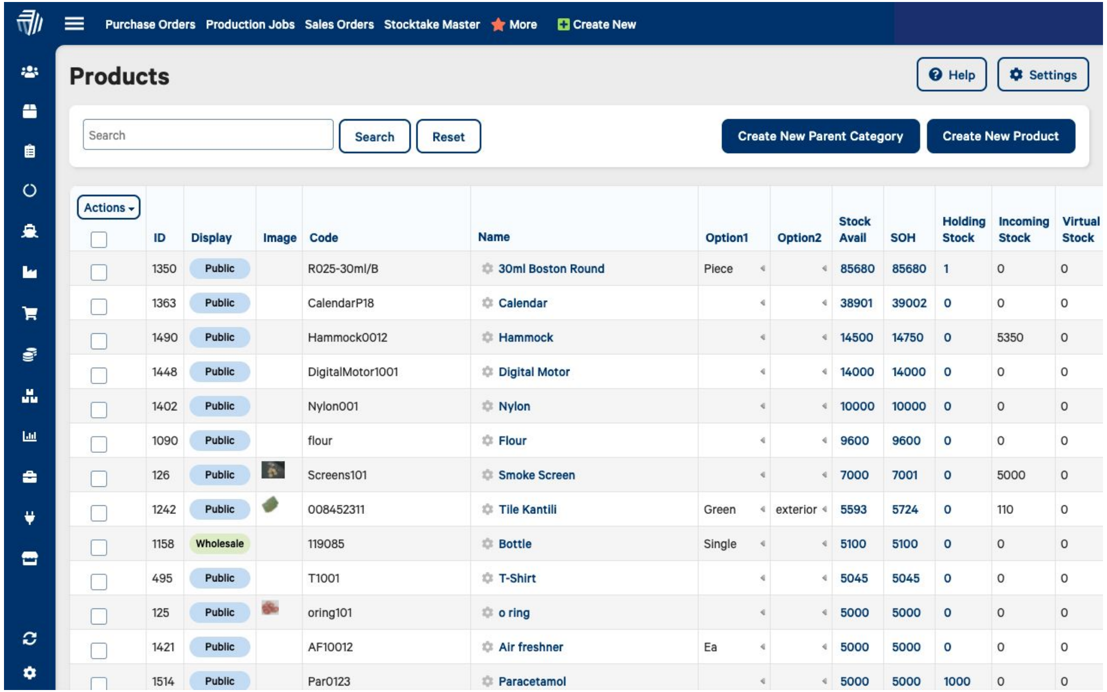
Robot UI





Components

IMS (Inventory Management System)

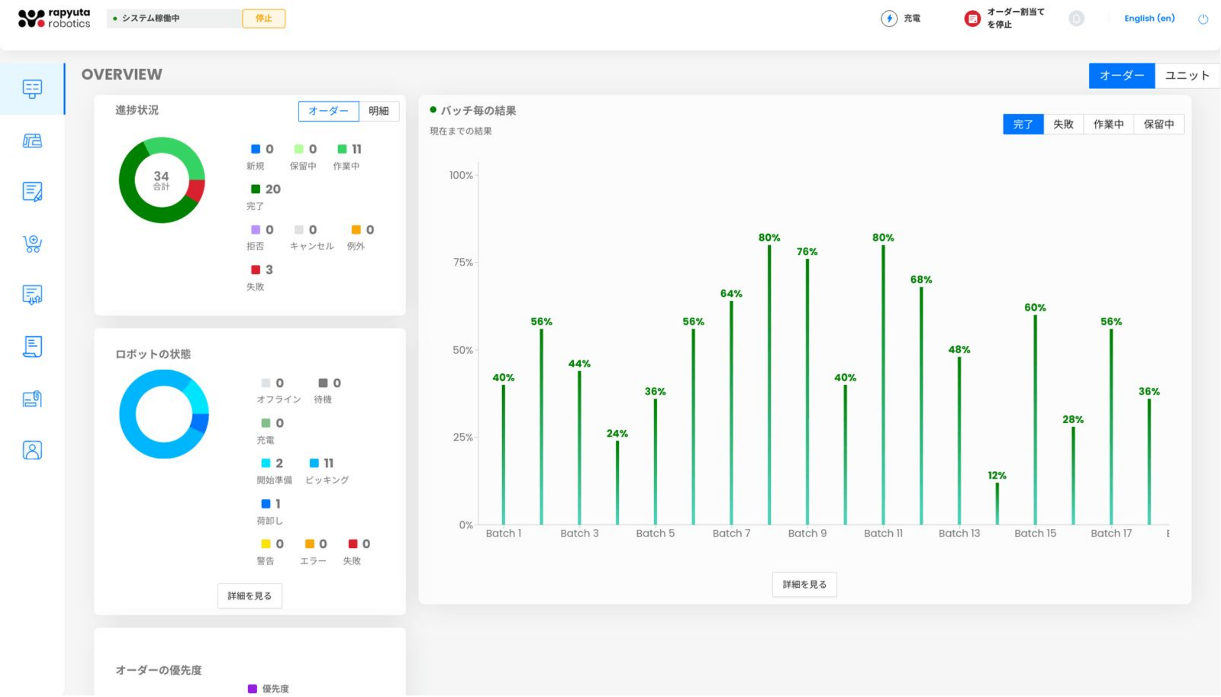


Products

□	ID	Display	Image	Code	Name	Option1	Option2	Stock Avail	SOH	Holding Stock	Incoming Stock	Virtual Stock
□	1350	Public		R025-30ml/B	30ml Boston Round	Piece		85680	85680	1	0	0
□	1363	Public		CalendarP18	Calendar			38901	39002	0	0	0
□	1490	Public		Hammock0012	Hammock			14500	14750	0	5350	0
□	1448	Public		DigitalMotor1001	Digital Motor			14000	14000	0	0	0
□	1402	Public		Nylon001	Nylon			10000	10000	0	0	0
□	1090	Public		flour	Flour			9600	9600	0	0	0
□	126	Public		Screens101	Smoke Screen			7000	7001	0	5000	0
□	1242	Public		008452311	Tile Kantili	Green	exterior	5593	5724	0	110	0
□	1158	Wholesale		119085	Bottle	Single		5100	5100	0	0	0
□	495	Public		T1001	T-Shirt			5045	5045	0	0	0
□	125	Public		oring101	o ring			5000	5000	0	0	0
□	1421	Public		AF10012	Air freshner	Ea		5000	5000	0	0	0
□	1514	Public		Par0123	Paracetamol			5000	5000	1000	0	0

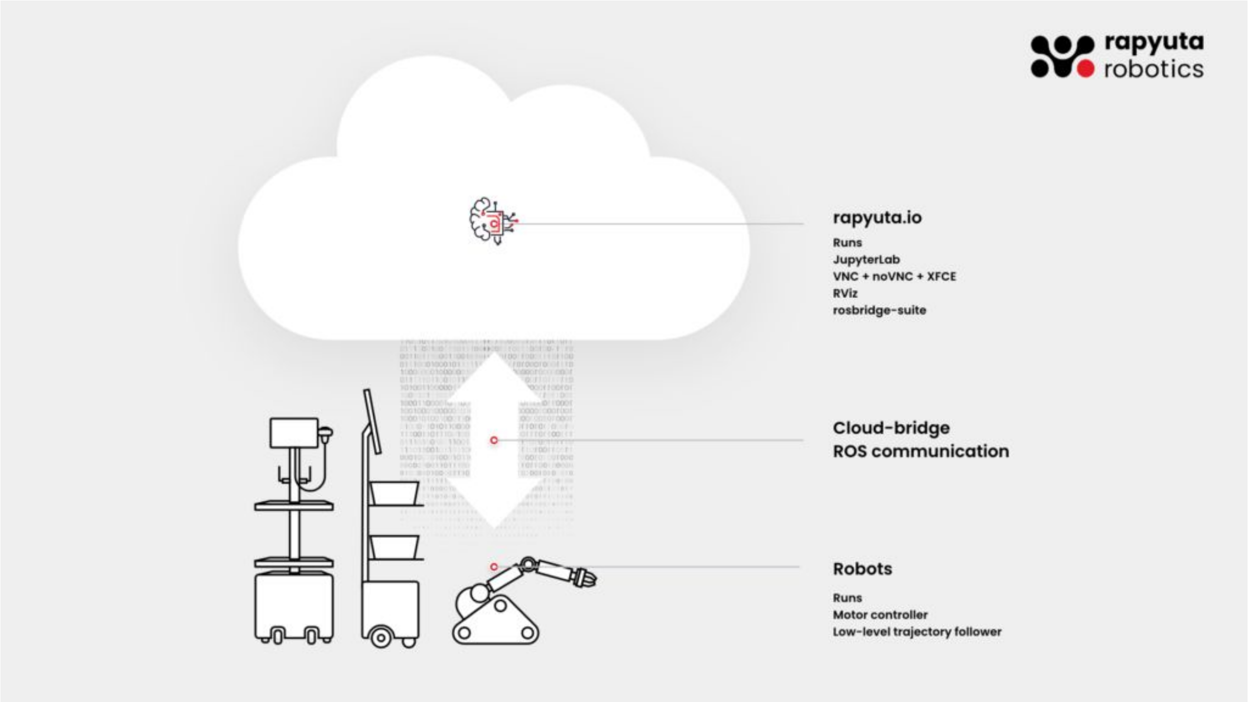
Components

EDGE PC



Components

Cloud Platform



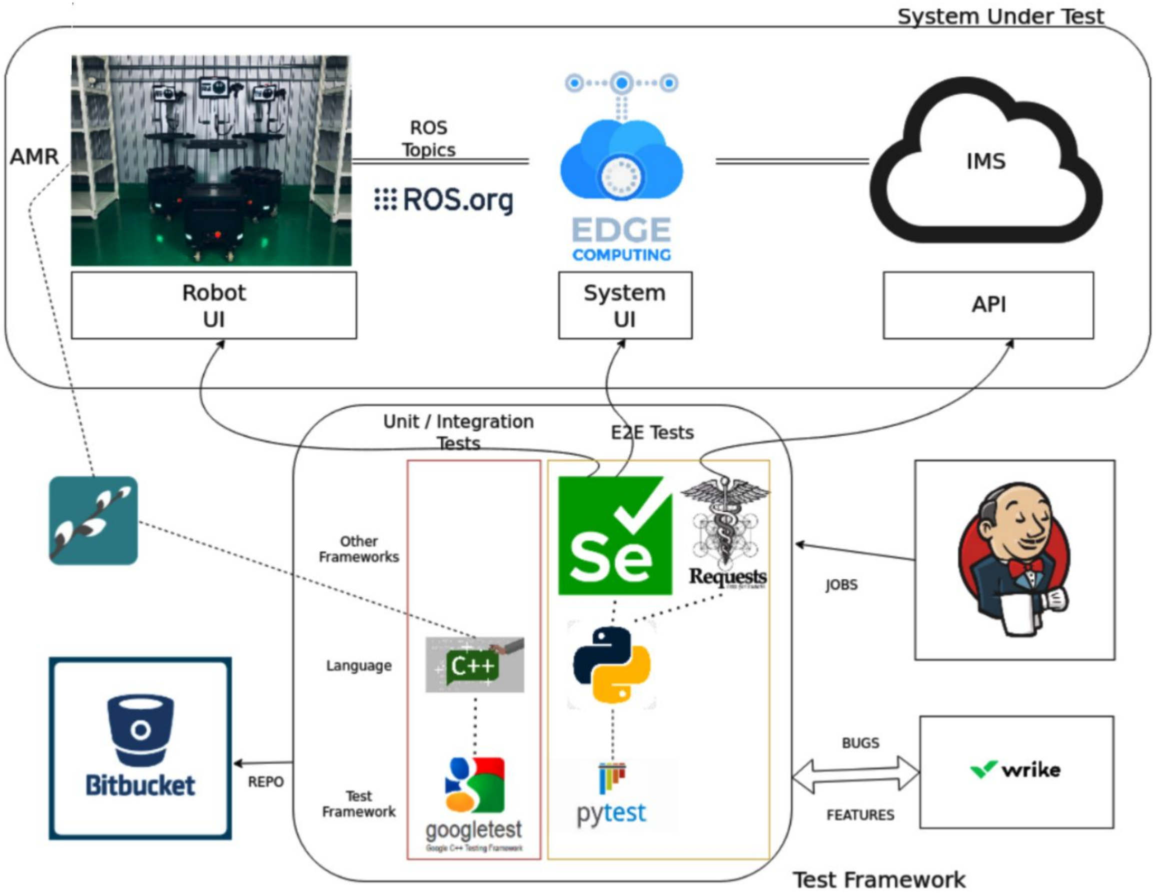
rapyuta.io

Runs
JupyterLab
VNC + noVNC + XFCE
RViz
rosbridge-suite

Cloud-bridge
ROS communication

Robots

Runs
Motor controller
Low-level trajectory follower

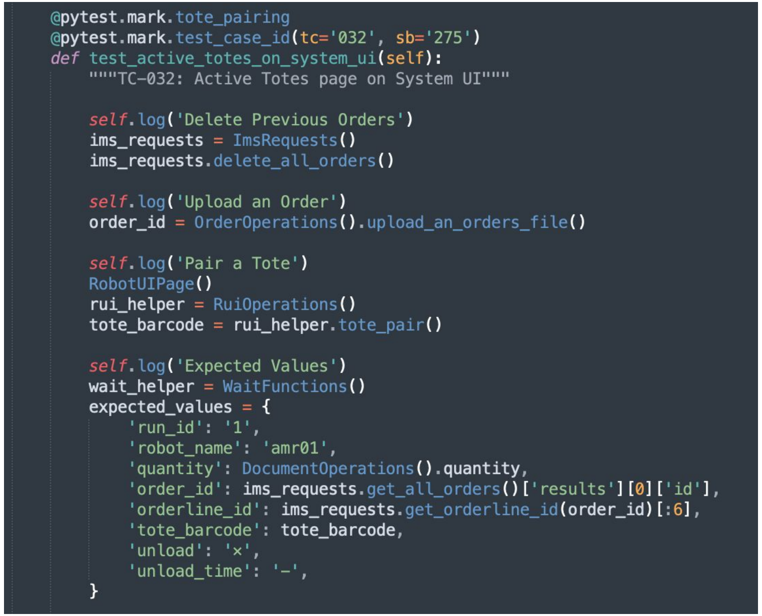

System Under Test
AMR
ROS
Topics
ROS.org
EDGE
COMPUTING
IMS
Robot
UI
System
UI
API
Unit / Integration
Tests
E2E Tests
Other
Frameworks
Language
Test
Framework
C++
googletest
Google C++ Testing Framework
Se
Requests
pytest
JOBS
BUGS
FEATURES
Bitbucket
REPO
wrike
Test Framework

Testing

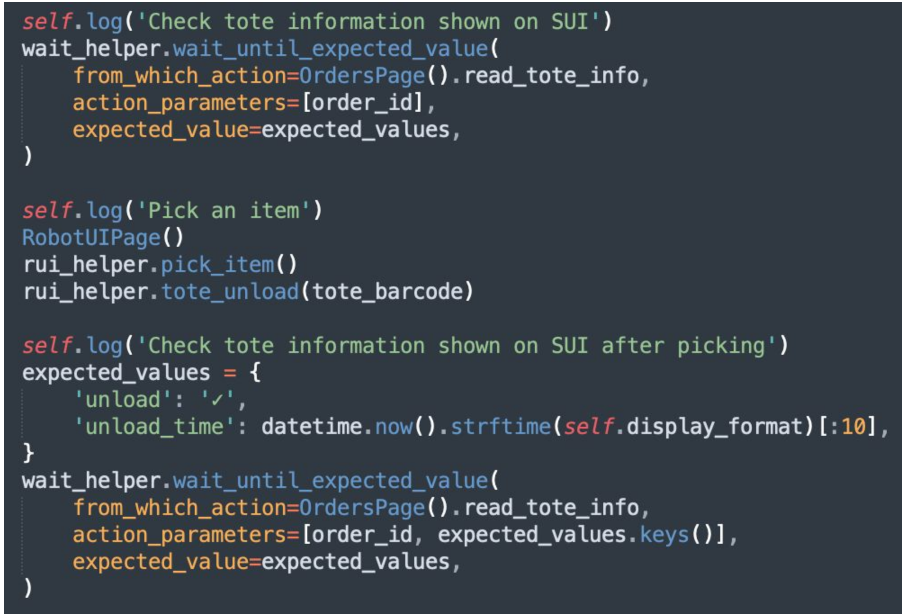
Alpha Testing

Testing

An example Scenario



```python
@pytest.mark.tote_pairing
@pytest.mark.test_case_id(tc='032', sb='275')
def test_active_totes_on_system_ui(self):
    """TC-032: Active Totes page on System UI"""

    self.log('Delete Previous Orders')
    ims_requests = ImsRequests()
    ims_requests.delete_all_orders()

    self.log('Upload an Order')
    order_id = OrderOperations().upload_an_orders_file()

    self.log('Pair a Tote')
    RobotUIPage()
    rui_helper = RuiOperations()
    tote_barcode = rui_helper.tote_pair()

    self.log('Expected Values')
    wait_helper = WaitFunctions()
    expected_values = {
        'run_id': '1',
        'robot_name': 'amr01',
        'quantity': DocumentOperations().quantity,
        'order_id': ims_requests.get_all_orders()['results'][0]['id'],
        'orderline_id': ims_requests.get_orderline_id(order_id)[:6],
        'tote_barcode': tote_barcode,
        'unload': '×',
        'unload_time': '–',
    }
```

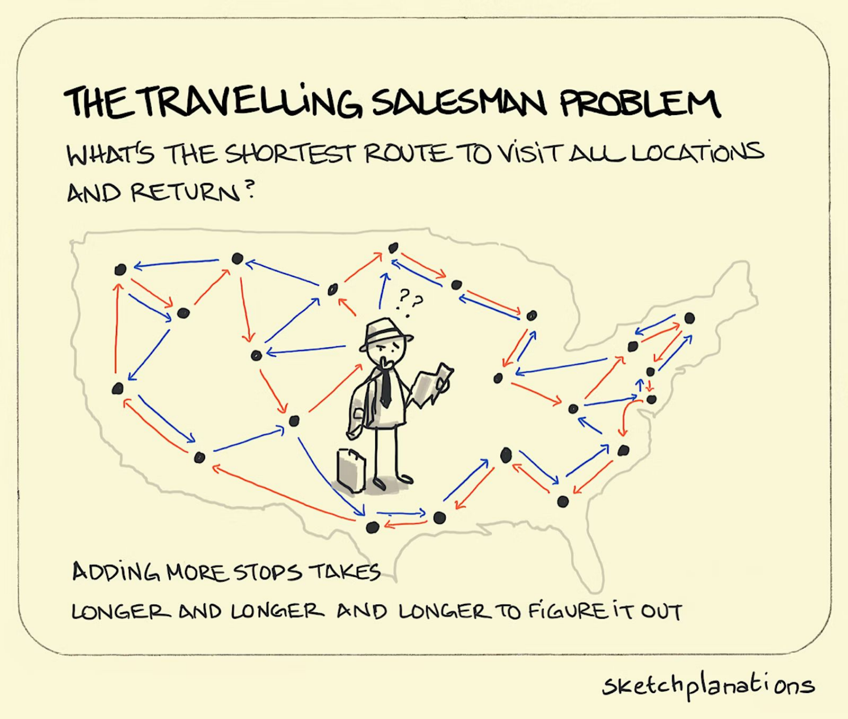


```python
self.log('Check tote information shown on SUI')
wait_helper.wait_until_expected_value(
    from_which_action=OrdersPage().read_tote_info,
    action_parameters=[order_id],
    expected_value=expected_values,
)

self.log('Pick an item')
RobotUIPage()
rui_helper.pick_item()
rui_helper.tote_unload(tote_barcode)

self.log('Check tote information shown on SUI after picking')
expected_values = {
    'unload': '✓',
    'unload_time': datetime.now().strftime(self.display_format)[:10],
}
wait_helper.wait_until_expected_value(
    from_which_action=OrdersPage().read_tote_info,
    action_parameters=[order_id, expected_values.keys()],
    expected_value=expected_values,
)
```

Problem 1
Testability

2.5x
Productivity improvement

25%
Reduced picking time


THE TRAVELLING SALESMAN PROBLEM
WHAT'S THE SHORTEST ROUTE TO VISIT ALL LOCATIONS AND RETURN?
??
ADDING MORE STOPS TAKES LONGER AND LONGER AND LONGER TO FIGURE IT OUT
sketchplanations

Performance Metrics

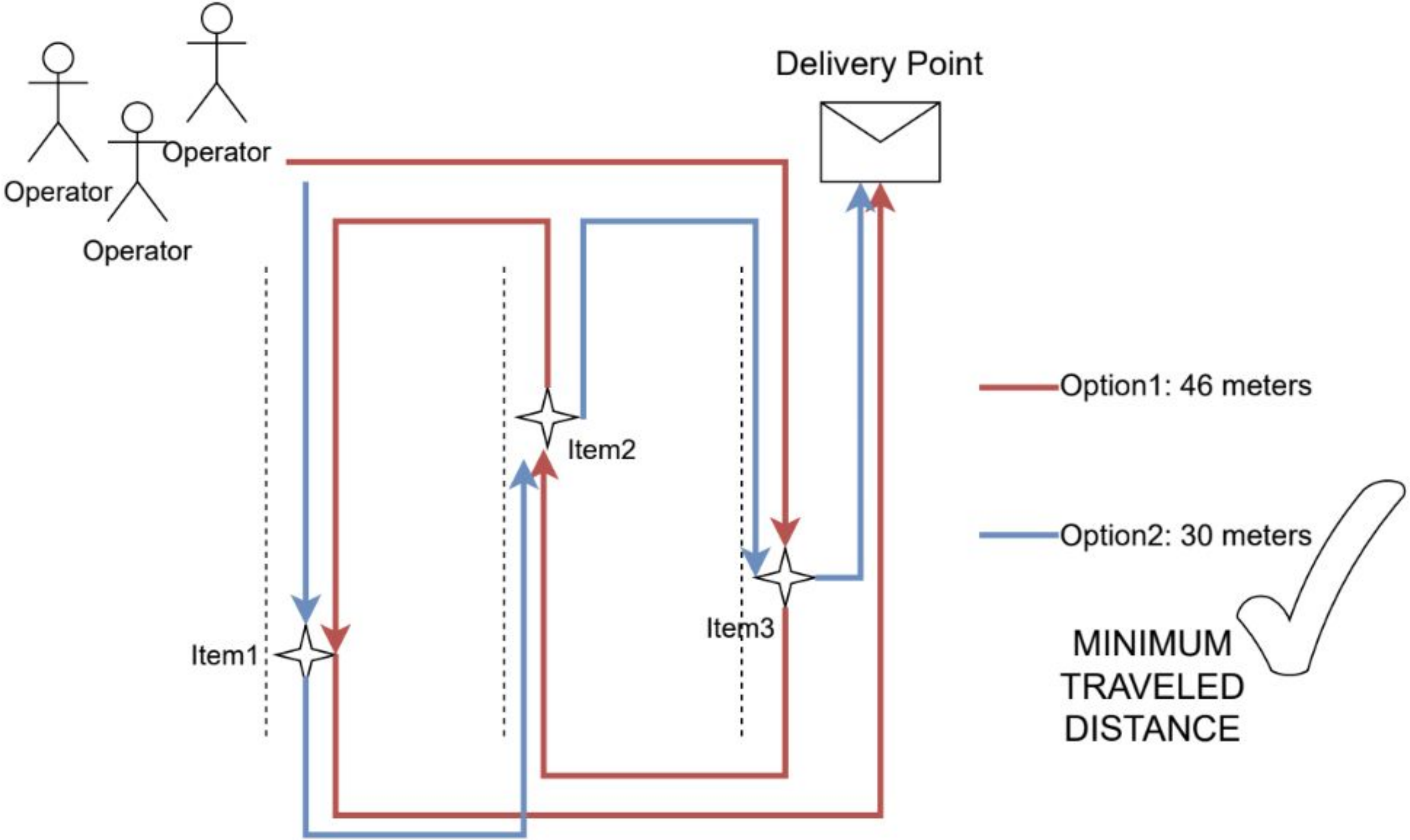
- Order Completion Time
- Walking Distance
- Human Operator Needed

Test if our product brings a benefit:

- Calculate metrics without robots
- Calculate metrics with robots

Benchmark to find the best performing configuration:

- Calculate metrics with V1
- Calculate metrics with V2


Operator
Operator
Operator
Delivery Point
Item1
Item2
Item3
Option1: 46 meters
Option2: 30 meters
MINIMUM
TRAVELED
DISTANCE

Robots in Action

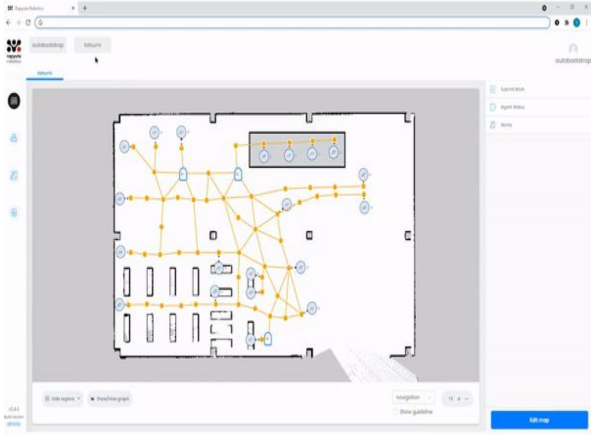
Birds Eye View

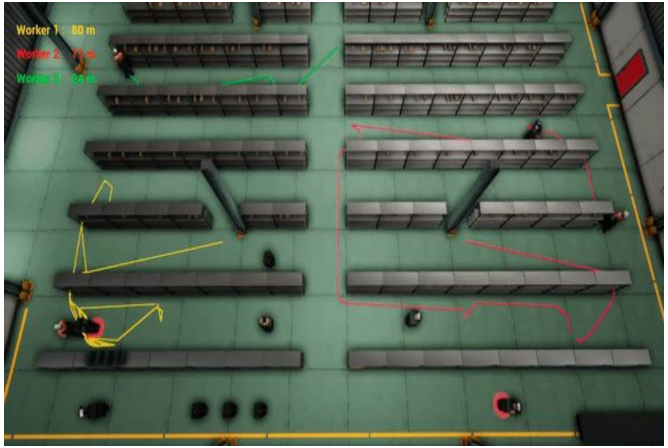
Problem 2

Hardware

Simulation

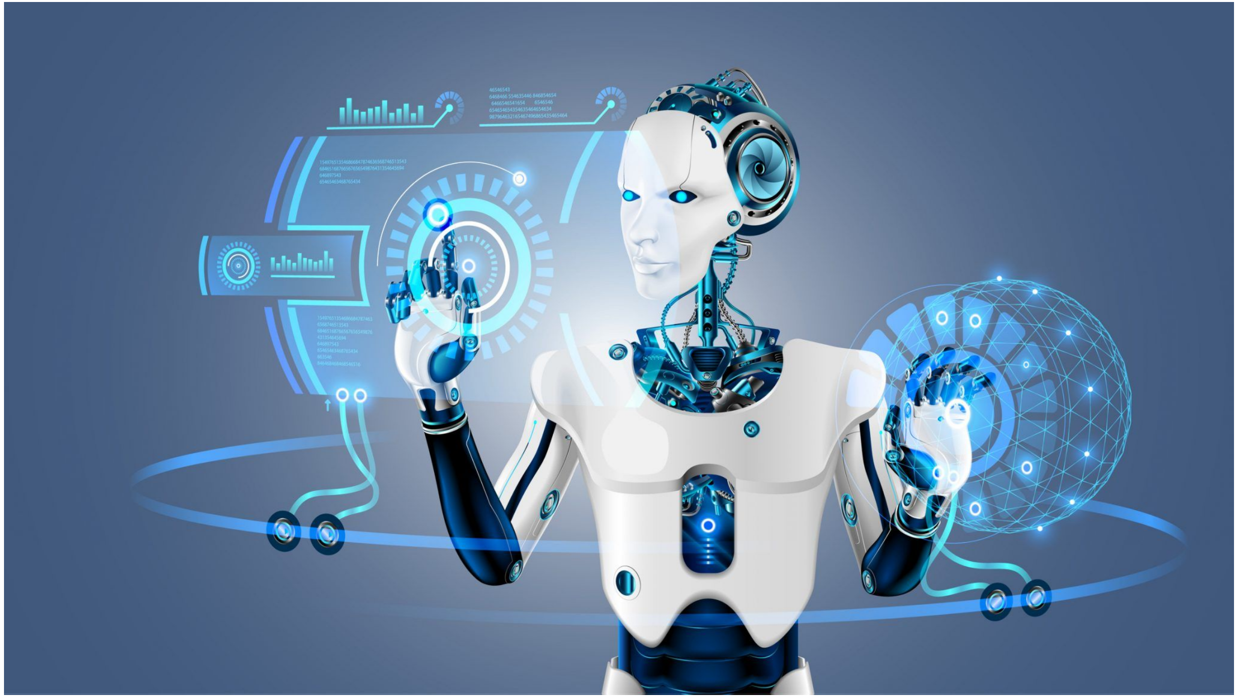
Robot Operating System





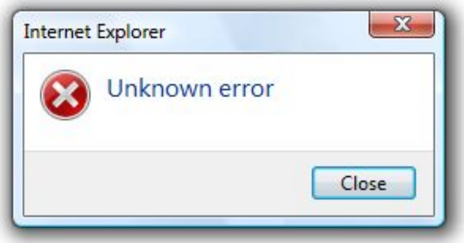
Problem 3

AI

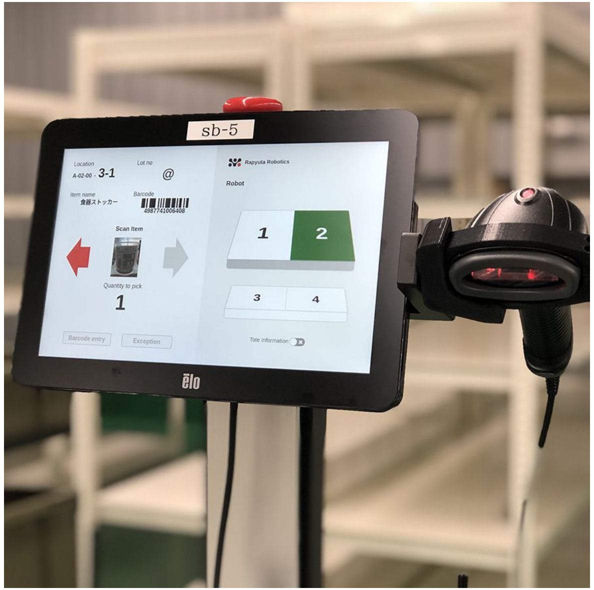


Problem 4

Usability

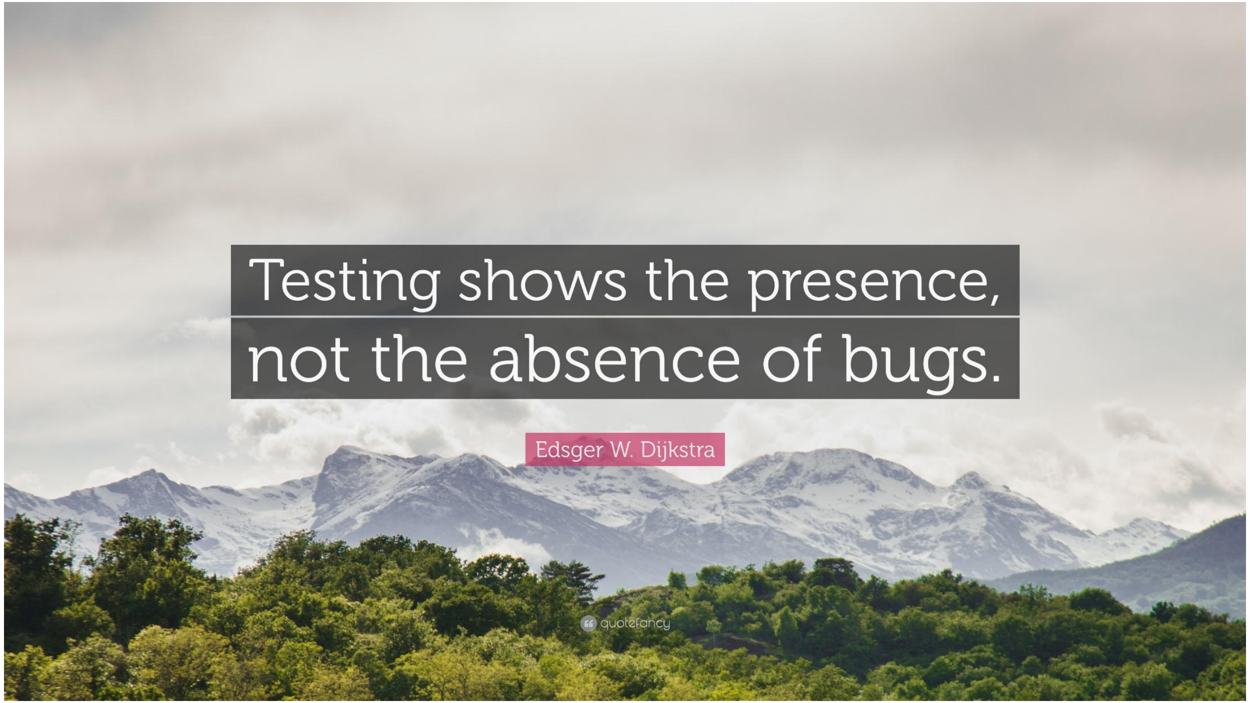

Internet Explorer

Unknown error



Problem 5

Recoverability



Testing shows the presence, not the absence of bugs.

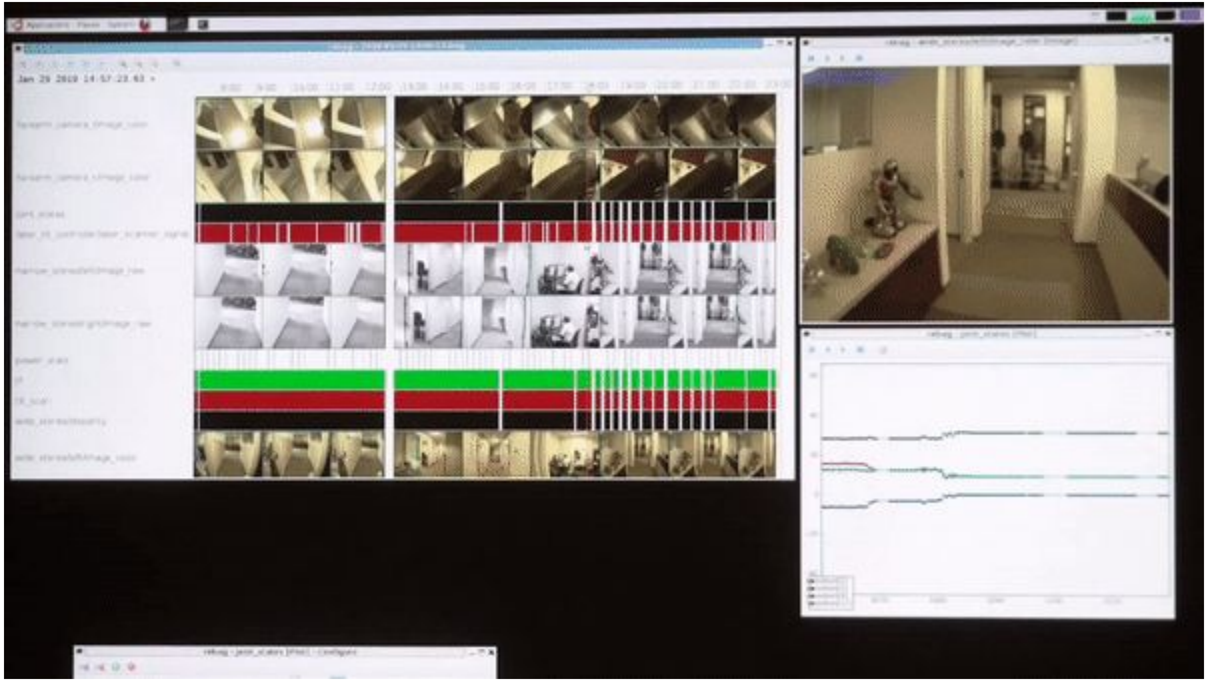
Edsger W. Dijkstra

Problem 6

Reproduction of Issues

Re-Play

ROS Bag



Problem 7

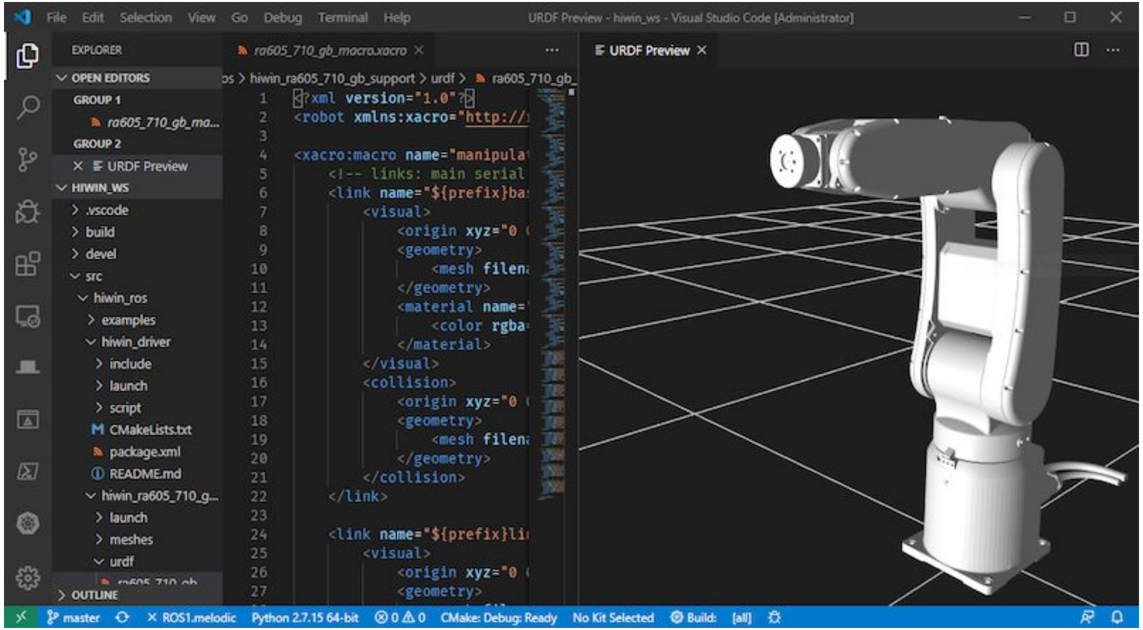
Customized Requirements

Problem 8

Domain Knowledge

Problem 9

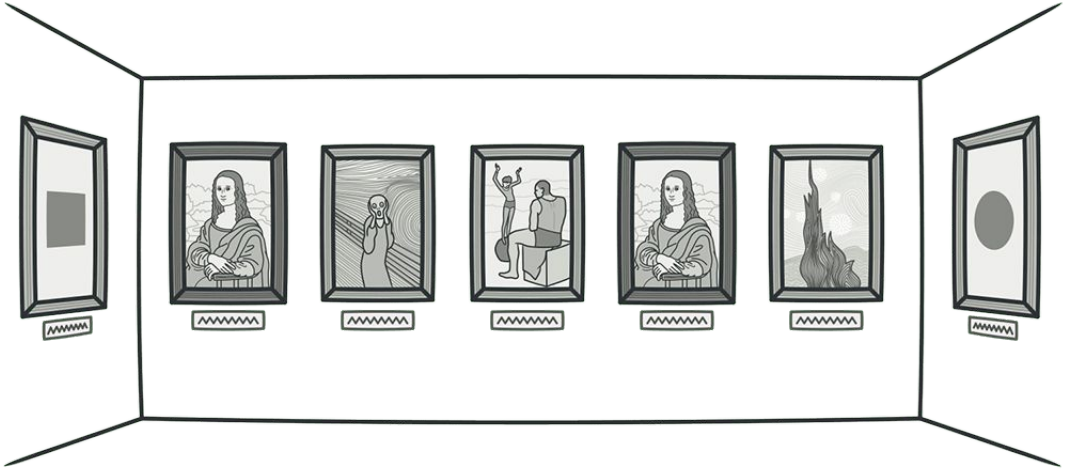
Technical Stack



```
<?xml version="1.0"?>
<robot xmlns:xacro="http://

<xacro:macro name="manipula
    <!-- links: main serial
    <link name="${prefix}ba
        <visual>
            <origin xyz="0 
            <geometry>
                <mesh filen
            </geometry>
            <material name=
                <color rgba
            </material>
        </visual>
        <collision>
            <origin xyz="0 
            <geometry>
                <mesh filen
            </geometry>
        </collision>
    </link>

    <link name="${prefix}li
        <visual>
            <origin xyz="0 
            <geometry>
```

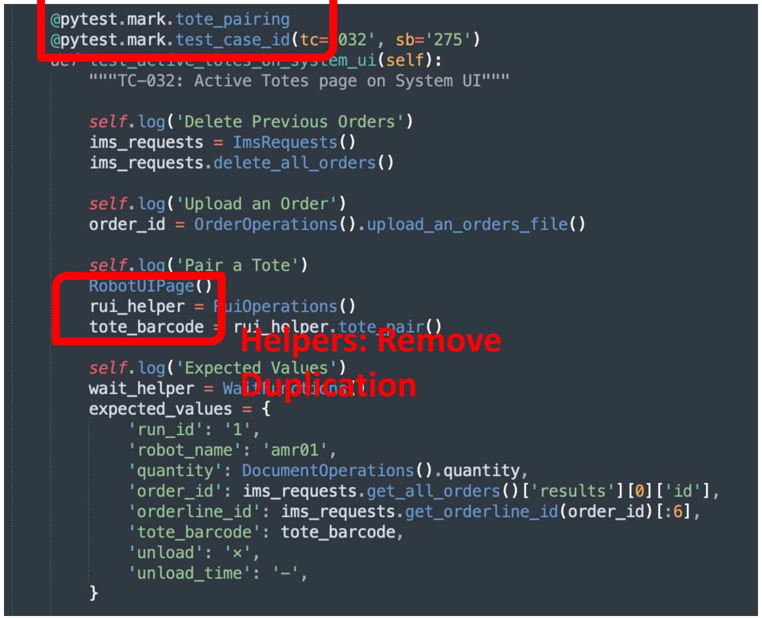
Problem 10

Maintenance



Testing

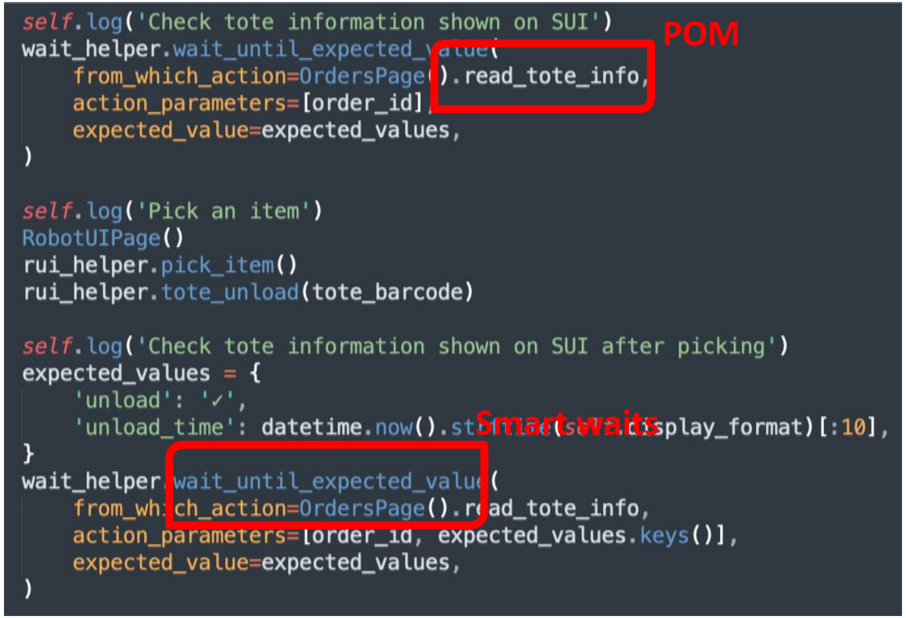
Revisit: Example Scenario



Tags

Helpers: Remove Duplication

```python
@pytest.mark.tote_pairing
@pytest.mark.test_case_id(tc='032', sb='275')
def test_active_totes_on_system_ui(self):
    """TC-032: Active Totes page on System UI"""

    self.log('Delete Previous Orders')
    ims_requests = ImsRequests()
    ims_requests.delete_all_orders()

    self.log('Upload an Order')
    order_id = OrderOperations().upload_an_orders_file()

    self.log('Pair a Tote')
    RobotUIPage()
    rui_helper = RuiOperations()
    tote_barcode = rui_helper.tote_pair()

    self.log('Expected Values')
    wait_helper = WaitFunctions()
    expected_values = {
        'run_id': '1',
        'robot_name': 'amr01',
        'quantity': DocumentOperations().quantity,
        'order_id': ims_requests.get_all_orders()['results'][0]['id'],
        'orderline_id': ims_requests.get_orderline_id(order_id)[:6],
        'tote_barcode': tote_barcode,
        'unload': '×',
        'unload_time': '-',
    }
```



POM

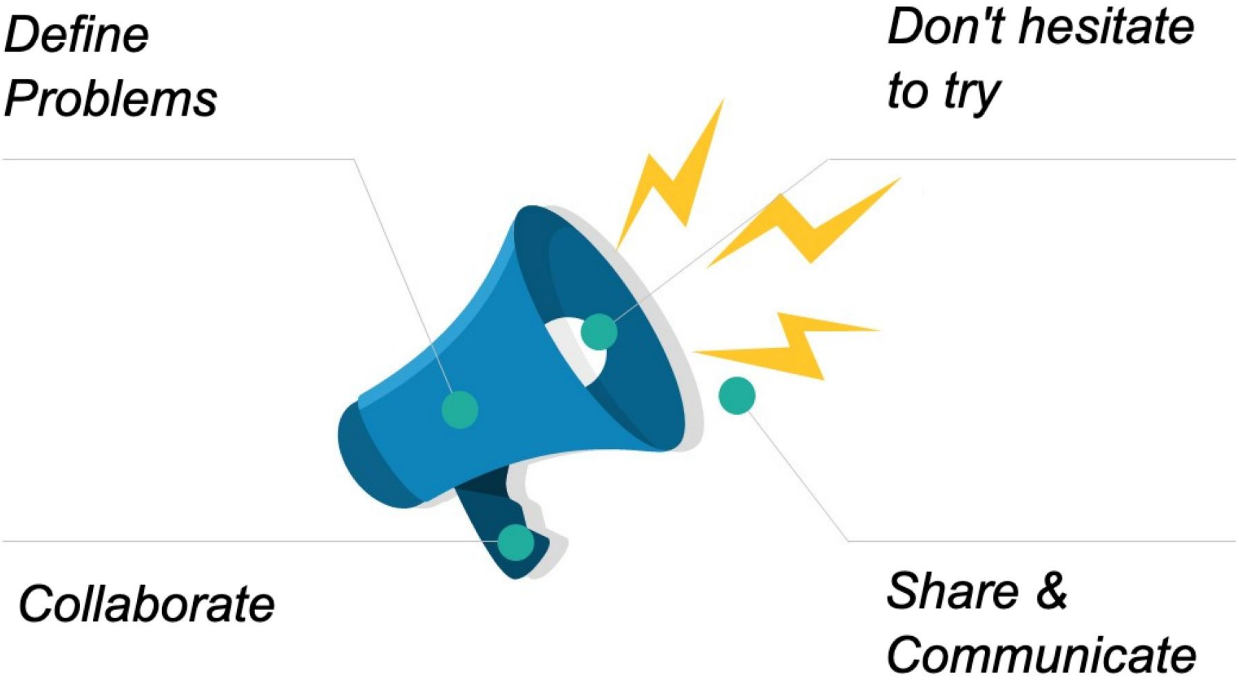
Smart waits

```python
self.log('Check tote information shown on SUI')
wait_helper.wait_until_expected_value(
    from_which_action=OrdersPage().read_tote_info,
    action_parameters=[order_id],
    expected_value=expected_values,
)

self.log('Pick an item')
RobotUIPage()
rui_helper.pick_item()
rui_helper.tote_unload(tote_barcode)

self.log('Check tote information shown on SUI after picking')
expected_values = {
    'unload': '✓',
    'unload_time': datetime.now().strftime(self.display_format)[:10],
}
wait_helper.wait_until_expected_value(
    from_which_action=OrdersPage().read_tote_info,
    action_parameters=[order_id, expected_values.keys()],
    expected_value=expected_values,
)
```

Summary

- Testability
- Hardware
- AI
- Usability
- Recoverability
- Reproduction of bugs
- Customization
- Domain
- Tech Stack
- Maintenance

- Performance Tests
- Simulation
- Experiments
- User Surveys
- Chaos Testing
- Rosbag
- Reusable tests
- Warehouse visits
- Onboarding docs
- Code Quality



Define Problems

Don't hesitate to try

Collaborate

Share & Communicate