



**HUSTEF**  
HUNGARIAN SOFTWARE TESTING FORUM

# Testing Autonomous Mobile Robots Warehouse Automation Systems

---

Mesut Durukal



mesutdurukal



@DurukalMesut



mesutdurukal.com



# AGENDA

01

## **System Under Test**

- Description
- Components

02

## **Test Activities**

- Test Environment
- Automation

03

## **Challenges**

- Problems
- Solutions

04

## **Summary**







***Warehouse Automation***

***Autonomous Mobile Robots***

***Testing & Automation***

# System Under Test

## Warehouse Automation



# Components

## AMR (Autonomous Mobile Robots)





# Components

## Robot UI



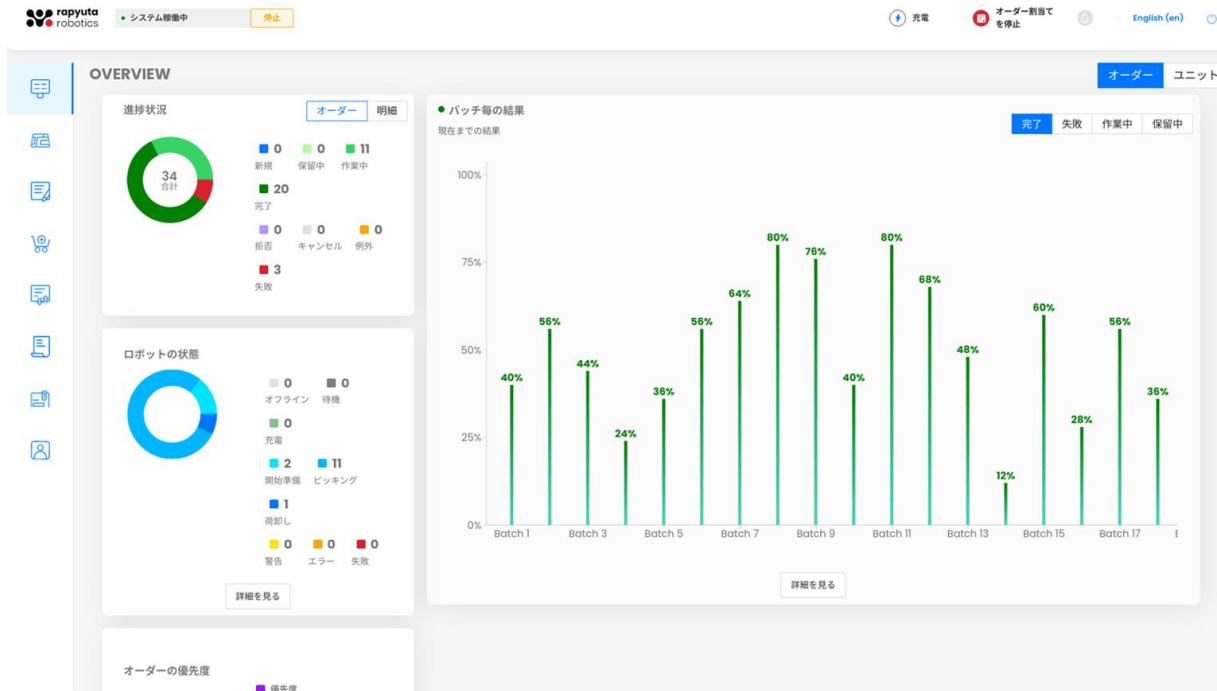


# IMS (Inventory Management System)

</

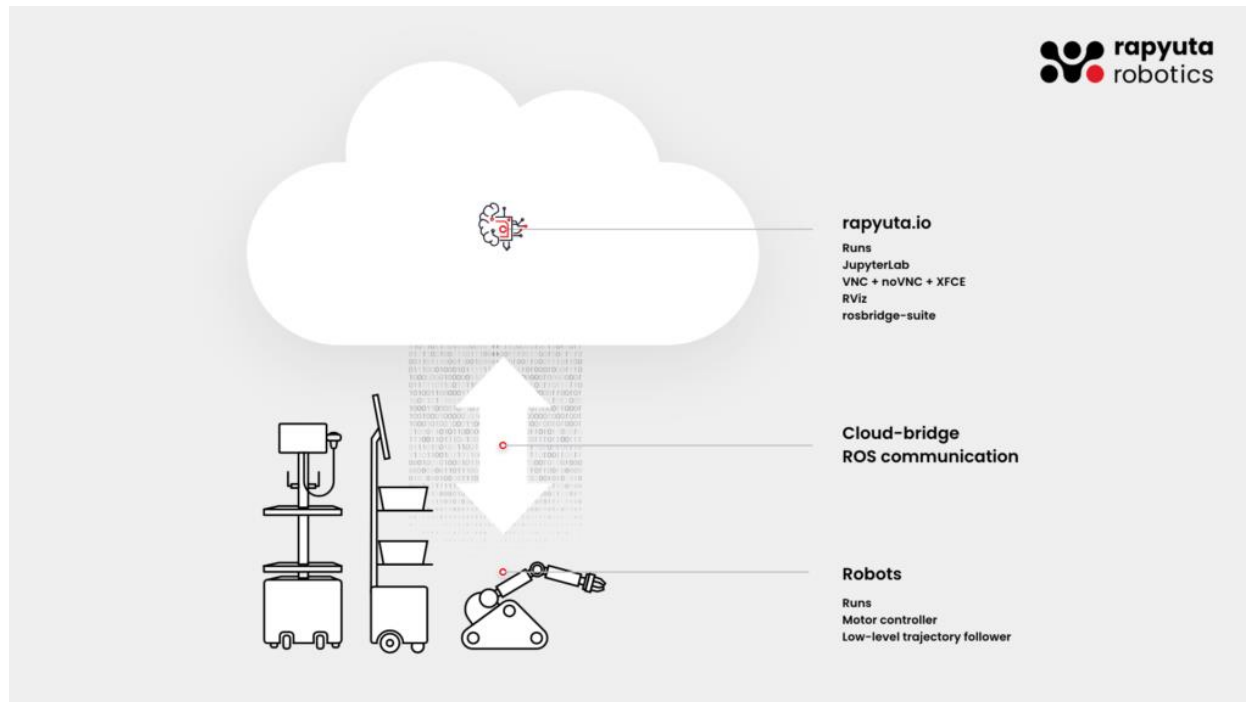
# Components

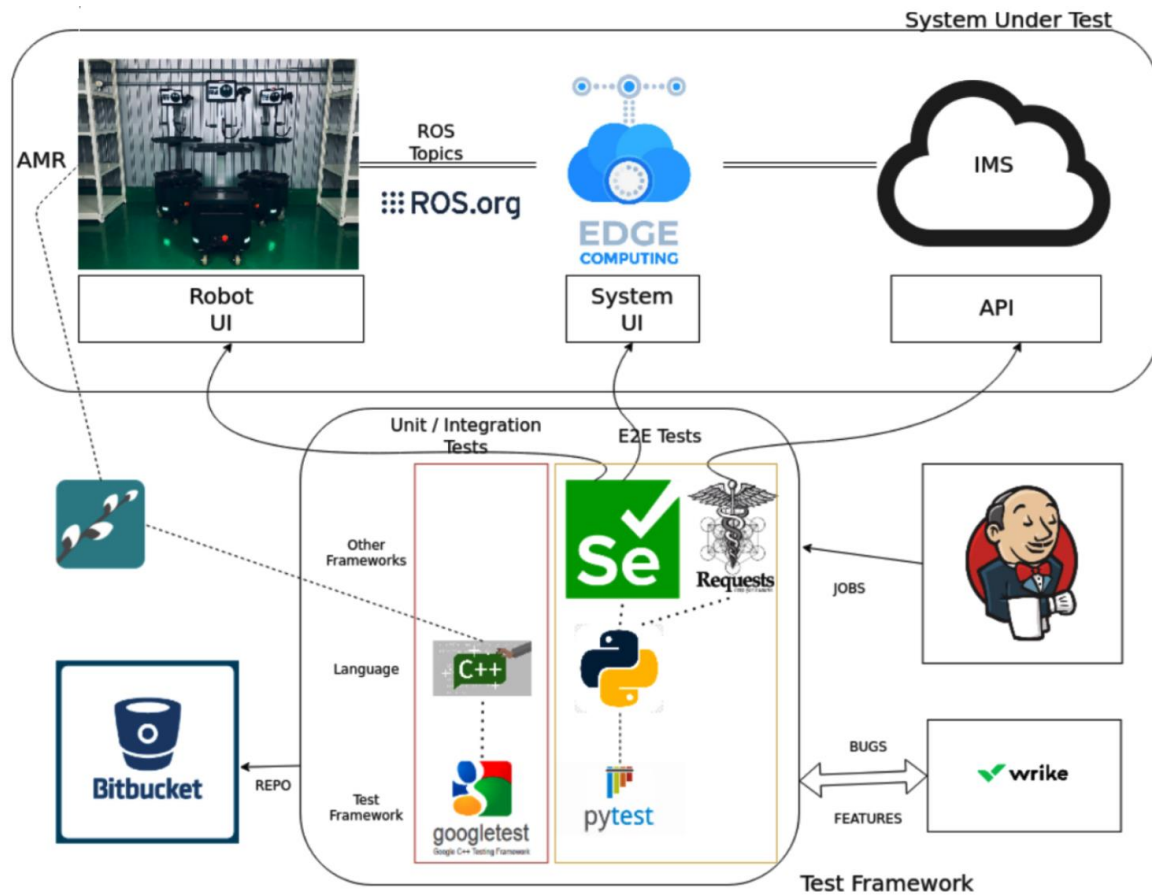
## EDGE PC



# Components

## Cloud Platform







# Testing

## Alpha Testing



# Testing

## An example Scenario

```
@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': 'x',
        'unload_time': '-',
    }
}
```

```
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



### 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

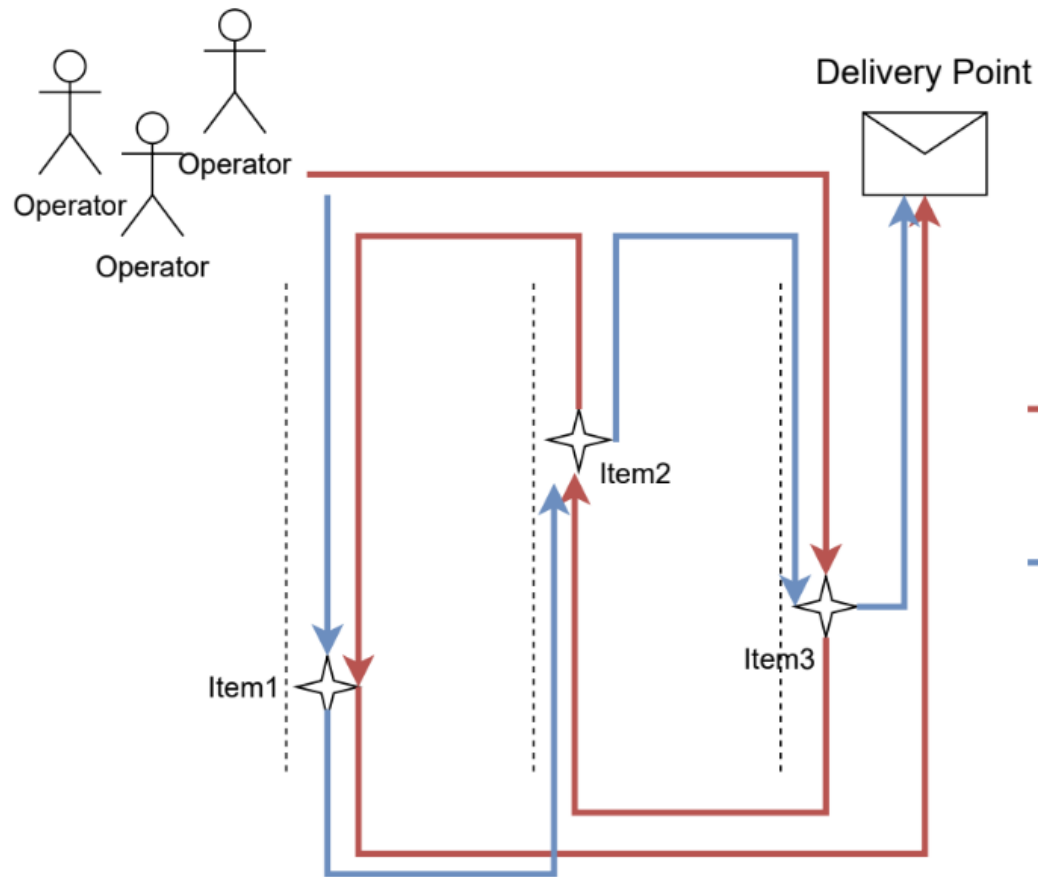
- Order Completion Time
- Walking Distance
- Human Operator Needed

#### Test if our product brings a benefit:

- Calculate metrics without robots
- Calculate metrics with robots

#### Benchmark to find the best performing configuration:

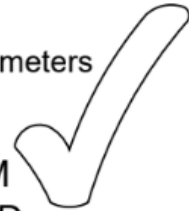
- Calculate metrics with V1
- Calculate metrics with V2



Option1: 46 meters

Option2: 30 meters

MINIMUM  
TRAVELED  
DISTANCE





# Robots in Action

## Birds Eye View



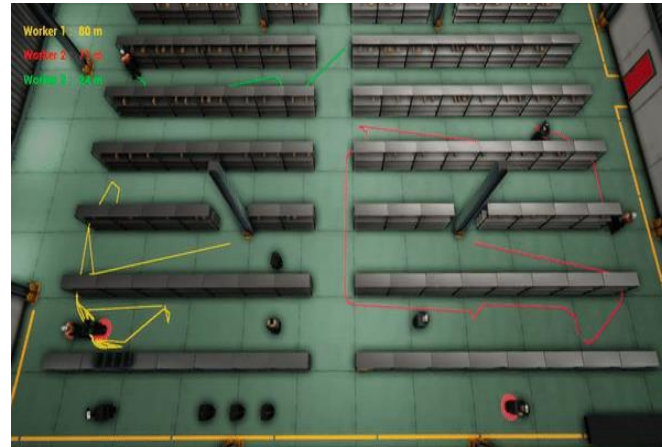
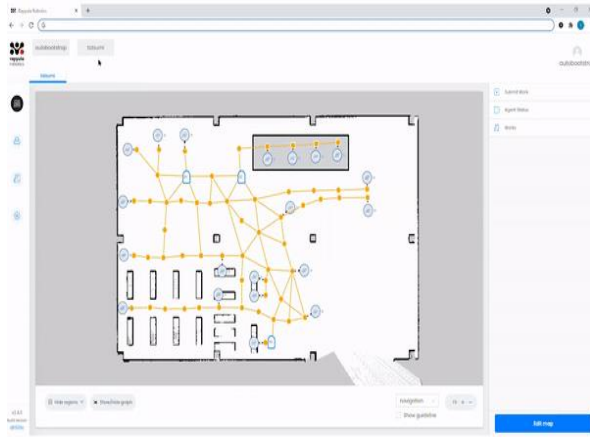
# Problem 2

## Hardware



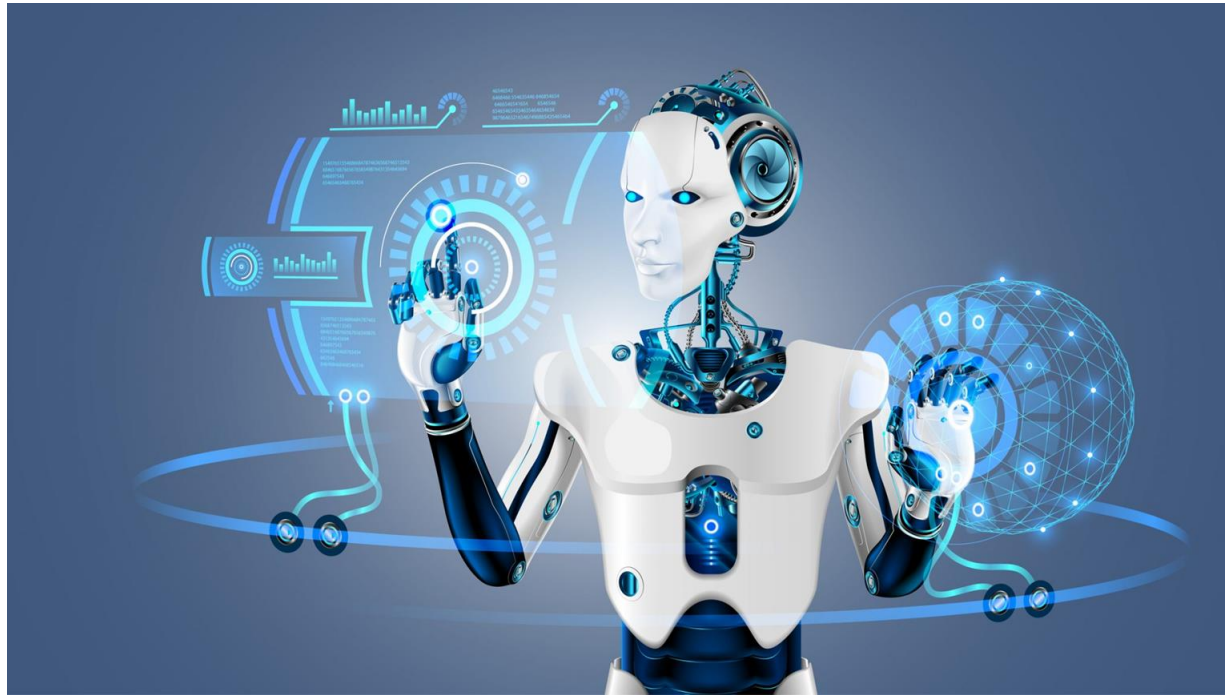
# Simulation

## Robot Operating System



# Problem 3

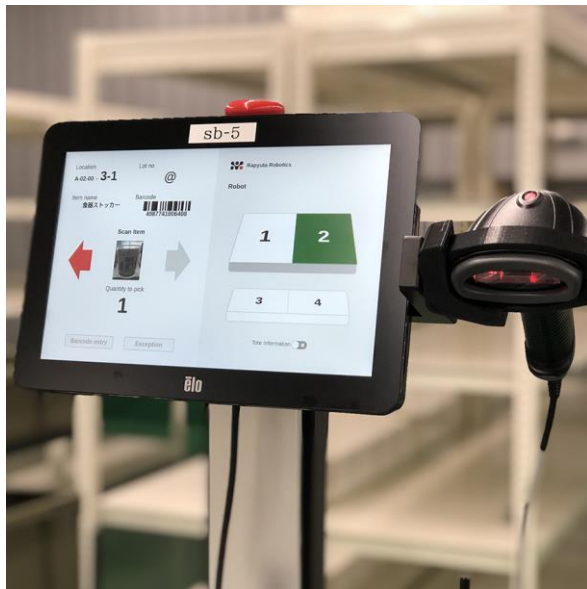
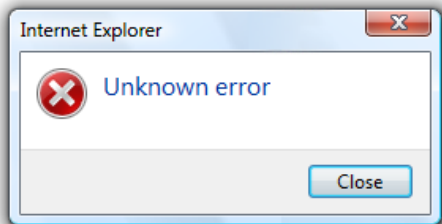
AI





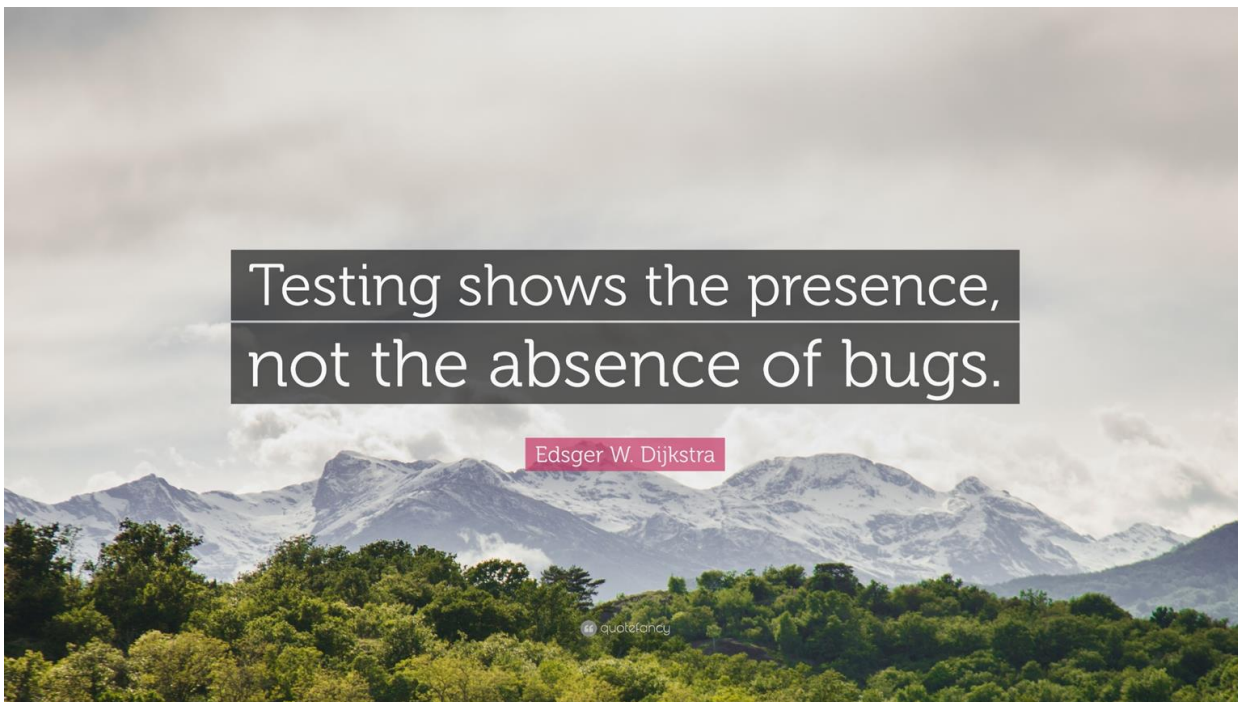
# Problem 4

## Usability



# Problem 5

## Recoverability



# Problem 6

## Reproduction of Issues



# Re-Play ROS Bag





# Problem 7

## Customized Requirements



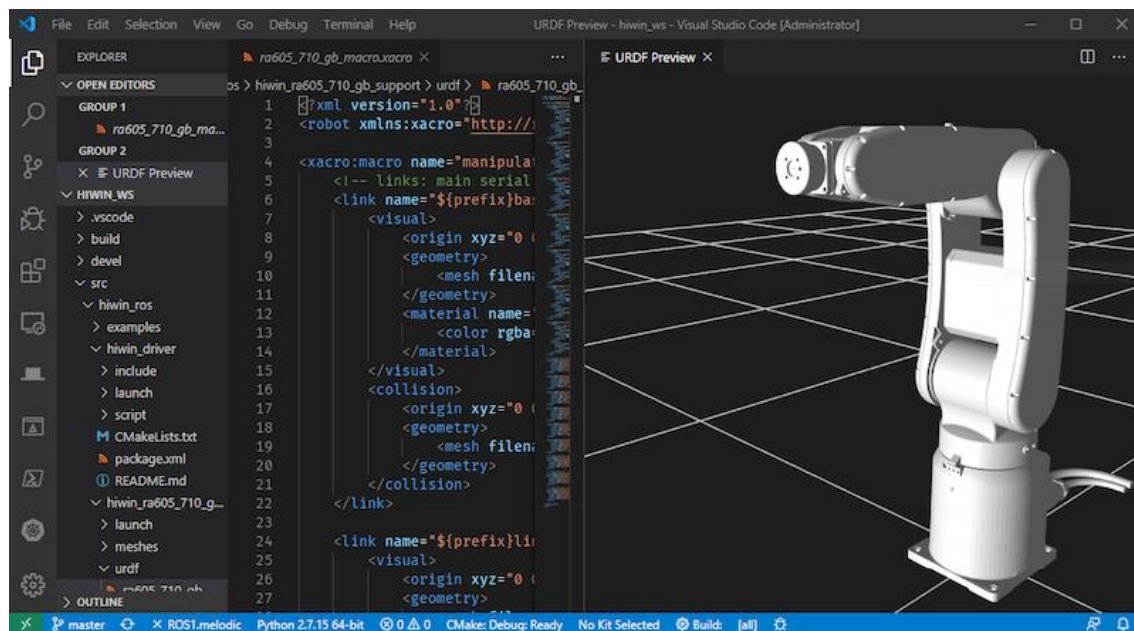
# Problem 8

## Domain Knowledge



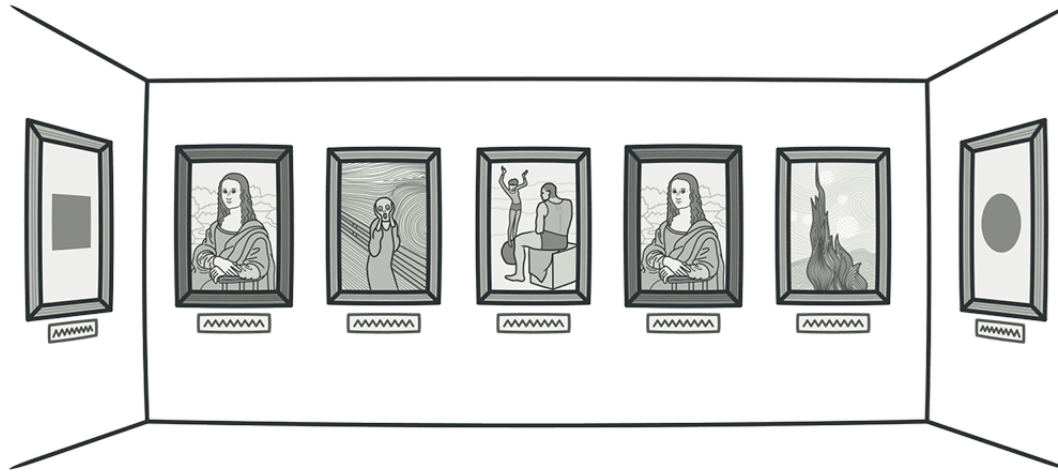
# Problem 9

## Technical Stack



# Problem 10

## Maintenance



# Testing

## Revisit: Example Scenario

### Tags

```
@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 = WaitHelper()
    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': 'x',
        'unload_time': '-',
    }
```

Helpers: Remove Duplication

```
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('%Y-%m-%d %H:%M:%S.%f')[:-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,
)
```

POM

Smart waits





# 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





mesutdurukal



@DurukalMesut



mesutdurukal.com

*Define  
Problems*

*Don't hesitate  
to try*



*Collaborate*

*Share &  
Communicate*



**HUSTEF**

HUNGARIAN SOFTWARE TESTING FORUM