GOOD MORNING!
早上好!
안녕하세요!

DAY 2

HOW TO WORK TOGETHER

- Participate, Participate, Participate!!!
- No long emails or Kakaotalk, prefer face to face
- Be open to suggestions and idea
- Be proactive, take initiative
- HOW is as important as WHAT

BRAINSTORMING RULES

- Every input is good input
- Do not critique inputs only seek to understand
- Organize inputs into logical groupings
- Sequence or show relationships as needed
- Use Posted Notes on Flip Chart

DAY 1 (DONE?)

- Welcome
- Project Introduction
- Introduction to Project Development Process
- Business Requirement Development
- System Requirement Development
- System(High Level) Design
- Time Management

DAY 2

- YOLOv8 기반 데이터 수집/학습/deploy (Detection Alert)
  - 감시용 데이터 수집(bus, truck, tank 등)
  - 감시용 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Detection
- Porting to ROS
  - Create Detection Alert Node
  - Generate Topics to send image and Obj. Det. results
  - Create Subscriber node and display image and print data from the Topic

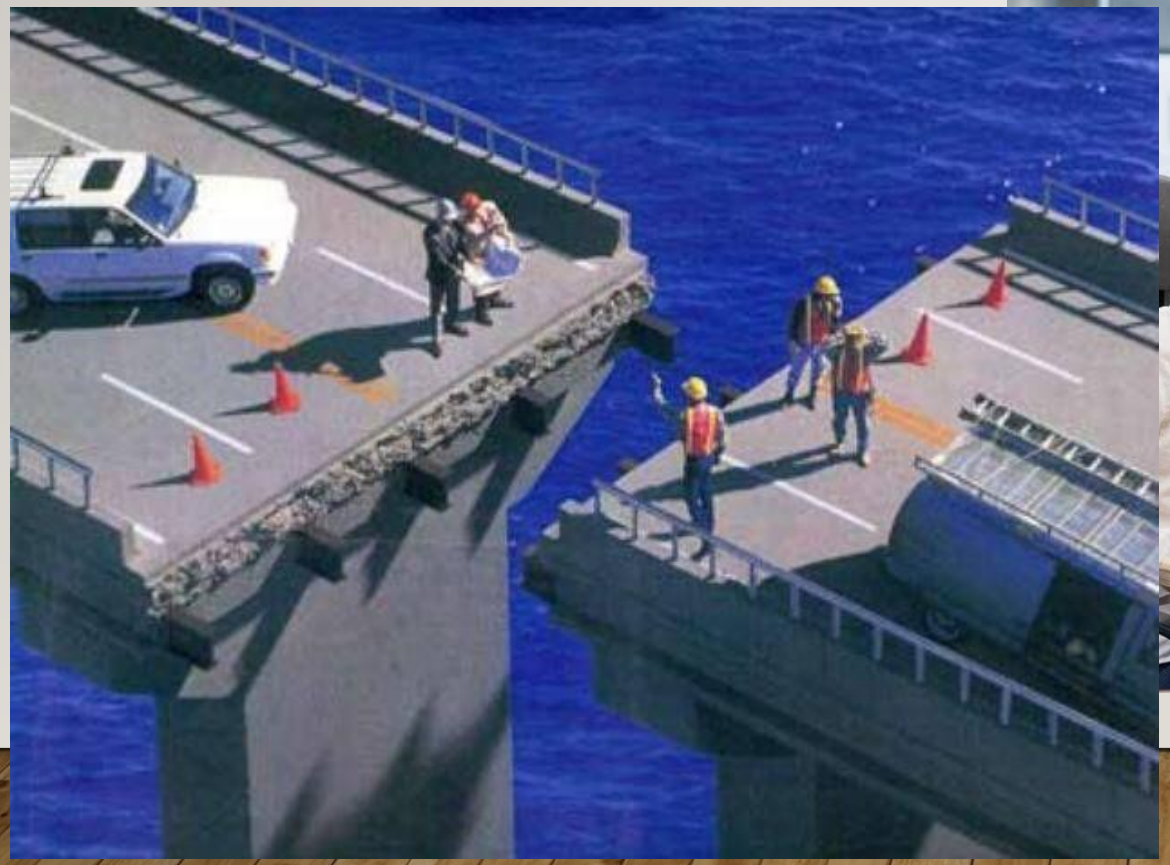
DAY 3

- AMR (Autonomous Mobile Robot)기반 카메라 인식 autonomous driving 시스템 with obstacle avoidance 구축 (AMR Controller)
  - Digital Mapping of environment
  - Goal Setting and Obstacle Avoidance using Navigation
  - Object Tracking w/ AMR camera
  - Control logic between navigation/obj. tracking/ obj. following (teleop)
- Porting to ROS
  - Create AMR Controller Node
  - Create and send Obj. Tracking Image and data to Sysmon
- Integrate and test with Detection

프로젝트 RULE

80/20 → 20/80

TEAMWORK AND PROJECT MANAGEMENT



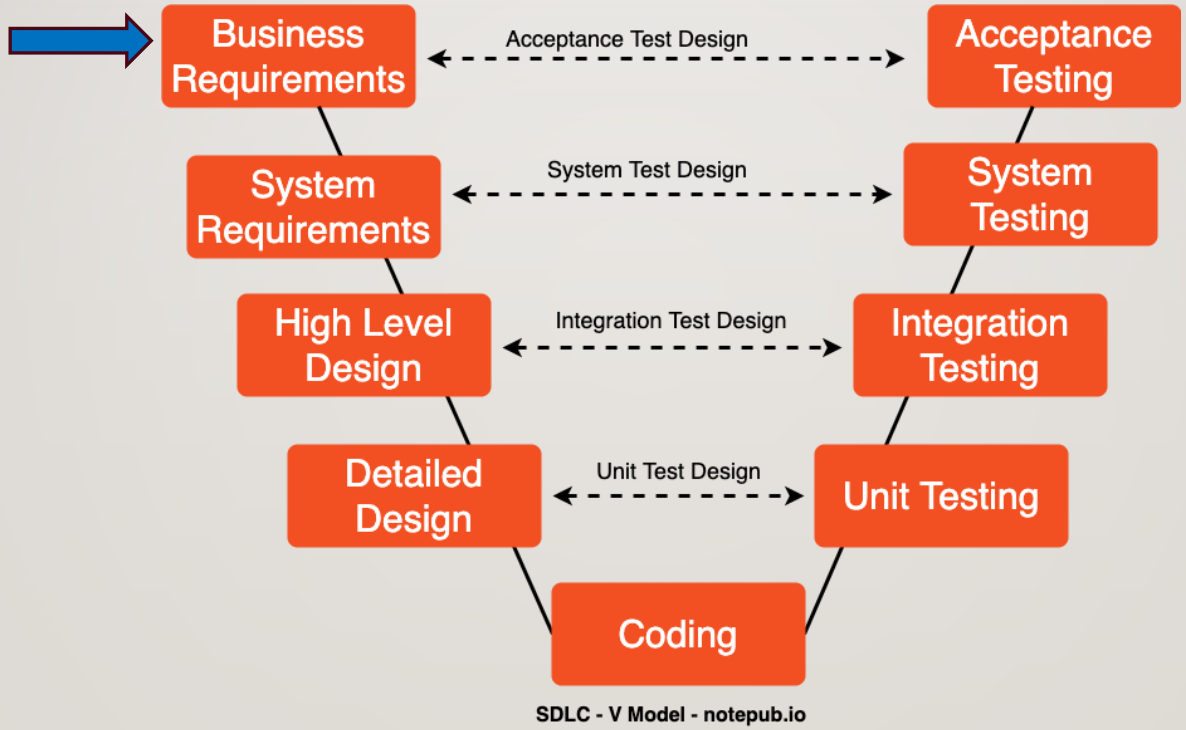
프로젝트 RULE NUMBER ONE!!!

Have Fun Fun Fun!

PROJECT DEVELOPMENT IS A PROCESS

SW DEVELOPMENT PROCESS

Why and What

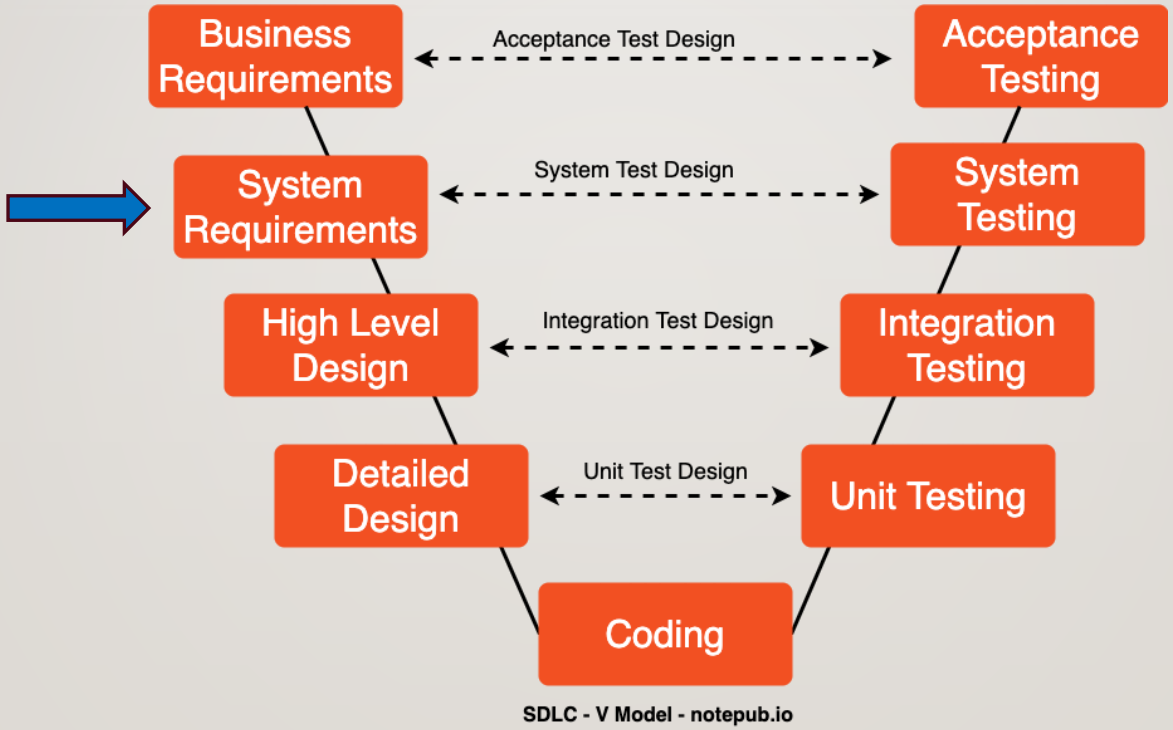


TEAM EXERCISE 1

Brainstorm Business Requirement for the project and write business requirement statement

SW DEVELOPMENT PROCESS

With What
And
How Much

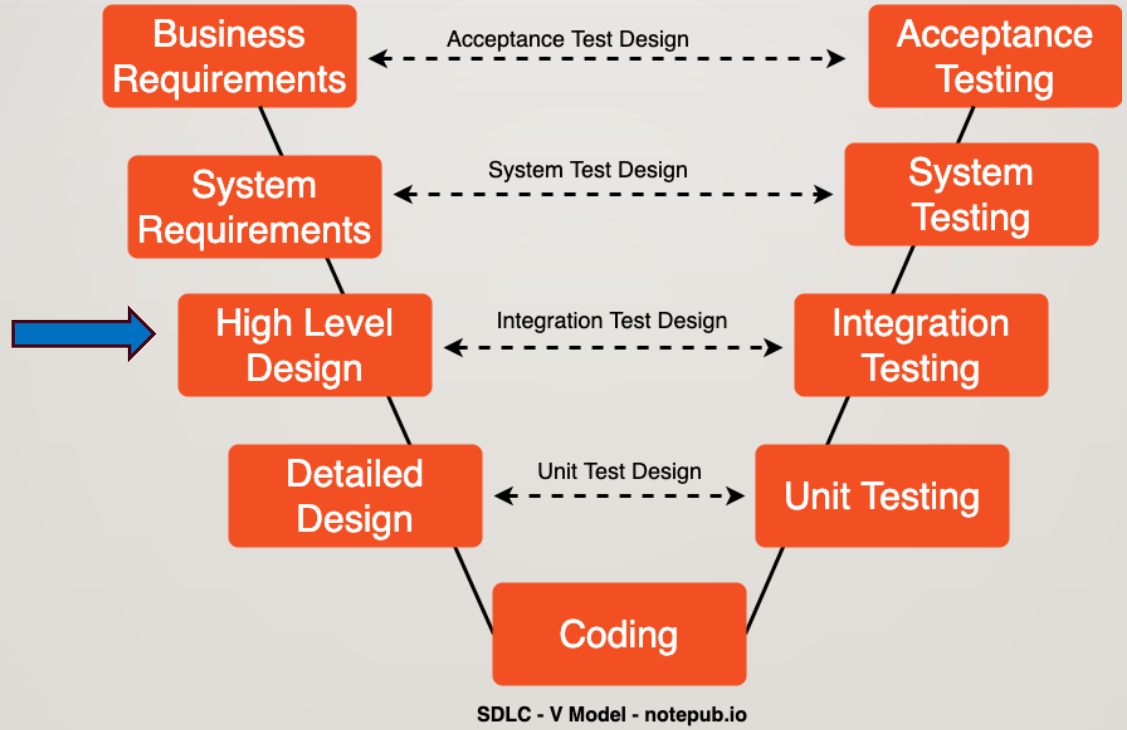


TEAM EXERCISE 2

Brainstorm System Requirement for the project and document

SW DEVELOPMENT PROCESS

HOW:
Where,What,
When,How Much



KEY SUBSYSTEM (MODULES) TO DEVELOP

- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems
- AMR Controller
  - Receive messages and act accordingly
  - Move using (SLAM) with Obstruction avoidance
  - Target Acquisition (Obj. Det.) and Tracking
  - Follow target using camera and motor control
- System Monitor (OPTION)
  - Receive and Display Detection Camera and info
  - Receive and Display AMR Camera and info
  - Store, display, and report Information and Alerts

VISUALIZATION – SYSTEM FUNCTIONAL PROCESS FLOW DIAGRAMS

- To-Be Functional Process Flow Diagram

  Detection Alert

  System Monitor

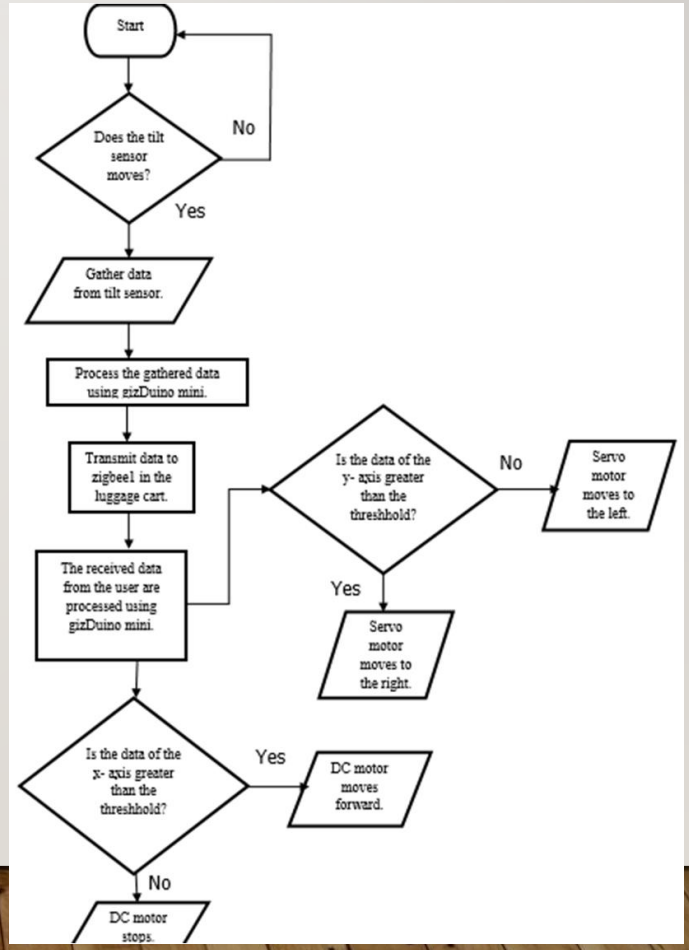
  AMR Controller

- Functions
- Interfaces

  Dataflow

- Testing

  Error and Exception Handling



TEAM EXERCISE 3

Create System Design using Process Flow Diagram.

Use the posted notes and flipchart as needed

SYSTEM DESIGN PRESENTATION BY EACH TEAM

EXAMPLE SYSTEM DESIGN DOCUMENT

System Design Document (SDD)

Project Title: Autonomous Mobile Robot (AMR) Security System
Version: 1.1
Date: [Insert Date]

1. Overview

The Autonomous Mobile Robot (AMR) Security System is designed to provide autonomous patrolling, threat detection, and alerting within a secure area using a single AI-enabled robot. The system consists of one AMR equipped with necessary hardware and software components to operate independently, processing data on-board without the need for a central server.

2. System Architecture

Since the system consists of a single AMR, data processing, navigation, threat detection, and alerting are all performed locally on the AMR itself. The AMR communicates directly with a user interface on a PC via a local network (Wi-Fi) for monitoring, alerts, and manual override if required.

시스템 설계 문서 (SDD)

프로젝트 제목: 자율 이동 로봇(AMR) 보안 시스템
버전: 1.1
날짜: [날짜 삽입]

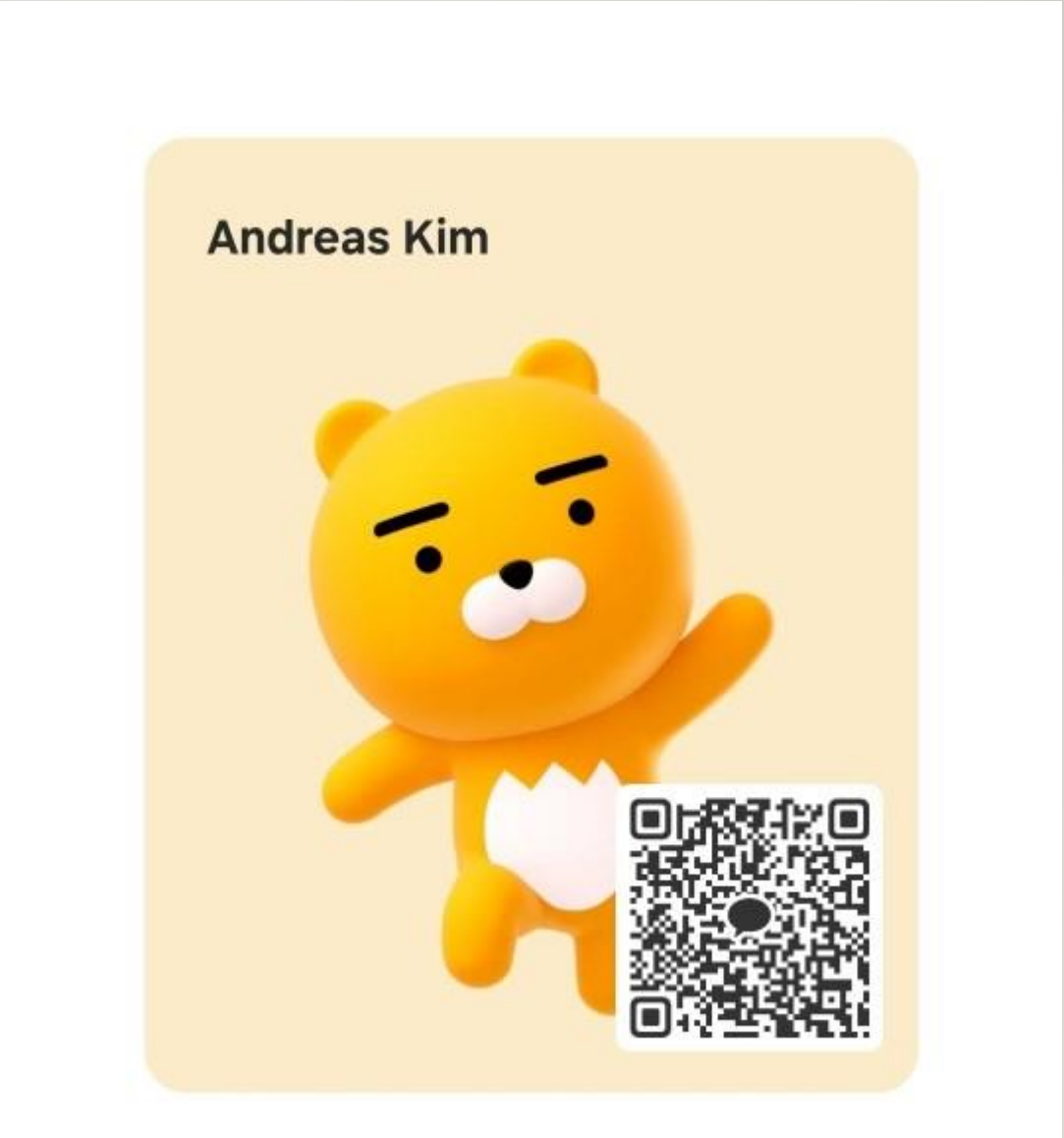
1. 개요

자율 이동 로봇(AMR) 보안 시스템은 단일 AI 기반 로봇을 사용하여 보안 구역 내에서 자율 순찰, 위협 탐지 및 경고를 제공하도록 설계되었습니다. 시스템은 단일 AMR 이 독립적으로 작동할 수 있도록 필요한 하드웨어 및 소프트웨어 구성 요소로 구성되며, 중앙 서버 없이 데이터를 현장에서 처리합니다.

2. 시스템 아키텍처

이 시스템은 단일 AMR 으로 구성되므로 데이터 처리, 네비게이션, 위협 탐지 및 경고가 모두 AMR 에서 로컬로 수행됩니다. AMR 은 모니터링, 알림 및 수동 제어를 위해 PC 의 사용자 인터페이스와 로컬 네트워크(Wi-Fi)를 통해 직접 통신합니다.

Send System Design Doc. Here:



PROJECT TIMELINE/CRITICAL PATH ITEM MANAGEMENT

EX. IMPLEMENTATION TIMELINE

Function Backlog	Owner	5월 20일	5월 21일	5월 22일	5월 23일	5월 24일	5월 25일
Unloading Module	John						
Input1	John						
Input2	John						
Output 1	John						
Unit Test	John						
Receiving Module	Jan						
Input1	Feb						
Input2	Mar						
Output 1	Apr						
Unit Test	John						
Integration Test	John/Jan						

이 타임라인을 생성할 때 먼저 시스템 및 시스템 설계의 기능 프로세스 다이어그램(To-Be)을 완료해야 합니다.

그런 다음 각 기능(하위 함수/모듈 및 입력/출력)에 대해 누가, 무엇을, 언제, 어떻 게를 정의합니다. 표에 설명 타임라인 형식의 무엇을, 누가, 언제를 입력합니다.

CRITICAL PATH ITEMS LIST

- tasks that directly impact the project timeline. Delays in these tasks would delay the project's overall completion because they represent the longest stretch of dependent activitie

- 프로젝트 타임라인에 직접적인 영향을 주는 작업입니다. 이러한 작업이 지연되면 종속 활동이 가장 길어지기 때문에 프로젝트의 전체 완료가 지연됩니다s

CRITICAL PATH ITEMS LIST

- Examples:
  1. **AI Model Development**: Fine-tuning deep learning models like CNNs for precise item recognition and sorting, requiring data gathering, model training, and testing.
  2. **Robot Integration**: Embedding AI software into robots to enable precise task execution, focusing on software-hardware compatibility and function tests.
  3. **System Testing**: Comprehensive testing of AI and robot performance in simulated environments to ensure operational reliability.

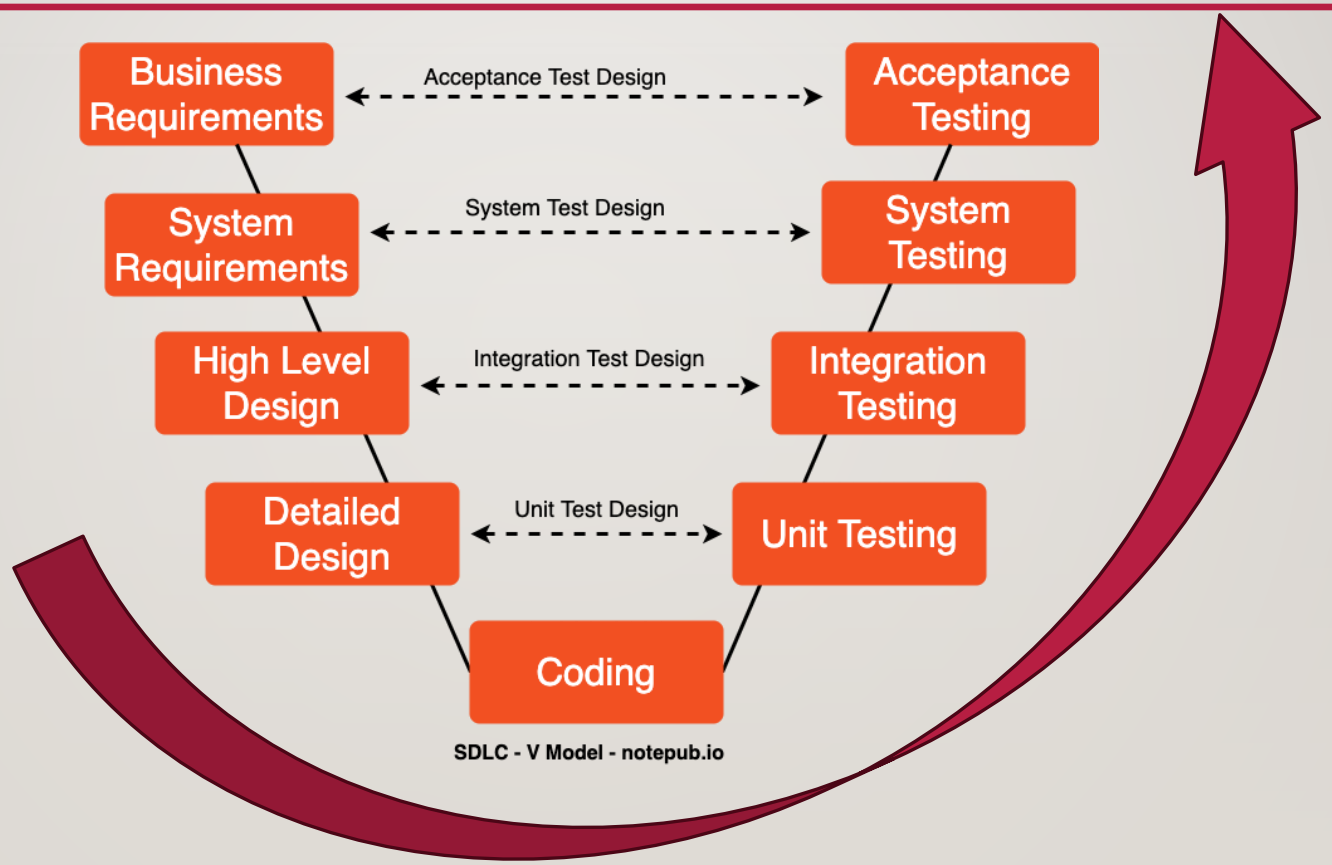
- AI 모델 개발: 정확한 항목 인식 및 정렬을 위해 CNN과 같은 딥 러닝 모델을 미세 조정하여 데이터 수집, 모델 학습 및 테스트가 필요합니다.
- 로봇 통합:AI 소프트웨어를 로봇에 내장하여 소프트웨어-하드웨어 호환성 및 기능 테스트에 중점을 두고 정확한 작업 실행을 가능하게 합니다.
- 시스템 테스트: 시뮬레이션 환경에서 AI 및 로봇 성능을 종합적으로 테스트하여 운영 안정성을 보장합니다.

GUIDE TO PROGRESS INDICATORS

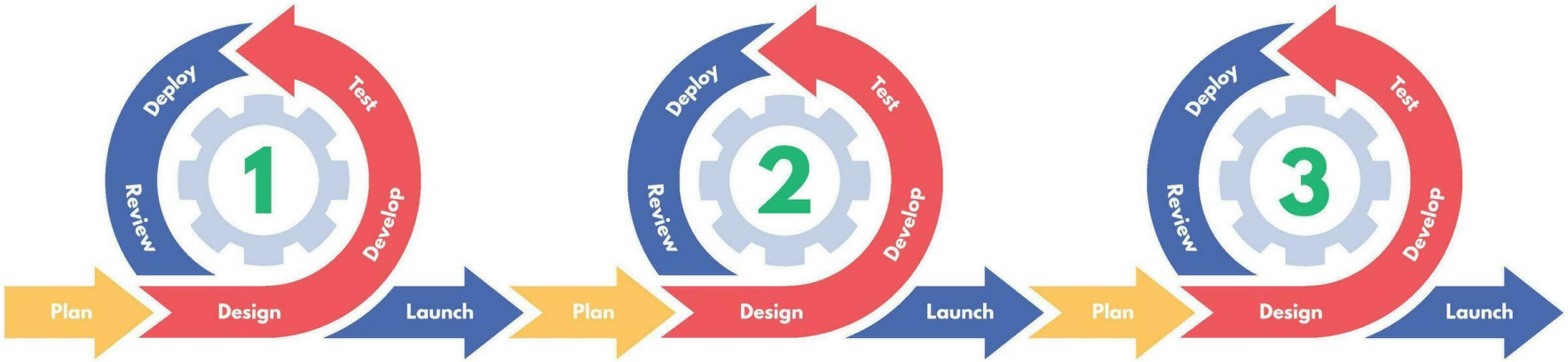
- Project Timeline
  - Green – less 10% of the listed items are delayed
  - Yellow – more than 10% but less than 20% of listed items are delayed
  - Red – more than 20% of listed items are delayed
- Critical Path Items
  - Green – reduced number of item(s)
  - Yellow – no new item(s)
  - Red – additional item(s)

DETAIL DESIGN TO USER ACCEPTANCE

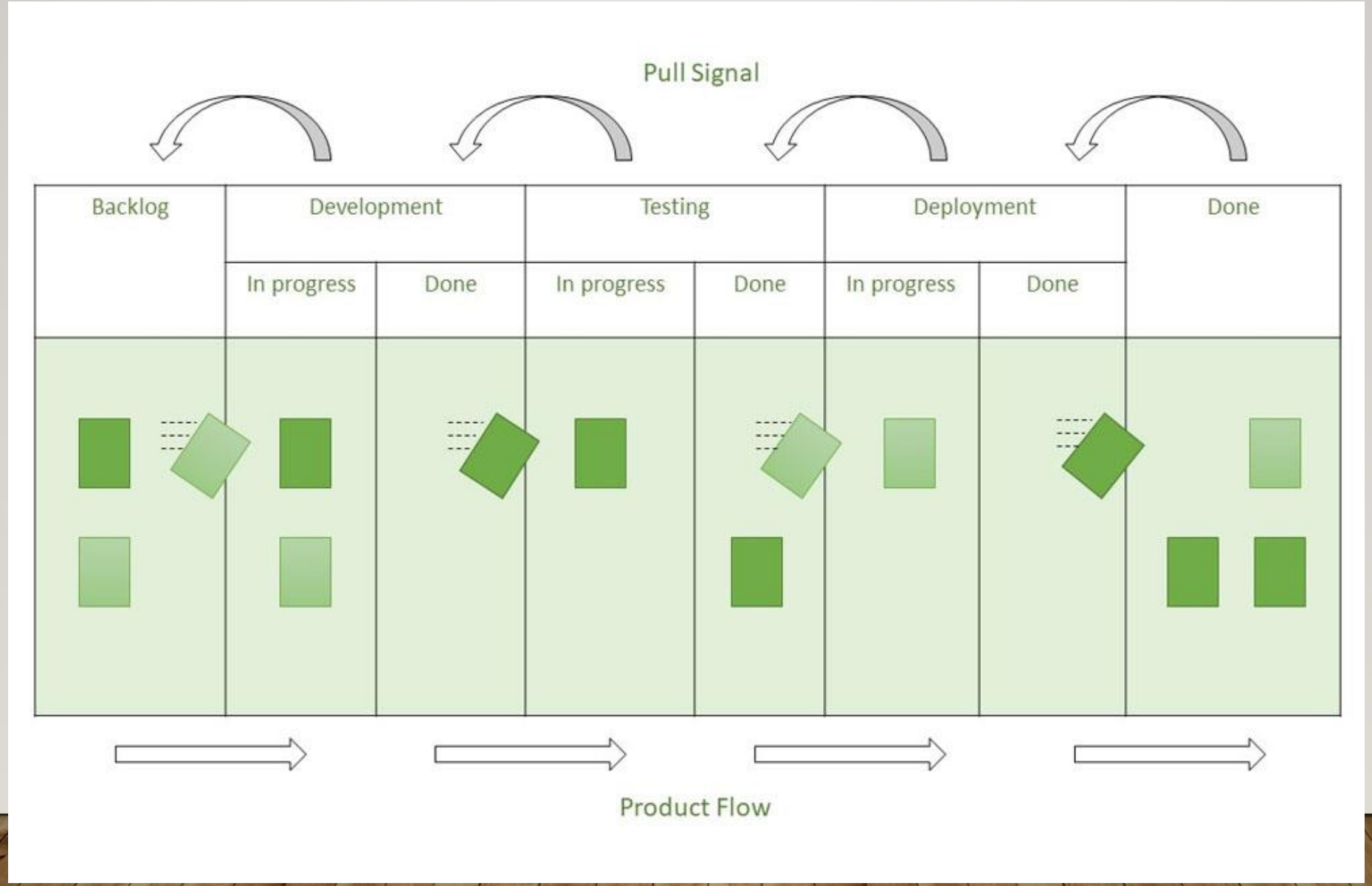
SW DEVELOPMENT PROCESS



AGILE DEVELOPMENT



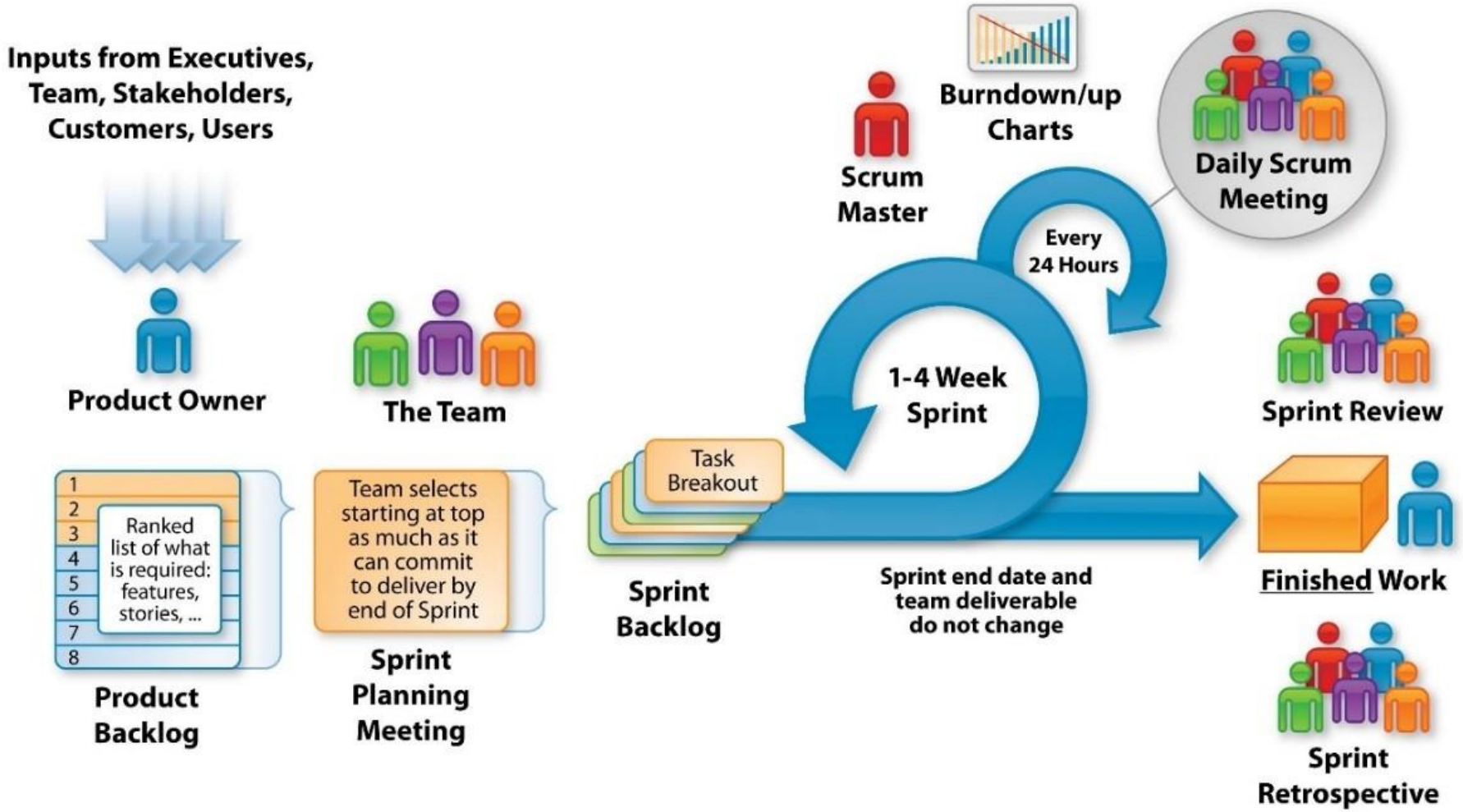
KANBAN METHODOLOGY




TO DO
IN PROGRESS
TESTING


FINANCIAL ANALYTICS
ESSET ALLOCATION
ANALYZE
DEBT
PROFIT
NEW VALUE CURVE

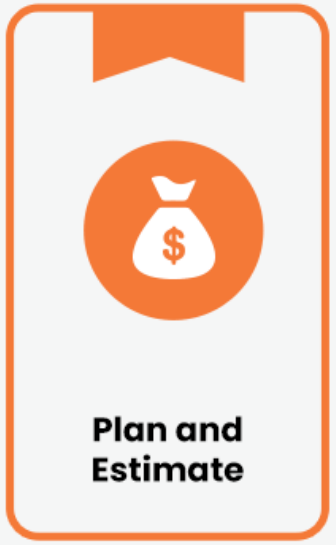
The Agile - Scrum Framework



5 Stages of Scrum Sprint

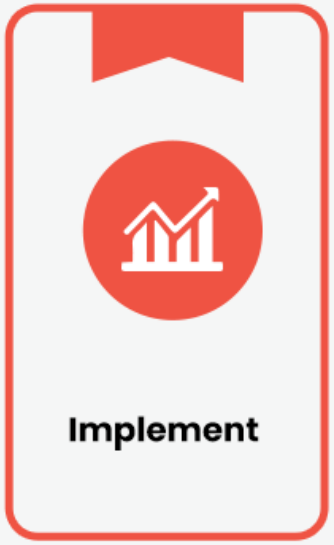
Initiate

This phase includes the processes related to the commencement of a project, such as a scope and objectives, creating and distributing its charter, and taking other steps to guarantee success.



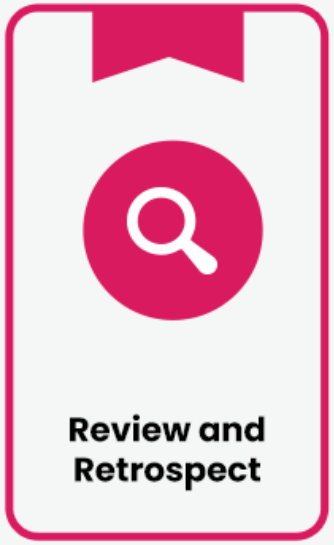
Plan and Estimate

This phase involves planning and estimating processes, including creating user stories, approving, assessing, committing user stories, creating tasks, evaluating tasks, and creating a Sprint backlog.



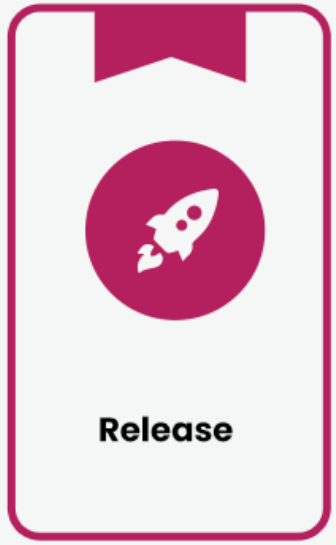
Implement

This phase is about executing the tasks and activities to create a product. These activities include building the various outputs, conducting daily standup meetings, and grooming the product backlog.



Review and Retrospect

This stage of the project lifecycle is concerned with evaluating what has been accomplished so far, whether the team has worked to plan, and how it can do things better in the future.

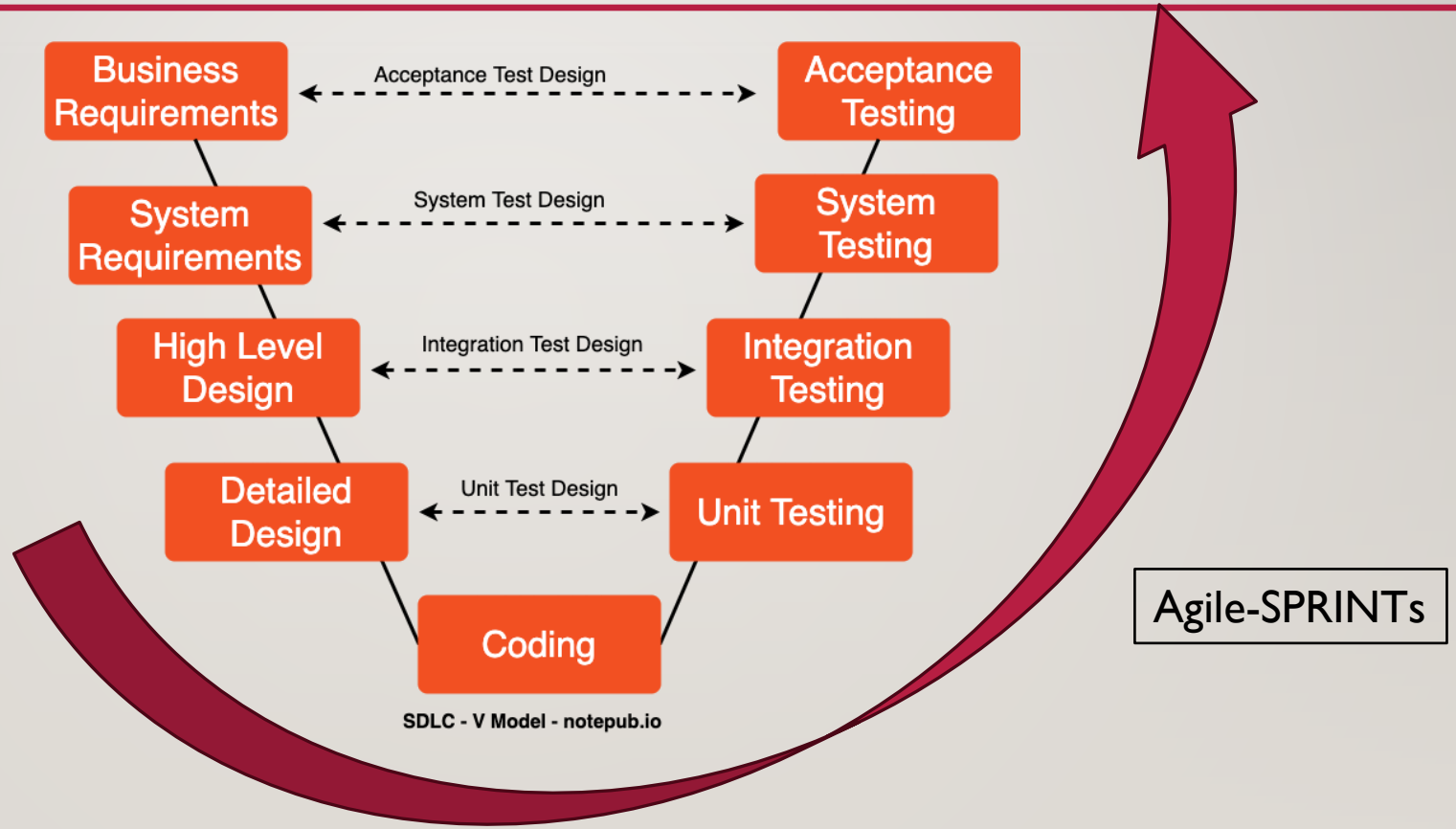


Release

This stage highlights delivering the accepted deliverables to the customer and determining, documenting, and absorbing the lessons learned during the project.

maruti techlabs

SW DEVELOPMENT PROCESS



Agile-SPRINTs

PROJECT SPRINTS

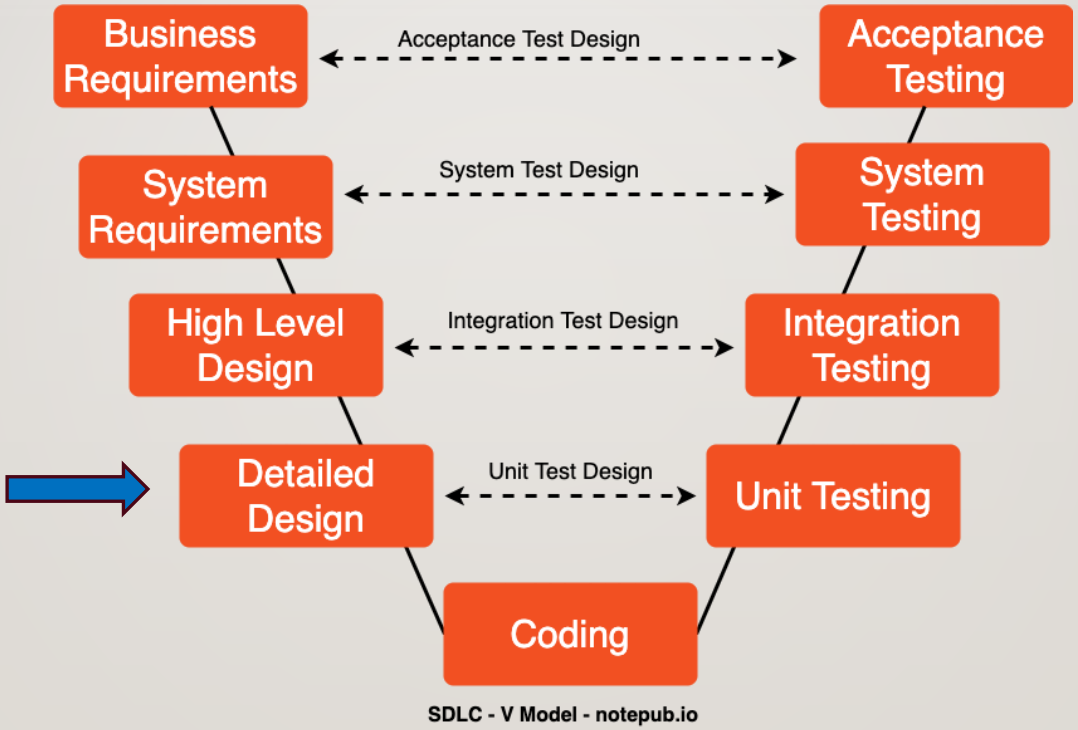
- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems

- AMR Controller
  - Receive messages and act accordingly
  - Move using (SLAM) with Obstruction avoidance
  - Target Acquisition (Obj. Det.) and Tracking
  - Follow target using camera and motor control

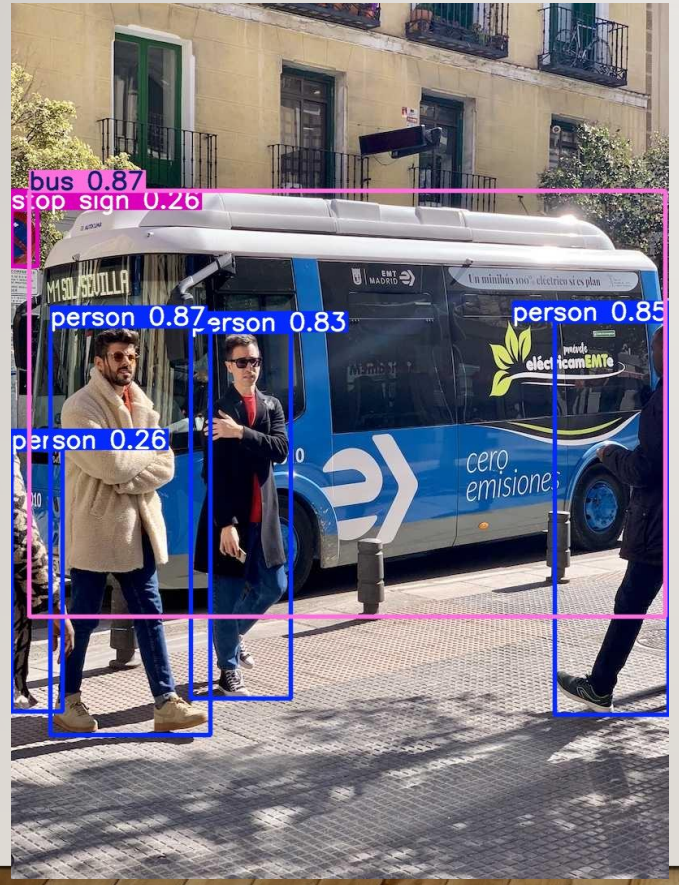
- System Monitor
  - Receive and Display Detection Camera and info
  - Receive and Display AMR Camera and info
  - Store, display, and report Information and Alerts

DETECTION ALERT SPRINT

SPRINT 1 - DETECTION ALERT



YOLO OBJ. DET. VS. YOLO TRACKING



- (469) YOLO people detection + SORT tracking – YouTube
- Bing Videos
- Track - Ultralytics YOLO Docs

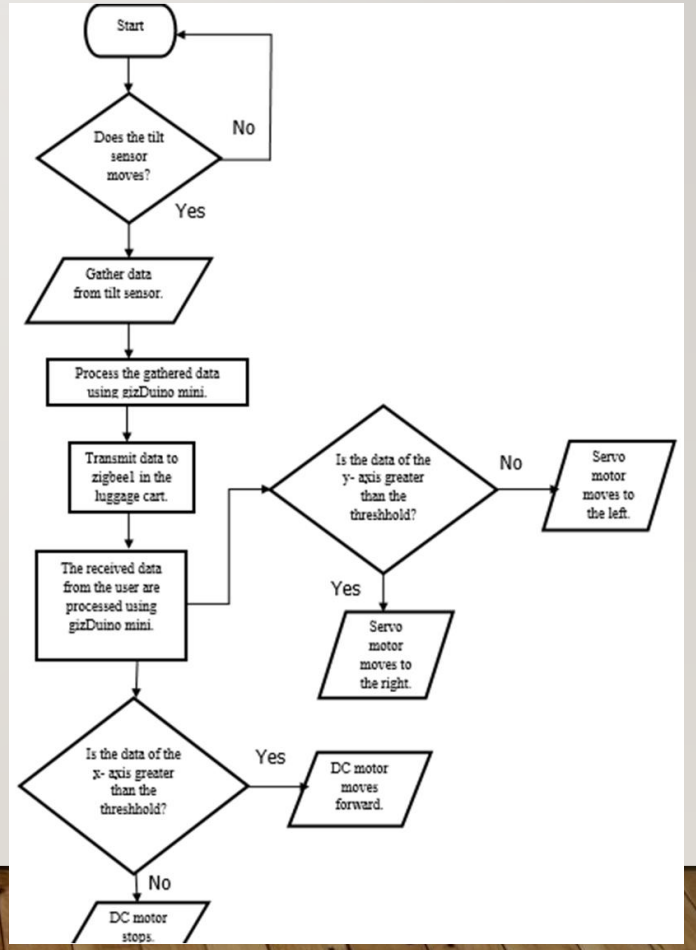
VISUALIZATION – DETAILED FUNCTIONAL PROCESS DIAGRAMS

- To-Be Functional Process Diagram

Detection Alert

PC or AMR or Both

Remember F.I.T.!!



TEAM EXERCISE 4

Perform Detail Design of Detection Alert Module using Process Flow Diagram

DETAIL DESIGN REVIEW BY EACH TEAM

Using the process flow diagram present team's design

EXAMPLE DETAILED DESIGN DOCUMENT

Detailed Design Document: AMR Navigation and Threat Detection

Project Title: Autonomous Mobile Robot (AMR) Security System
Version: 1.0
Date: [Insert Date]

1. Overview

This document outlines the detailed design for the Autonomous Mobile Robot (AMR) navigation and threat detection components. It covers the architecture, algorithms, data processing, and system interactions necessary to enable autonomous navigation within a secure area and real-time threat detection using onboard sensors.

2. System Architecture

The AMR system relies on onboard hardware (e.g., sensors, cameras, Jetson-Orin processor) and software (ROS2, OpenCV, YOLO) for autonomous navigation and real-time threat detection. All processing occurs locally on the AMR, with the capability to transmit alerts to a monitoring PC via Wi-Fi.

상세 설계 문서: AMR 네비게이션 및 위협 탐지

프로젝트 제목: 자율 이동 로봇(AMR) 보안 시스템
버전: 1.0
날짜: [날짜 삽입]

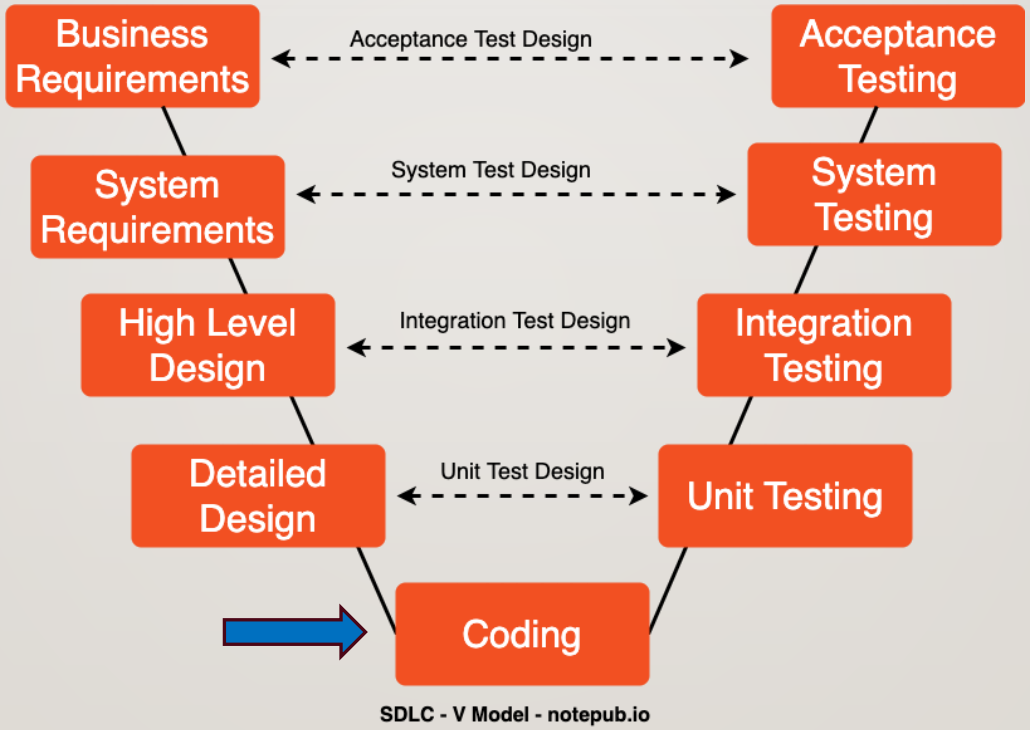
1. 개요

이 문서는 자율 이동 로봇(AMR)의 네비게이션 및 위협 탐지 구성 요소에 대한 상세 설계를 다룹니다. 자율 네비게이션과 실시간 위협 탐지를 위해 온보드 센서를 사용하는 데 필요한 아키텍처, 알고리즘, 데이터 처리 및 시스템 상호작용이 포함되어 있습니다.

2. 시스템 아키텍처

AMR 시스템은 자율 네비게이션 및 실시간 위협 탐지를 위해 온보드 하드웨어(예: 센서, 카메라, Jetson-Orin 프로세서)와 소프트웨어(ROS2, OpenCV, YOLO)를 활용합니다. 모든 처리는 AMR 내에서 로컬로 수행되며, 잠재적인 위협이 감지되면 Wi-Fi 를 통해 모니터링 PC 로 알림을 전송할 수 있습니다.

SPRINT 1 - DETECTION ALERT



USEFUL COMMANDS

$ lsb_release –a

$ echo $ROS_DISTRO

$ code --version

$ python3 --version

- Linux distribution info
- ROS: Humble
- Vscode
- Python

$ sudo apt update

$ sudo apt upgrade

$ python -m ensurepip –upgrade

- Assumes Linux (Ubuntu 22.04), ROS Humble, VScode, and Python are already installed globally

REQUIRED PACKAGES SETUP

$ pip list | grep opencv

If doesn't exist….

$ pip3 install opencv-python

$ pip3 install opencv-contrib-python

$ pip list | grep ultra

If doesn't exist….

$ pip install ultralytics

$ pip freeze > requirements.txt

$ pip install -r requirements.txt

ROS2 DEVELOPMENT WORKSPACE

```
$ cat ~/.bashrc

$ echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc  #check path

$ sudo apt install python3-colcon-common-extensions
$ echo "source /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash" >> ~/.bashrc
  #check path

$ source ~/.bashrc
```

PERFORM DATA COLLECTION FOR DETECTION ALERT

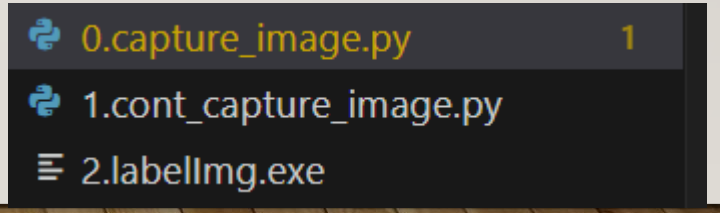
CODING HINTS

- create your team working folder and place files and work from here

$ mkdir ~/rokey2_<grp_letter><grp_num>_ws

(i.e. mkdir ~/rokey2_A2_ws)

- Image Capture

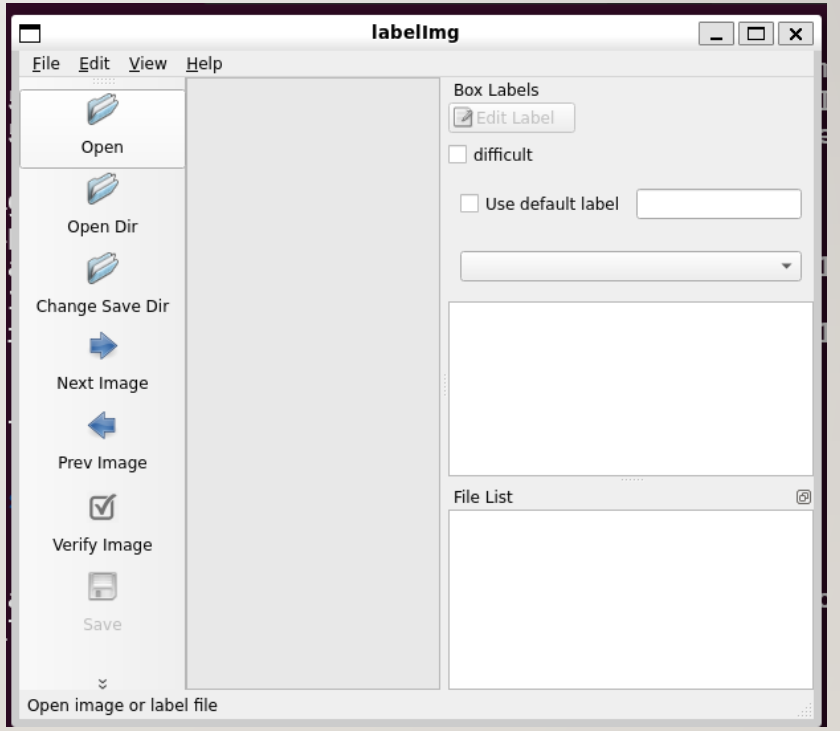


CODING HINTS

- Data Labelling : use previously installed LabelImg

- Or

- pip3 install PyQt5 lxml
- pip3 install labelImg
- labelImg



CODING HINTS

- Data Labelling : LabelImg

라벨링 순서

1. 이미지파일 불러오기 (Open Dir)
2. 저장형식 변경 (PascalVOC, YOLO)
3. 이미지 선택
4. 바운딩 박스 그리기(create rectbox)
5. Class 지정
6. 저장경로 생성 및 변경(Change Save Dir)
7. 저장(Save)

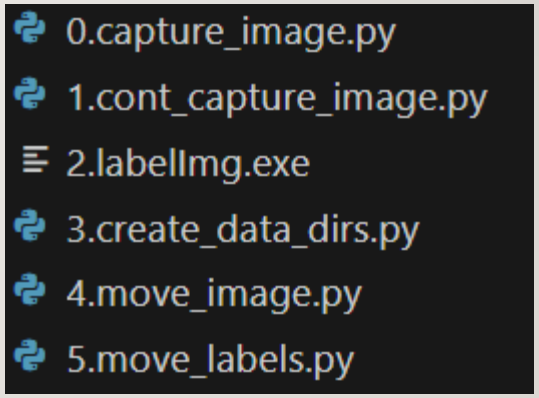
단축키

Ctrl + u	Load all of the images from a directory
Ctrl + r	Change the default annotation target dir
Ctrl + s	Save
Ctrl + d	Copy the current label and rect box
Ctrl + Shift + d	Delete the current image
Space	Flag the current image as verified
w	Create a rect box
d	Next image
a	Previous image
del	Delete the selected rect box
Ctrl++	Zoom in
Ctrl--	Zoom out
↑→↓←	Keyboard arrows to move selected rect box

CODING HINTS

- Image Capture
- Data Labelling

- Data Preprocessing

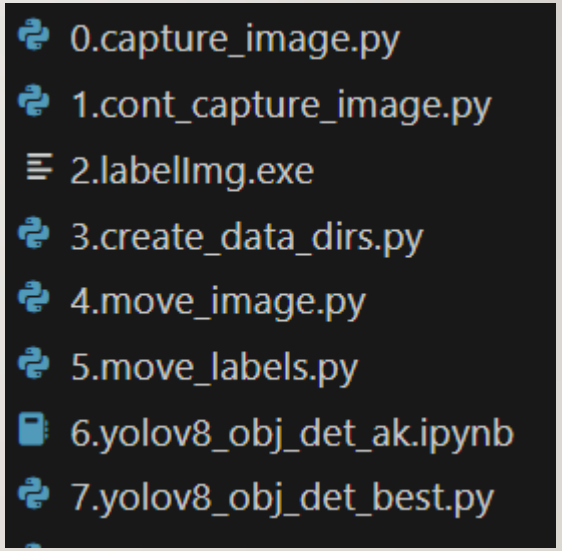


PERFORM YOLO TRAINING & INFERENCE FOR DETECTION ALERT

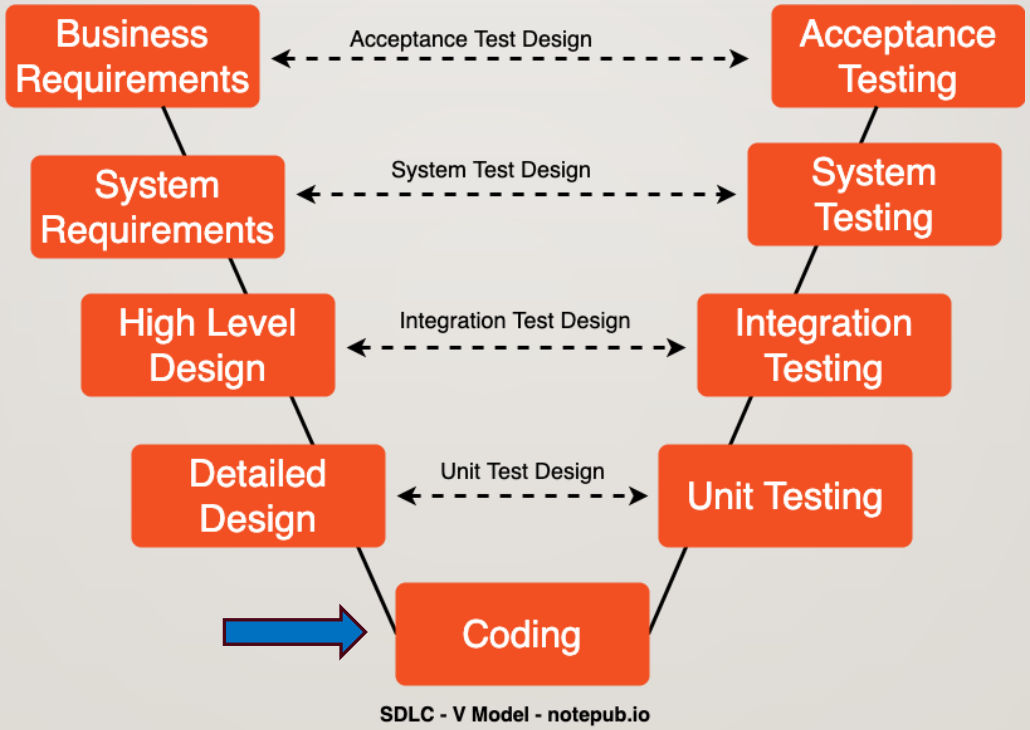
CODING HINTS

- Image Capture
- Data Labelling
- Preprocessing

- Yolo8 Object Det.

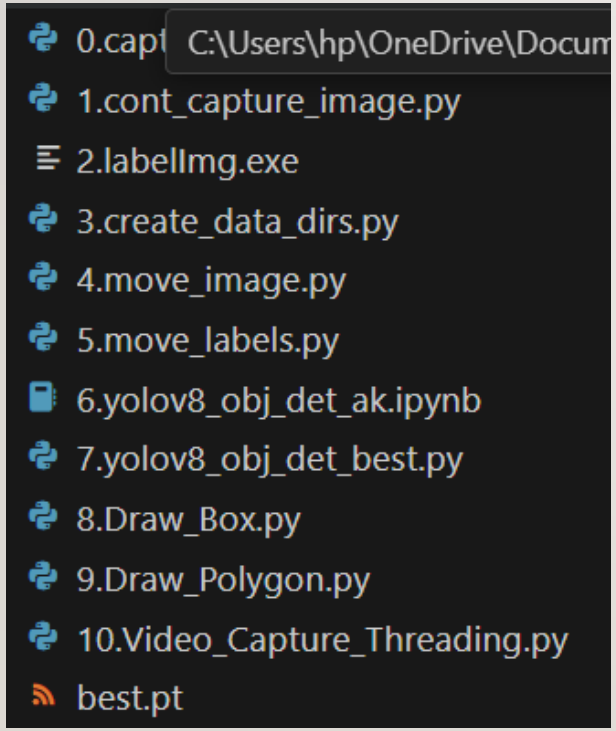


SPRINT 1 - DETECTION ALERT



HOW DID YOU/DO YOU NEED TO DEFINE A DETECTION AREA?

CODING HINTS



- Create your detection/alert condition

PORTING TO ROS

ROS2 DEVELOPMENT WORKSPACE

CREATE/GOTO WORKSPACE

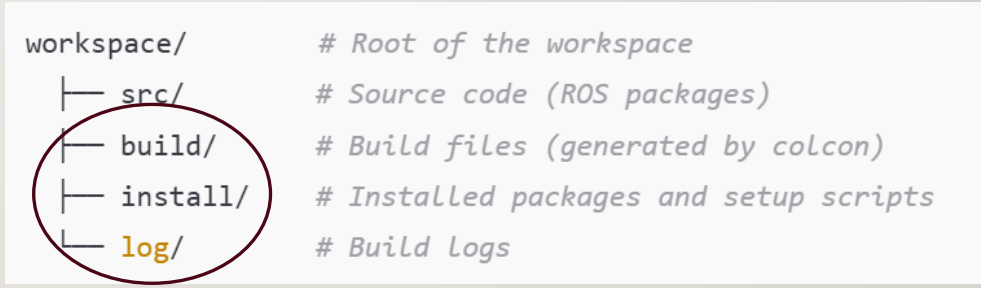
$ mkdir
~/rokey2_<grp_letter><grp_num>_ws

(i.e. mkdir ~/rokey2_A2_ws)

Or

$ cd ~/rokey2_A2_ws

*NOT CREATED UNTIL COLCON



ROS2 DEVELOPMENT WORKSPACE

CREATE ROS VIRTUAL ENVIRONMENT

If you want to usual virtual env, you should create it at the workspace level

$ cd ~/rokey2_A2_ws

$ python3 -m venv <NAME>

$ source <NAME>/bin/activate

$ deactivate

ROS2 DEVELOPMENT WORKSPACE

$ cd ~/rokey1_A2_ws

$ ros2 pkg create --build-type ament_python <my_package>

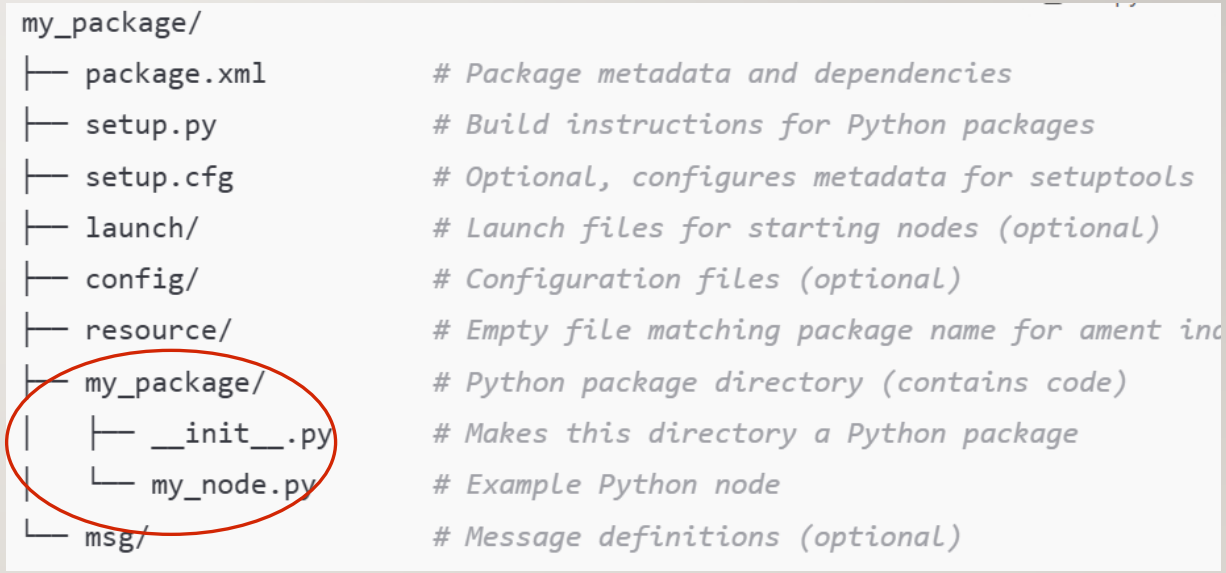
Or

$ ros2 pkg create <package_name> --build-type ament_python --dependencies rclpy std_msgs ament_index_python

```
my_package/
├── package.xml          # Package metadata and dependencies
├── setup.py             # Build instructions for Python packages
├── setup.cfg            # Optional, configures metadata for setuptools
├── launch/              # Launch files for starting nodes (optional)
├── config/              # Configuration files (optional)
├── resource/            # Empty file matching package name for ament ind
├── my_package/          # Python package directory (contains code)
│   ├── __init__.py      # Makes this directory a Python package
│   └── my_node.py       # Example Python node
└── msg/                 # Message definitions (optional)
```

ROS2 DEVELOPMENT WORKSPACE

Write you code below the my_package/ directory under my_package/ package directory



ROS2 DEVELOPMENT WORKSPACE

$ cd ~/rokey1_A2_ws

$ colcon build

or

$ colcon build --packages-select <package_name>

or

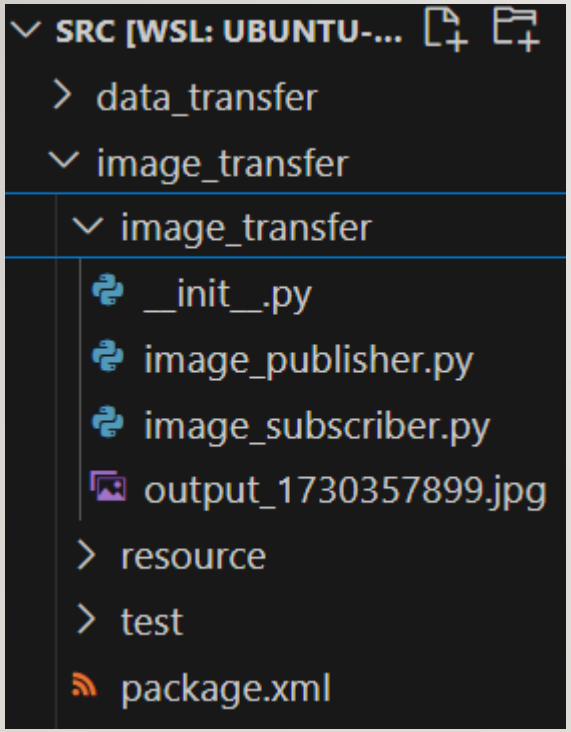
$ colcon build --packages-select <package_name> --packages-skip-up-to-date

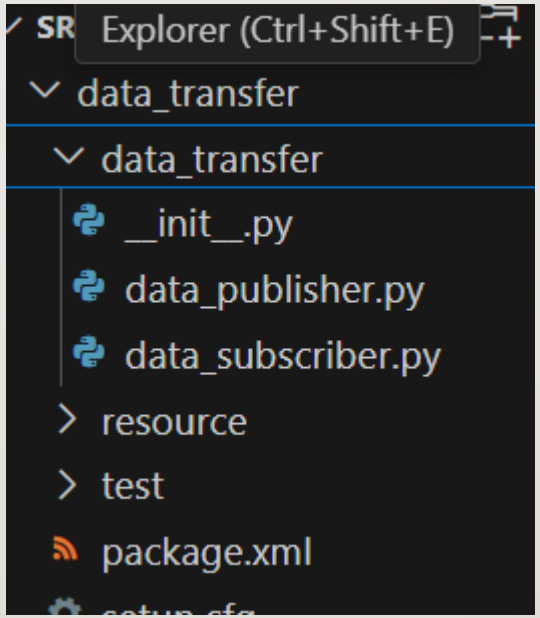
$ rm -rf build/<package_name> install/<package_name> log

$ colcon build --packages-select <package_name>

```
workspace/          # Root of the workspace
  ├── src/          # Source code (ROS packages)
  ├── build/        # Build files (generated by colcon)
  ├── install/      # Installed packages and setup scripts
  └── log/          # Build logs
```

ROS HINTS





$ ros2 interface list

ROS HINTS

- Edit setup.py under<pakage_name> directory add entry for each node

entry_points={ 'console_scripts':
['<command_name> =
<package_name>.<code_filename>:main',},

<command_name> is used when ros2 run is executed i.e. data_publisher

```
entry_points={
    'console_scripts': [
        'data_pub = data_transfer.data_publisher:main',
        'data_sub = data_transfer.data_subscriber:main',
    ],
},
```

ROS HINTS

$ cd ~/rokey1_A2_ws

$ source ~/rokey1_A2_ws/install/setup.bash

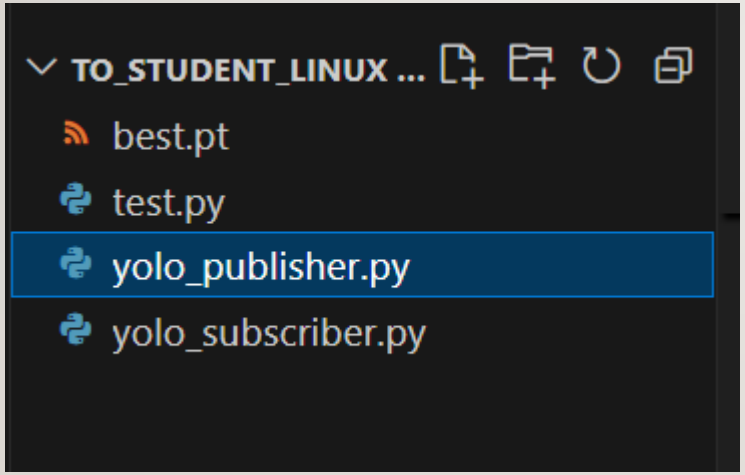
$ ros2 run <package_name> <command_name>

```
391  ros2 run data_transfer data_pub
```

$ sudo apt update

$ sudo apt install terminator

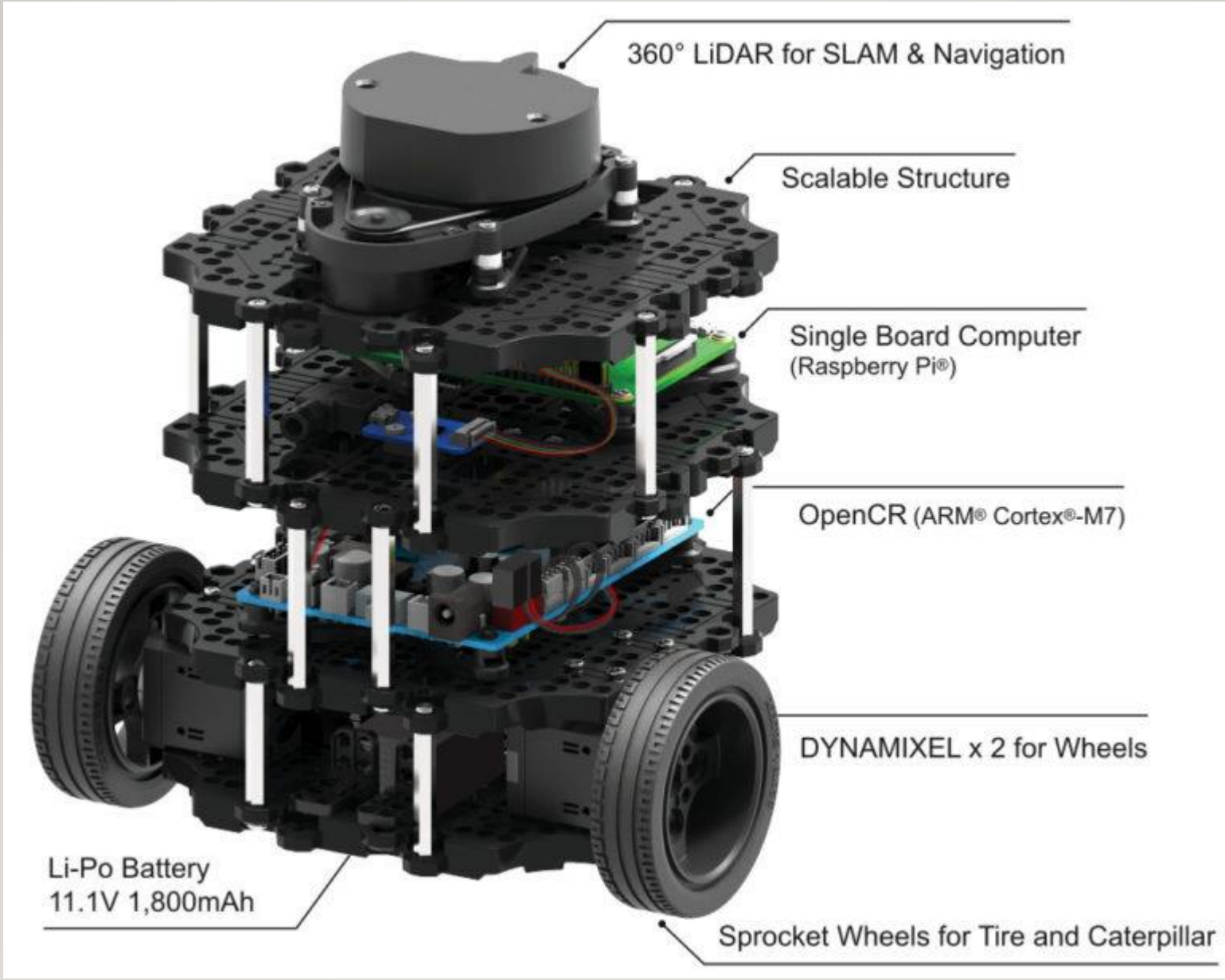
ROS HINTS



- $ ros2 run rqt_graph rqt_graph
- $ ros2 node list
- $ ros2 node info <node_name>
- $ ros2 topic list
- $ ros2 topic info <topic_name>
- $ ros2 topic echo /chatter
- $ ros2 interface list
- $ ros2 interface show <package_name>/msg/<MessageName>

INTRODUCTION TO AMR

- TurtleBot3
- https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/


360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer
(Raspberry Pi®)
OpenCR (ARM® Cortex®-M7)
DYNAMIXEL x 2 for Wheels
Li-Po Battery
11.1V 1,800mAh
Sprocket Wheels for Tire and Caterpillar

POWER ON AND OFF AMR (VERY IMPORTANT!!!!)

POWER ON

- Make sure both Jetson and OpenCR is powered off.
- Must turn on OpenCR first!!!!
- Wait for the music from OpenCR
- Finally turn on Jetson

POWER OFF

- Execute proper shutdown

```
$ sudo shutdown now
```

Turn off Jetson, Turn off OpenCR

Do **NOT INSTALL** any packages to AMR **WITHOUT** speaking with me first!!!

CONNECTING TO AMR -- SSH

Connect to AMR Hotspot (AP) from the PC wifi selection list

Turtlebot3_AP_<n>

Open a terminal window

$ dpkg -l | grep openssh

If not installed…

$ sudo apt install openssh-server –y

Connect to AMR via SSH

$ ssh -X rokey<n>@<ip_address>

HOW TO MOVE FILE FROM PC MOVE MAP TO AMR

$ scp <dir_path>/<file_name> rokey<n>@<rokey IP>:$HOME/<dir_path>/

SETTING UP VSCODE FOR REMOTE EDITING

- Install VSCode Remote - SSH Extension:
  - Open VSCode on your local machine.
  - Go to the Extensions view (Ctrl + Shift + X).
  - Search for "Remote - SSH" and install it.

- VSCode Remote - SSH 확장 프로그램 설치:
  - 로컬 컴퓨터에서 VSCode를 엽니다.
  - 확장 프로그램 보기로 이동합니다 (Ctrl + Shift + X).
  - "Remote - SSH"를 검색하여 설치합니다.

SETTING UP VSCODE FOR REMOTE EDITING

- Connect to the Remote Server:
  - Press F1 or Ctrl + Shift + P to open the Command Palette.
  - Type Remote-SSH: Connect to Host and select it.
  - Enter the SSH connection string (e.g., user@hostname) and connect.

- 원격 서버에 연결:
  - F1 또는 Ctrl + Shift + P를 눌러 명령 팔레트를 엽니다.
  - Remote-SSH: Connect to Host를 입력하고 선택합니다.
  - SSH 연결 문자열(예: user@hostname)을 입력하고 연결합니다.

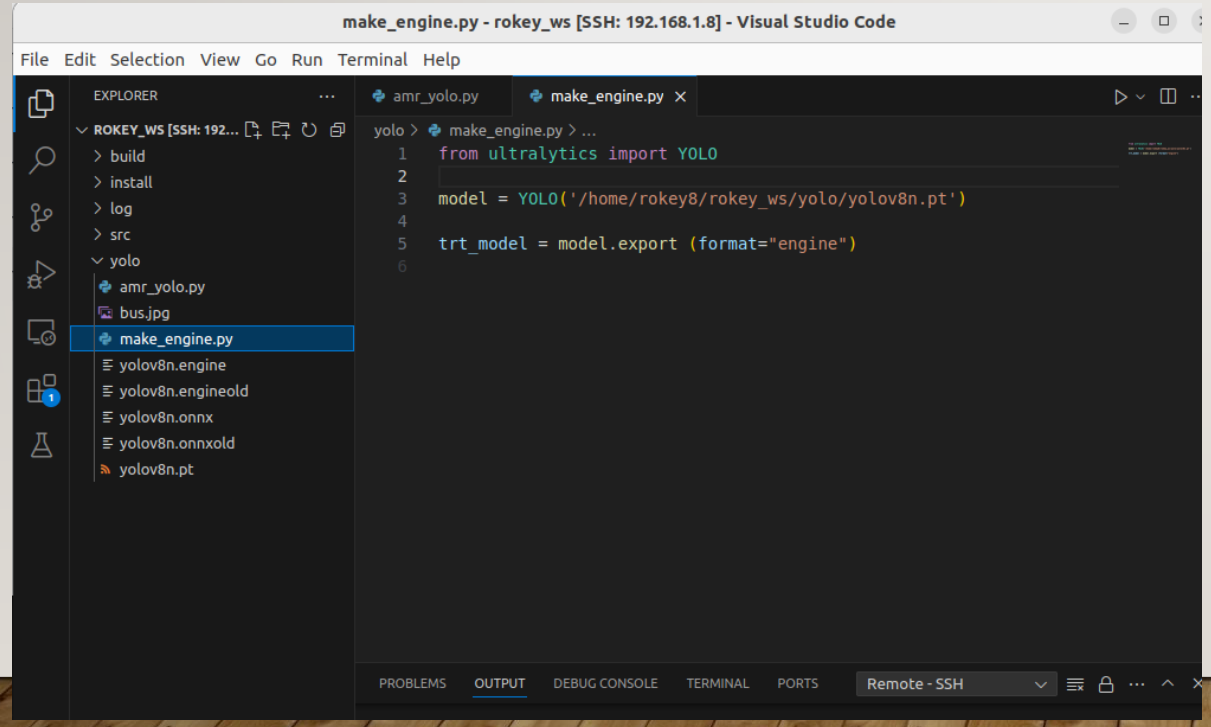
SETTING UP VSCODE FOR REMOTE EDITING

- Open a Remote Folder:
  - Once connected, VSCode will display a new window with a remote indicator in the bottom-left corner.
  - You can open any folder or file from the remote server and edit it in your local VSCode instance.

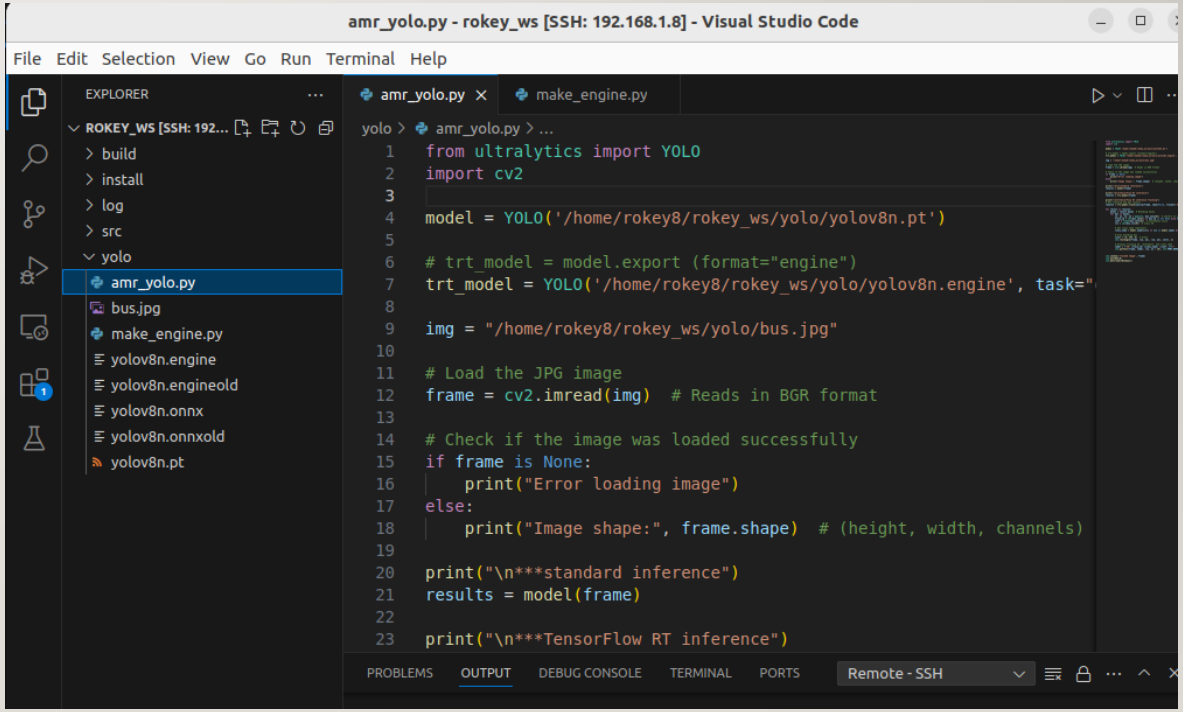
- 원격 폴더 열기:
  - 연결되면 VSCode는 왼쪽 하단 모서리에 원격 표시기가 있는 새 창을 표시합니다.
  - 원격 서버에서 모든 폴더나 파일을 열고 로컬 VSCode 인스턴스에서 편집할 수 있습니다.

CODING HINT: OPTIMIZING YOLO FOR AMR

- Make Engine
  - .PT vs. .ONNX and .ENGINE files



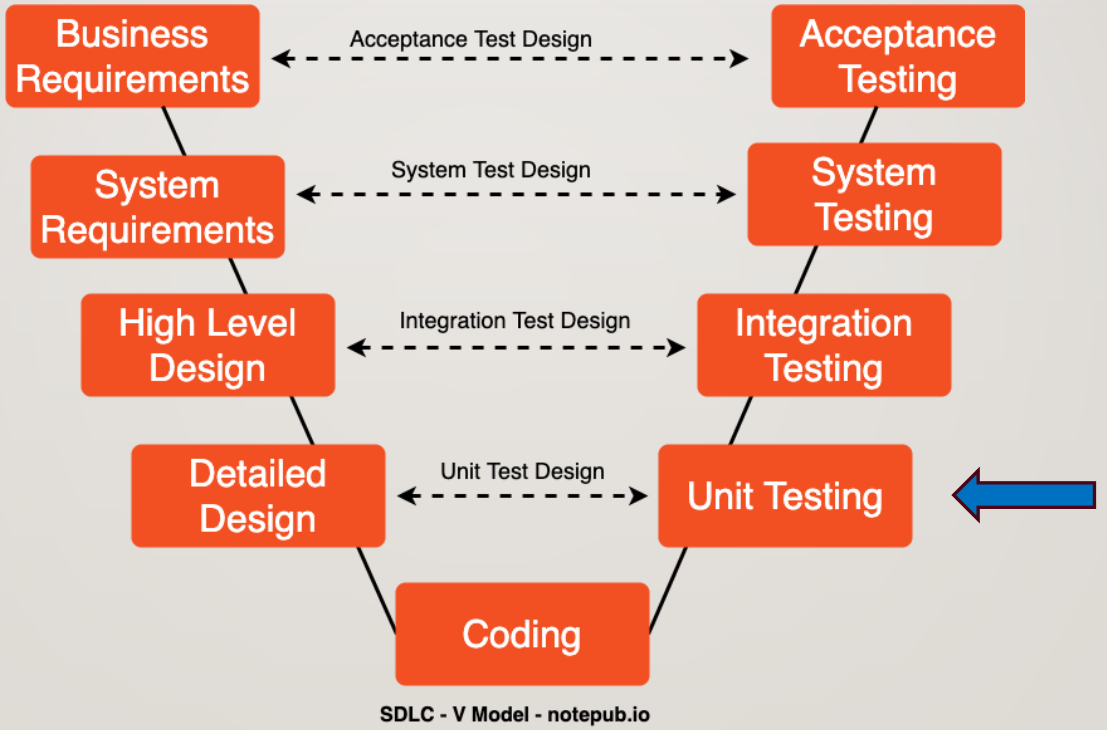
- AMR Engine Demo



TEAM EXERCISE 5

Perform coding and testing of Detection Alert Module

SPRINT 1 - DETECTION ALERT



EXPECTED OUTCOME

- Successful object detection
- ROS Nodes, and Topics created to send and display images and data

RESULTS & CODE REVIEW BY EACH TEAM

Show actual results against the expected results and explain the code written

프로젝트 RULE NUMBER ONE!!!

Are we still having FUN!