

# GOOD MORNING!

早上好!

안녕하세요!

---

DAY 1



# WELCOME TO:

[지능 - 1] Computer Vision and Automoous Driving 을 활용한 협동 로봇 구현		
차시	내용	목표
1	로봇 AI 시스템 개발 프로세스 이해	시스템 개발 프로세스의 이해 개발 환경 구축
2	프로젝트에 필요 기술 검증	AI VISION 기술 탐색 및 검증
3	프로젝트에 필요 기술 검증	로봇 AMR 제어 기술 탐색 및 검증
4	MINI 프로젝트	통합 시스템 설계 및 개발
5	파이널 프로젝트 와 프로젝트에 필요 기술 검증	웹 시스템 모니터 기술 탐색 및 검증
6-7	파이널 프로젝트	부분 시스템 설계 및 개발
8-9	파이널 프로젝트	통합 시스템 설계 및 개발
10	최종 프레젠테이션 및 시연	시스템 발표 및 시연

# Who Am I?

## Andreas (Andy) A. Kim

Born in Korea; Immigrated to US in 1976

### Education:

BS Math, CS,; MS CSE; MSM(MBA);

### Work Experience: 35+ yrs.

Recent Positions: VP/MD Asia, CTO, CSO, 고문

### Worked For:



### Mentored/Mentoring:



### Instructor:

Doosan KG KAIROS 1기; Rokey Boot Camp 1~4 기





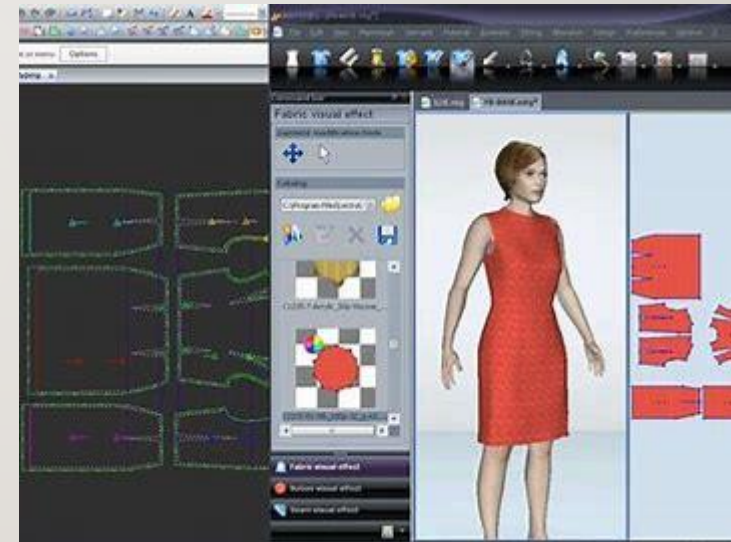
# EXAMPLE LARGE PROJECT EXPERIENCE (SW DEVELOPMENT)

---



# EXAMPLE SOLUTION PROJECT EXPERIENCE (APPLICATION)

---





# WHAT KIND OF LEADER I WANT TO BE “DEAD POET SOCIETY”

---

Andy  
Or  
“Captain”



# NOW, TELL US:

---

Who You are.

당신이 누구인지 말해 주세요

What is your objective of joining this program?

이 프로그램에 참여하는 목적이 무엇인가요?

What and how much Project and Programming experiences do you have?

어떤 프로젝트와 프로그래밍 경험을 가지고 있으며, 그 양은 어느 정도입니까?

# HOW TO WORK TOGETHER

---

- Participate, Participate, Participate!!!
- No long emails or Kakaotalk, prefer face to face
- Be open to suggestions and idea
- Be proactive (적극적), take initiative (주도적)
- HOW is as important as WHAT
- Ask the right questions? (to **YOU, team** and me)
- Investigate/Research/Analyze



# 일정 및 진행 내용

---

- 운영기간 : 10일간 진행
- 운영시간 : 9:30~18:30 (점심 13:00 ~14:00)
- 세부운영내용
  - 기본 개념부터 step by step으로 기능을 구현하여 쉽게 따라하면서 원리를 습득할 수 있도록 구성
  - 실습을 통해 직접 구현해보며 기술을 습득 - 모듈 단위로 완성된 코드 제공
  - 각 단계별 학습 내용을 통합하여 최종 프로젝트 완성

## 일정 및 진행 내용

구분	시간	내용	담당	교육방법
정규 교육	09:30 ~ 09:40	오전 출석 확인	각 담임조교	대면 교육
	09:40 ~ 11:00	프로젝트 - 1교시	교과강사 / 기술조교	
	11:10 ~ 12:30	프로젝트 - 2교시		
	12:30 ~ 13:30	점심시간		
	13:30 ~ 13:40	오후 출석 확인	각 담임조교	대면 교육
	13:40 ~ 15:00	프로젝트 - 3교시	기술조교	
	15:10 ~ 16:30	프로젝트 - 4교시		
	16:40 ~ 18:20	프로젝트 - 5교시		
	18:20 ~ 18:30	마무리 및 안내사항	각 담임조교	
비정규 교육*	18:30 ~ 19:30	저녁시간		
	19:30 ~ 21:00	자율 훈련		

# MATERIAL & ITEMS CHECK

---





# 프로젝트 주제, 계획, 및 평가

---



## 2 PROJECTS

---

- Mini Project (Individual Team)
  - For learning techniques

차시	구분	세부사항
1	프로젝트 계획 및 환경 구축	시스템 개발 프로세스의 이해, 개발 환경 구축
2	기술 탐색 및 검증	AI VISION 기술 탐색 및 검증
3	기술 탐색 및 검증	AMR 제어 기술 탐색 및 검증
4	기술 탐색 및 검증	Mini project 완성 및 발표

## 2 PROJECTS

---

- Final Project (2 Teams in One)

차시	구분	세부사항
5	프로젝트 설계	외부 시스템 모니터 기술 탐색 및 검증 파이널 프로젝트 시스템 요구사항 설계 및 프로세스 정립
6	개발	기능 구현 및 Unit Test
7	개발	기능 구현 및 Unit Test
8	개발	통합 시스템 구축 및 테스트
9	개발	통합 시스템 구축 및 테스트
10	최종 프레젠테이션 및 시연	프로젝트 발표 및 시연, 산출물 정리, 기술 컨퍼런스



# ASSUMPTION ABOUT YOUR KNOWLEDGE

---

- OS
  - Linux
- Language
  - Python3
- Packages
  - ROS2
  - OpenCV
  - Yolo8
  - Flask
  - SQLite3
- Tools
  - Labellmg
  - VSCode
  - Git/Github

# DAY I

---

- Welcome
- Project Introduction
- Introduction to Project Development Process
- Business Requirement Development
- System Requirement Development
- System and Development environment Setup

# DAY 2 (MINI PROJECT)

---

- Yolo객체 인식 모델 활용과 성능 평가 방법 이해
  - Custom Dataset과 Fine Tuning으로 자체 객체 인식 모델 구현 및 평가
  - (Optional)경량화 모델 등 개별 요구사항에 적합한 모델 탐색 및 성능 검증



# DAY 2 (MINI PROJECT)

---

## WEB-CAM 기반 객체 인식

- YOLOv8 기반 데이터 수집/학습/deploy (Detection Alert)
  - 감시용 데이터 수집(rc\_car, dummy, 등)
  - 감시용 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Detection

## AMR-CAM 기반 객체 인식

- AMR(Autonomous Mobile Robot) Turtlebot4 개발 환경 구축
- 로봇 개발 환경에 완성 모델 서빙 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출
  - Tracking 데이터 수집((rc\_car, dummy, 등)
  - Tracking 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object **Tracking**

# DAY 3 (MINI PROJECT)

---

- Auto. Driving 시스템 학습
  - Digital Mapping of environment
  - Operate AMR (Sim. & Real)
  - Tutorial 실행
  - Detection, Depth and AMR 주행
  - 로봇 개발 환경에 적용 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출

## TURTLEBOT4 시뮬레이션 DEMO

- SLAM과 AutoSLAM으로 맵 생성
- Sim.Tutorial 실행
- Detection, Depth and AMR 주행 example

# DAY 3 (MINI PROJECT)

---

## REAL ROBOT

- Manually operating the AMR (Teleops)
- autonomous driving 시스템 with obstacle avoidance
  - Digital Mapping of environment
  - Launching Localization, Nav2, and using Rviz to operate a robot
  - Goal Setting and Obstacle Avoidance using Navigation

## TUTORIAL

- Turtlebot4 API를 활용한 Initial Pose Navigate\_to Pose 구현
- Turtlebot4 API를 활용한 Navigate\_Through\_pose, Follow Waypoints 구현



# DAY 4 (MINI PROJECT)

---

- System(High Level) Design (Mini Project)
  - System Architectural Diagram
- Detail Design to Acceptance - Agile Development (SPRINTs)
  - Detection
  - AMR Control

# DAY 4 (MINI PROJECT)

---

## CODING, TEST & INTEGRATION

- Coding and Test all modules
- Porting to ROS
- And finally, Integration and Test of Detection Alert & AMR Controller

## MINI PROJECT DEMO

- Prepare and demo completed project

# DAY 5 (FINAL PROJECT)

---

- Flask 를 이용한 웹 서버 구축 (System Monitor)
  - Flask/HTML Intro
  - Deploy YOLOv8 Obj. Det results to web
  - Log in 기능 구현
  - Sysmon 웹기능 구현
- SQLite3를 이용한 데이터베이스 구축 및 연동 (System Monitor)
  - SQLite3 기본 기능 구현
  - DB 기능 구축
  - 저장된 내용 검색하는 기능 구현

# DAY 5 (FINAL PROJECT)

---

- 비즈니스/System 요구 사항 업데이트
- 시스템 설계 및 프로세스 정립
  - System(High Level) Design
  - Schedule/Time Management
- 역할 분담 및 일정 조율
- 개발 환경 구축(맵 디자인, SW 개발, 문서 통합 관리)
- 멀티 로봇 환경 구축 및 네비게이션
- 멀티 로봇 개별 업무 수행
- 멀티 로봇 협동 업무 수행
- (Optional) Turtlebot4 각종 센서 데이터의 이해와 적용



# DAY 6 (FINAL PROJECT)

---

- 시스템 설계에 기반한 객체 감지 모델 구현
  - 로봇 환경에 적용 및 Unit Test
  - 모듈로 제작하고 launch파일로 구현
  - code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑
- 시스템 설계에 기반한 SysMon 설계 구현
  - 로봇 환경에 적용 및 Unit Test
  - 모듈로 제작하고 launch파일로 구현
  - code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑

# DAY 7 (FINAL PROJECT)

---

- 시스템 설계에 기반한 **AMR 제어** 구현
- 로봇 환경에 적용 및 Unit Test
- 모듈로 제작하고 launch 파일로 구현
- code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑

# DAY 8-9 (FINAL PROJECT)

---

- 개별 기능 통합 구현 및 Integration 테스트
- 통합 Launch 파일로 구현
- Robust한 시스템 구축을 위한 예외 처리 및 Code Refactoring
- code 정리 및 버전관리, 문서 작성 및 영상 촬영, 팀 내 기술 브리핑

# DAY 10 (FINAL PROJECT)

---

- 프로젝트 발표 및 시연
- 최종 산출문 정리(소스코드, 발표 PPT, 동작 영상)
- 팀 간 기술 컨퍼런스를 통한 기술 극복 경험담, 노하우 교류(채점 대상X)



# PROJECT 평가

## 6. 평가 방법(★★★ 매우중요)

### ① 평가 구성(실무 프로젝트 교육)

평가 항목	평가 방법	평가 시기	평가 횟수	평가 주체
기술 역량 평가	<ul style="list-style-type: none"><li>· 별도 제공된 양식에 따라 각 그룹별 평가</li><li>· 조 단위 평가(개인 평가X)</li><li>· 교강사가 직접 평가표 작성 후 운영팀에 제출</li></ul>	각 그룹별 수업 종료 후	6회	교강사
조직 역량 평가	<ul style="list-style-type: none"><li>· 별도 구글 설문 시트를 통한 평가</li><li>· 동료/개인 평가</li></ul>	실무 프로젝트 교육 전체 종료 후	1회	교육생

※ 『조직 역량 평가』 운영은 "운영팀"에서 별도 운영 합니다.

# PROJECT 평가

## ▶ 평가 항목

평가 항목	평가기준
기능 구현 완전성	* 교육에서 요구하는 기능이 모두 구현되어 있는지 여부를 평가한다.
기능 구현 정확성	* 구현된 모든 기능들이 정상적으로 동작하는지 여부를 평가한다.
동작 및 운용 안정성	* 산출물 운용시 안정적으로 동작하는지 여부를 평가한다.
입출력 데이터 이해도	* 데이터 입출력 방법 및 절차가 편리하고 기능 요구 내용에 적합한 지 여부를 평가한다.
기능 동작 지속성	* 장애/오류 발생 시에도 지속적인 동작/운영이 가능한지 여부를 평가한다.

# PROJECT 평가

Mini Project	
1. 비즈니스 요구 사항 작성	10
2. Detection Alert Module의 코딩 및 테스트 수행	10
3. AMR Controller Module의 코딩 및 테스트 수행	10
4. System Monitor Module 코딩 및 테스트 수행	10
5. Process Flow Diagram을 사용하여 시스템 설계를 생성	30
6. System Design Doc 완성도	10
7. 모든 모듈의 시스템 통합 및 테스트 수행	50
ap 상 장애물 회피, undock dock init 자동화 goal arrival	10
Detection: Object detection, depth 변환, 이동	10
Realtime 장애/오류 Handling Function	10
Quality of System Monitor Completed	10
Level of System Integrated	20
8. 최종 프로젝트 발표	10
Final Project Sub Total	150

# 프로젝트 RULE

---

80/20 → 20/80



# TEAMWORK AND PROJECT MANAGEMENT

---



프로젝트 RULE NUMBER ONE!!!

---

Have Fun Fun Fun!





Let's Get Started!!!!

# NOTION FOR STUDENTS

---



## Doosan Rokey 5

강의 장소	오프라인
강의 시간	9:30 ~ 18:30 (8시간)
점심 시간	13:00~14:00
대상	두산로보틱스 5기
사전 학습	Linux, Python, ROS2, YOLO, OpenCV
강사	Andy Kim

깃허브:

 [GitHub - kimandreas/to\\_students](#)

구글미트:

 [Meet](#)

산출물 구글드라이브

 [Google Drive Doosan\\_5기](#) - [Google Drive](#)

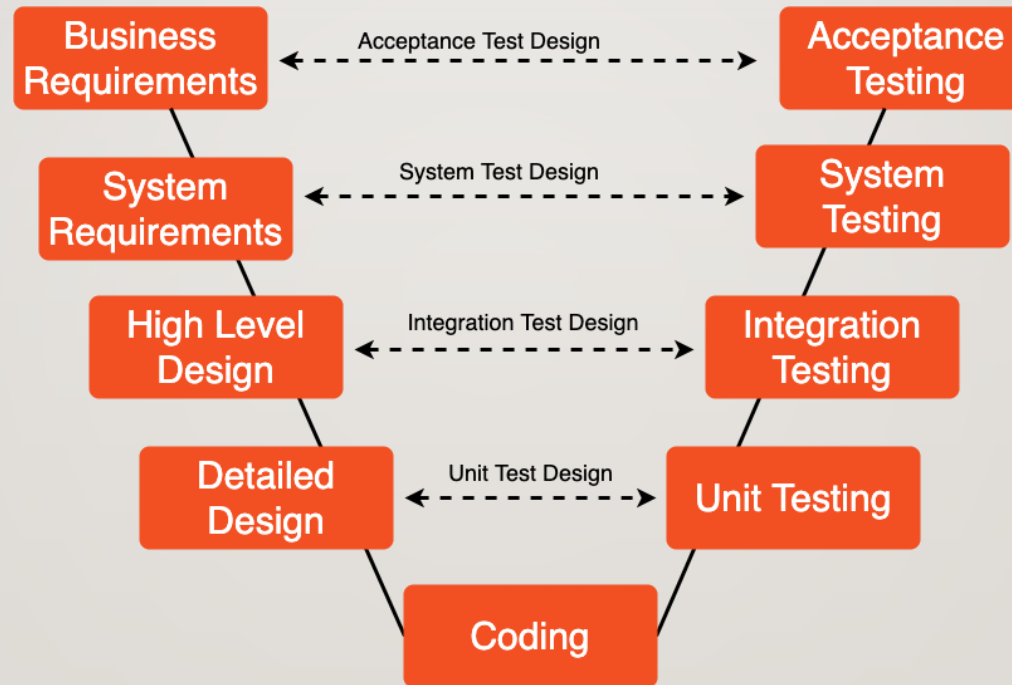


PROJECT DEVELOPMENT IS A PROCESS

---

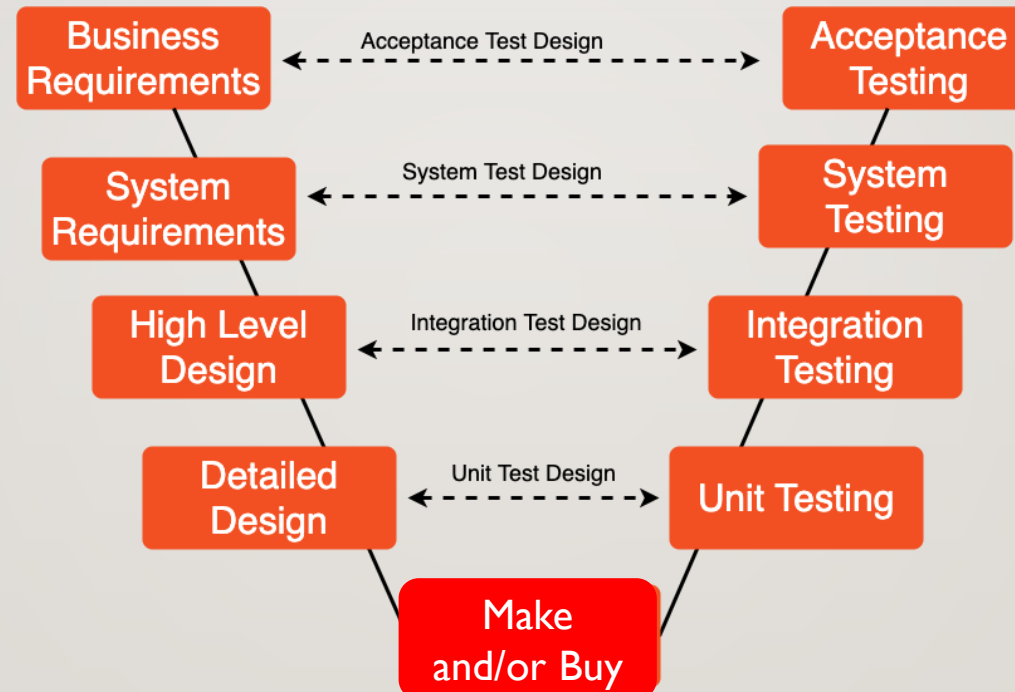


# SW DEVELOPMENT PROCESS



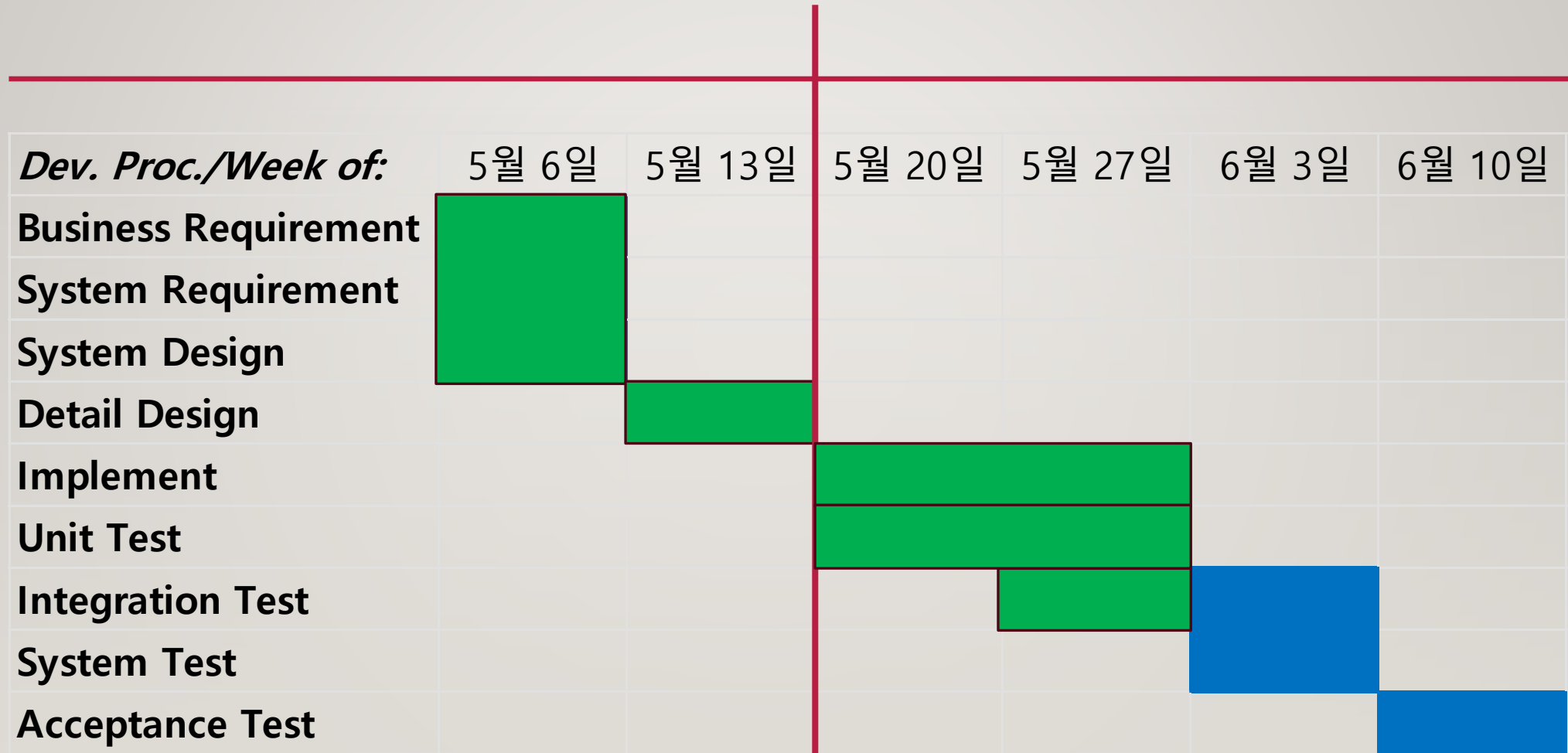
SDLC - V Model - notepub.io

# HW DEVELOPMENT PROCESS



SDLC - V Model - notepub.io

# EXAMPLE TIMELINE



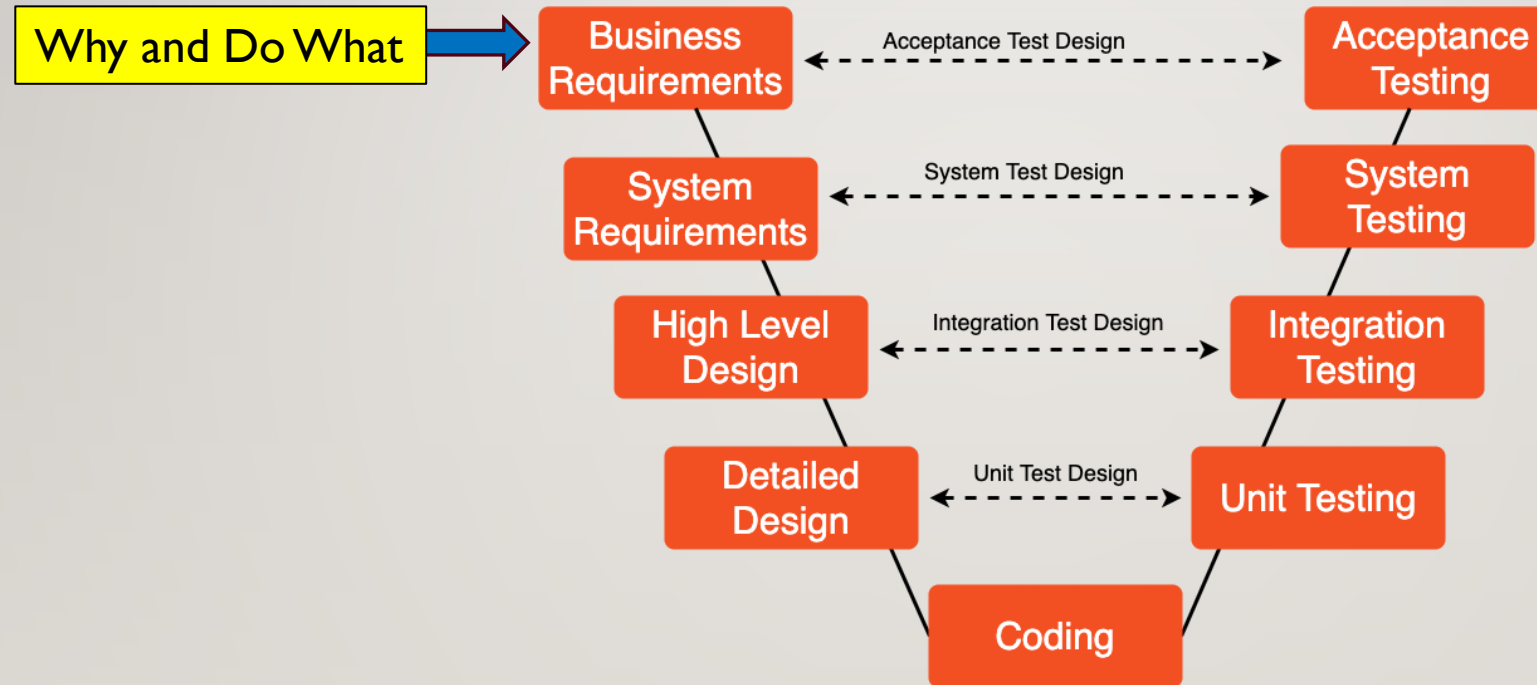


# PROJECT DEVELOPMENT PROCESS

---



# SW DEVELOPMENT PROCESS



SDLC - V Model - notepub.io

# PROJECT/SOLUTION JUSTIFICATION (WHY)

---

- Situation Analysis
  - evaluates both external and internal factors to determine the necessity and feasibility of a project. It helps justify resource allocation by outlining how the project aligns with strategic goals, identifying potential challenges and opportunities, and providing a detailed understanding of the project's context for informed decision-making.
- 상황 분석
  - 프로젝트의 필요성과 타당성을 결정하기 위해 외부 및 내부 요인을 모두 평가합니다. 프로젝트가 전략적 목표에 어떻게 부합하는지 설명하고, 잠재적인 과제와 기회를 식별하고, 정보에 입각한 의사 결정을 위해 프로젝트의 컨텍스트에 대한 자세한 이해를 제공하여 리소스 할당을 정당화하는 데 도움이 됩니다.

# PROJECT/SOLUTION JUSTIFICATION (WHY)

---

- Business Needs/Pain Point Analysis

- identifies and assesses the problems and unmet needs of customers. This process helps businesses tailor their solutions to enhance customer satisfaction and loyalty by directly addressing these issues.

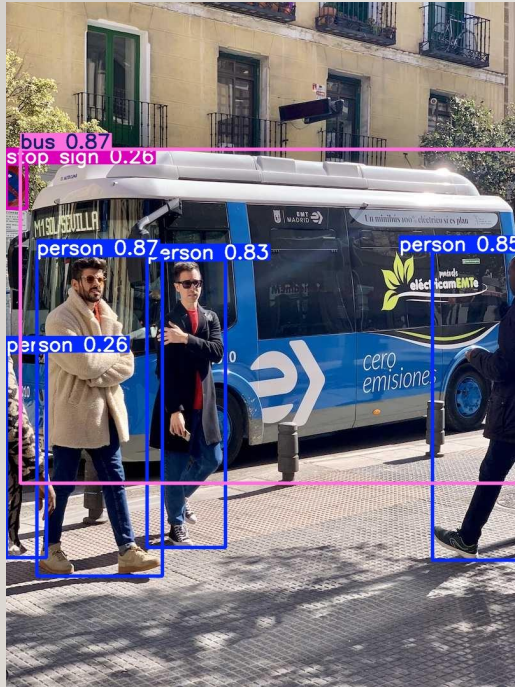
- 비즈니스 니즈/문제점 분석

- 문제와 충족되지 않은 요구를 식별하고 평가합니다. 이 프로세스는 기업이 이러한 문제를 직접 해결하여 고객 만족도와 충성도를 높일 수 있도록 솔루션을 맞춤화하는 데 도움이 됩니다.



# ADVANCED TECHNIQUES THAT WE HAVE

- AI Object Detection

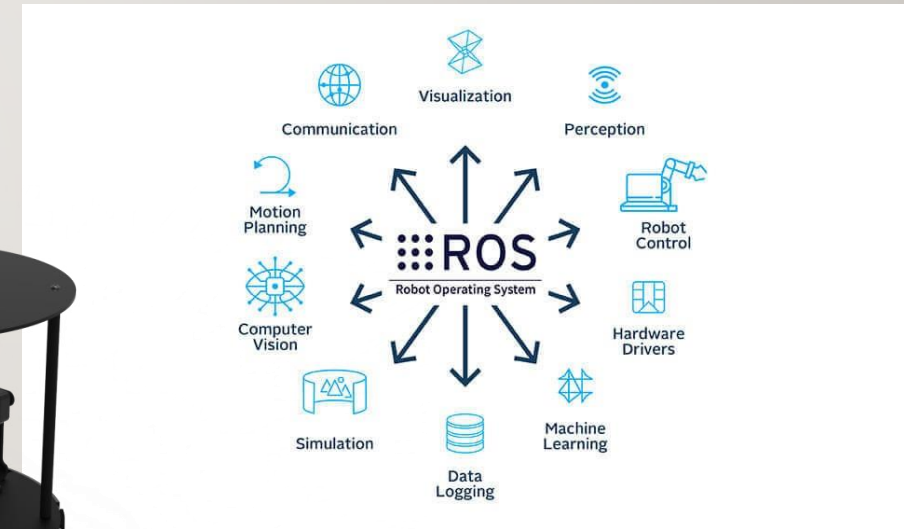


- AMR

- Navigation with obstruction avoidance
- Sensors



- ROS2



# BRAINSTORM A SITUATION THAT WILL BENEFIT FROM **YOUR** SOLUTION

---

Must have measurable benefits. Search for them online



# 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





PRESENT THE **SITUATION** AND  
**MEASURABLE BENEFITS** OF **YOUR**  
SOLUTION TO YOUR TEAM

---





# BUSINESS/SOLUTION REQUIREMENT (WHAT EXAMPLE)

---

- **Business Requirements with Metrics:** The company aims to deploy a robotic system integrated with a deep learning model to automate quality inspection in manufacturing. The goal is to reduce human error by achieving 98% accuracy in defect detection and increase production efficiency by minimizing inspection time to under 2 seconds per item.
- 이 회사는 딥 러닝 모델과 통합된 로봇 시스템을 배포하여 제조 시 품질 검사를 자동화하는 것을 목표로 합니다. 목표는 결함 감지에서 **98%의 정확도**를 달성하여 인적 오류를 줄이고 검사 시간을 품목당 2초 미만으로 최소화하여 생산 효율성을 높이는 것입니다.

# EXAMPLE BUSINESS/SOLUTION REQUIREMENT DOCUMENT

Business Requirements Document (BRD)

Project Title: Autonomous Mobile Robot (AMR) Security System

Project Owner: Kim Andreas  
Date: [Insert Date]  
Version: 1.0

1. Business Objectives

Implement an AI-powered robotic security solution using Autonomous Mobile Robots (AMRs) to monitor and safeguard a secure area, reducing reliance on human security personnel, increasing operational efficiency, and minimizing security risks.

2. Project Scope

Develop and deploy an AMR-based security system to perform real-time surveillance, threat detection, and autonomous response in restricted or

프로젝트 제목: 자율 이동 로봇(AMR) 보안 시스템

↓  
프로젝트 소유자: 김 안드레아스  
날짜: [날짜 삽입]  
버전: 1.0

1. 비즈니스 목표

인공지능 기반 자율 이동 로봇(AMR)을 활용하여 보안 지역을 감시하고 보호하는 로봇 보안 솔루션을 구현하여, 인력 보안 의존도를 줄이고 운영 효율성을 높이며 보안 위험을 최소화합니다.

2. 프로젝트 범위

제한된 또는 민감한 지역에서 실시간 감시, 위협 탐지 및 자율 대응을 수행하는 AMR 기반 보안 시스템을 개발하고 배포합니다. 주요 기능에는 순찰, 이상 탐지, 위협 대응, 경고 상승이 포함됩니다.

포함 범위:

(on [https://github.com/kimandreas/to\\_students](https://github.com/kimandreas/to_students))

# TEAM EXERCISE I

---

Brainstorm Business/Solution Requirement for **Your** Solution and write business requirement statement

Using the posted notes and flipchart as needed

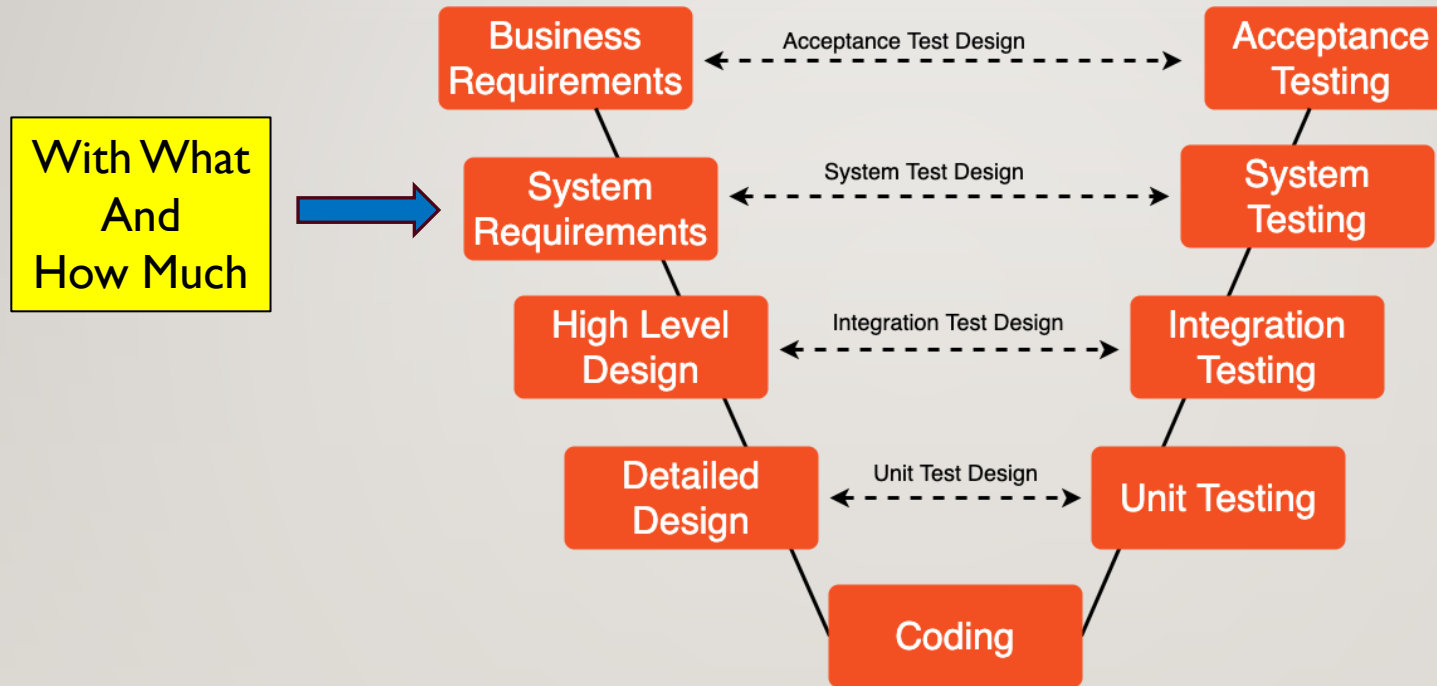
# BUSINESS/SOLUTION REQUIREMENT PRESENTATION (SELECTED)

---

Using the posted notes and flipchart as needed



# SW DEVELOPMENT PROCESS



SDLC - V Model - notepub.io

# SYSTEM/TECHNICAL/SOLUTION REQUIREMENT (EXAMPLE)

---

## Technical Requirements and Metrics:

- 1. Deep Learning Models:** Train a CNN to achieve at least 98% accuracy on defect detection in validation datasets.

검증 데이터 세트에서 결함 감지에 대해 최소 98%의 정확도를 달성하도록 CNN을 훈련시킵니다.

- 2. Robotics Hardware:** Ensure the robot processes images and delivers results within 2 seconds per item, with 99.9% system uptime.

로봇이 99.9%의 시스템 가동 시간으로 항목당 2초 이내에 이미지를 처리하고 결과를 제공하도록 보장합니다.



# SYSTEM/TECHNICAL/SOLUTION REQUIREMENT (EXAMPLE)

---

## Technical Requirements and Metrics:

- 3. Interface and Control System:** Design for less than 0.1% downtime and a response time under 1 second for user interactions.

다운타임이 0.1% 미만이고 사용자 상호 작용에 대한 응답 시간이 1초 미만으로 설계됩니다.

# EXAMPLE SYSTEM REQUIREMENT DOCUMENT

## System Requirements Document (SRD)↵

↓

**Project Title:** Autonomous Mobile Robot (AMR) Security System↓

**Version:** 1.0↓

**Date:** [Insert Date]↵

### 1. Introduction↵

The system requirements define the technical specifications for developing and implementing the AMR-based security solution. This includes hardware, software, networking, and integration requirements.↵

### 2. System Overview↵

This system will provide autonomous patrolling, threat detection, and reporting for secure areas using AI-enabled Autonomous Mobile Robots (AMRs). It integrates navigation, sensor data processing, real-time alerts, and user interface management.↵

## 시스템 요구사항 문서 (SRD)↵

**프로젝트 제목:** 자율 이동 로봇(AMR) 보안 시스템↓

**버전:** 1.0↓

**날짜:** [날짜 삽입]↵

### 1. 소개↵

이 시스템 요구사항 문서는 AMR 기반 보안 솔루션의 개발 및 구현을 위한 기술 사양을 정의합니다. 여기에는 하드웨어, 소프트웨어, 네트워킹, 통합 요구사항이 포함됩니다.↵

### 2. 시스템 개요↵

이 시스템은 AI 기반 자율 이동 로봇(AMR)을 사용하여 보안 구역의 자율 순찰, 위험 탐지 및 보고를 제공합니다. 네비게이션, 센서 데이터 처리, 실시간 경고 및 사용자 인터페이스 관리를 통합합니다.↵

(on [https://github.com/kimandreas/to\\_students](https://github.com/kimandreas/to_students))



# TEAM EXERCISE 2-1

---

Brainstorm System/Solution Requirement for **Your** solution and document

Using the posted notes and flipchart as needed

# SYSTEM REQUIREMENT PRESENTATION (SELECTED)

---

Using the posted notes and flipchart as needed

# MINI PROJECT

---



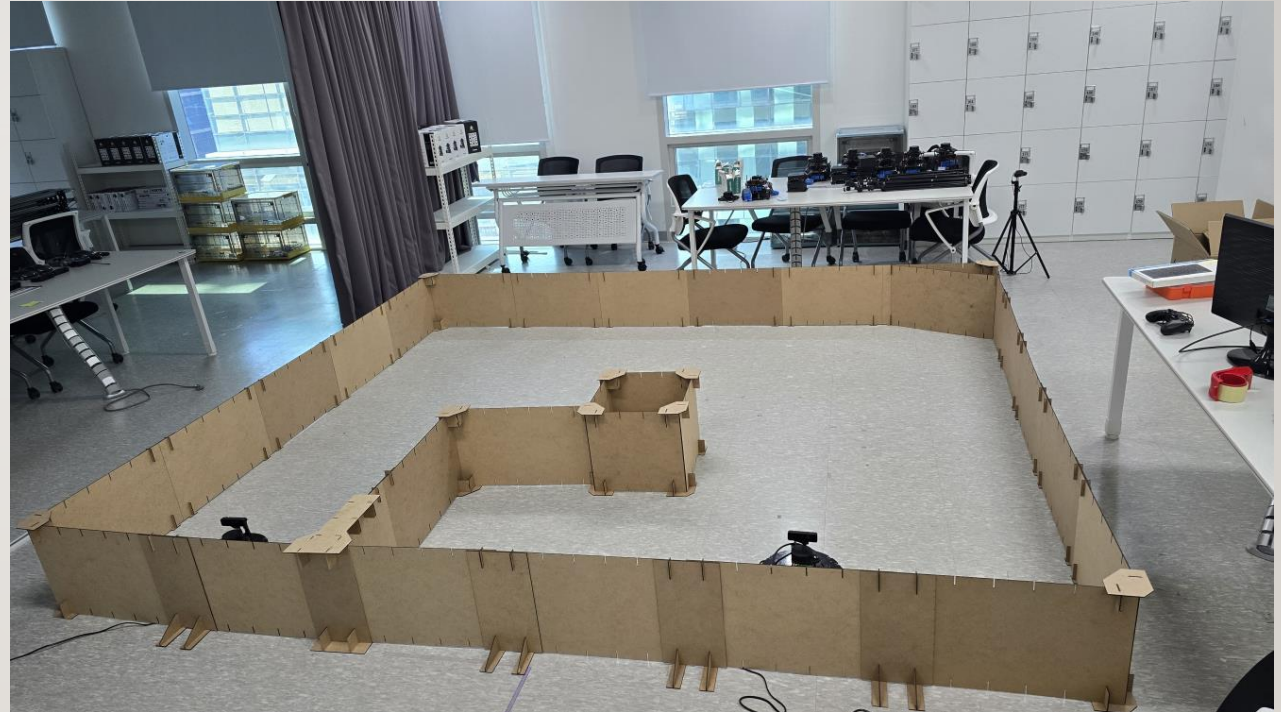
# HOW MUCH (USE CASE)

---



## MINI PROJECT DESCRIPTION

---



DETECTION ALERT

---





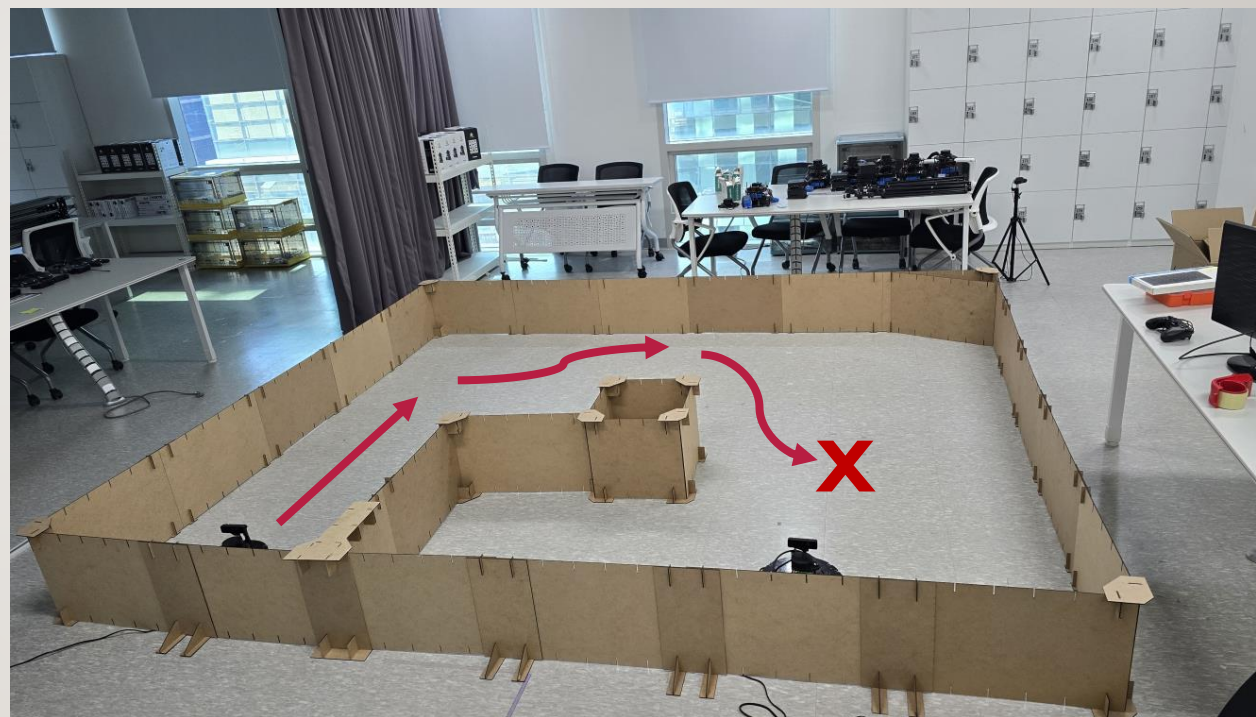
START

---



## NAVIGATE TO A POSITION

---





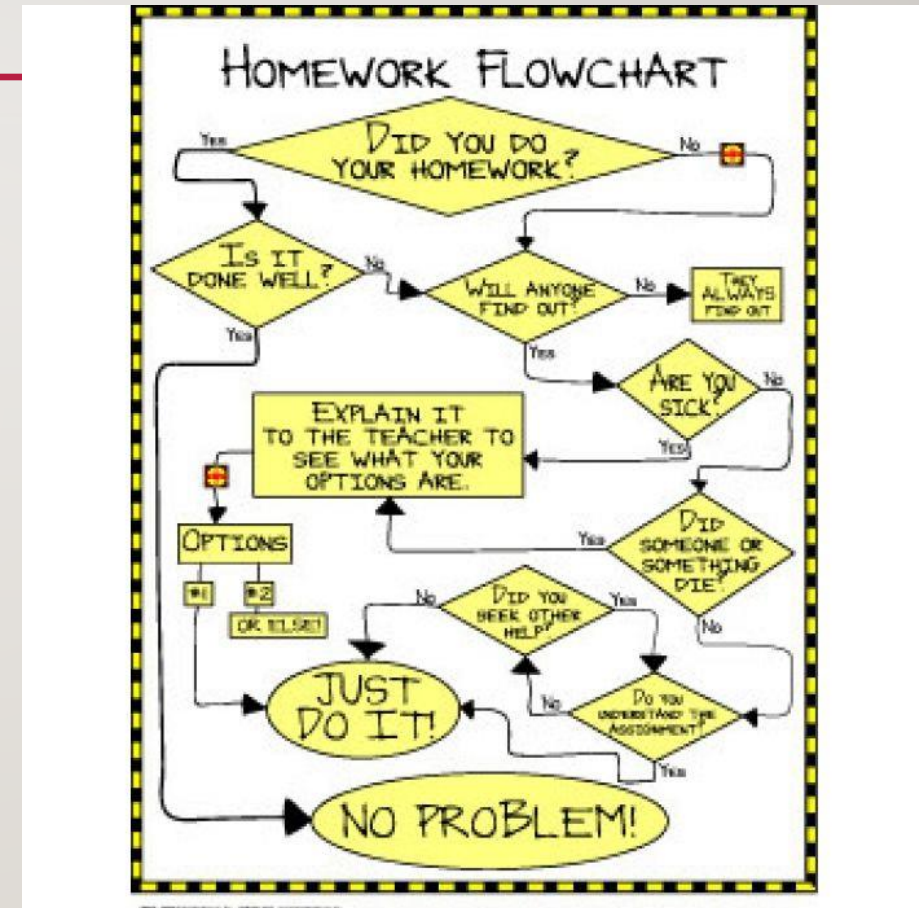
## TRACK & APPROACH

---



# VISUALIZATION – SCENARIO PROCESS DIAGRAMS

- As-Is Functional Process Diagram
  - Current states
- To-Be Functional Process Diagram
  - Future states
- [Untitled Diagram - draw.io](#)
- <https://app.diagrams.net/>



Process

Decision

Document

Data

Sub  
Process

Start

Data

Card

Manual  
Operation

Preparation

Loop Limit

\_\_\_\_\_

\_\_\_\_\_

paper Tape

Terminator

Display

Sequential  
Data



# DEVELOP MINI PROJECT SCENARIO (USE-CASE) PROCESS DIAGRAM

---

Using the posted notes and flipchart as needed



WITH WHAT

---



# YOUR PROJECT ENVIRONMENT

---



# BASE HW/OS

---

- PC

- Ubuntu 22.04
- USB Camera



- Network
  - Wifi



- AMR

- TurtleBot4
- Ubuntu 22.04





# TURTLEBOT4 ROBOT 만나기

---

## 강의 보조 문서

### DAY 1 - Setup/Development Process

Aa 이름

☼ 상태

● 평가 지표

● 완료

● Turtlebot4 Robot 만나기

● 완료

More information can be found on:

- [User Manual · Turtlebot4 User Manual](#)
- <https://turtlebot.github.io/turtlebot4-user-manual/>



# AMR (DEMO)

---

- [User Manual · Turtlebot4 User Manual](#)
- <https://turtlebot.github.io/turtlebot4-user-manual/>
- Teleop with keyboard
- Navigation with SLAM



# OBJ. DET.

---

TARGET

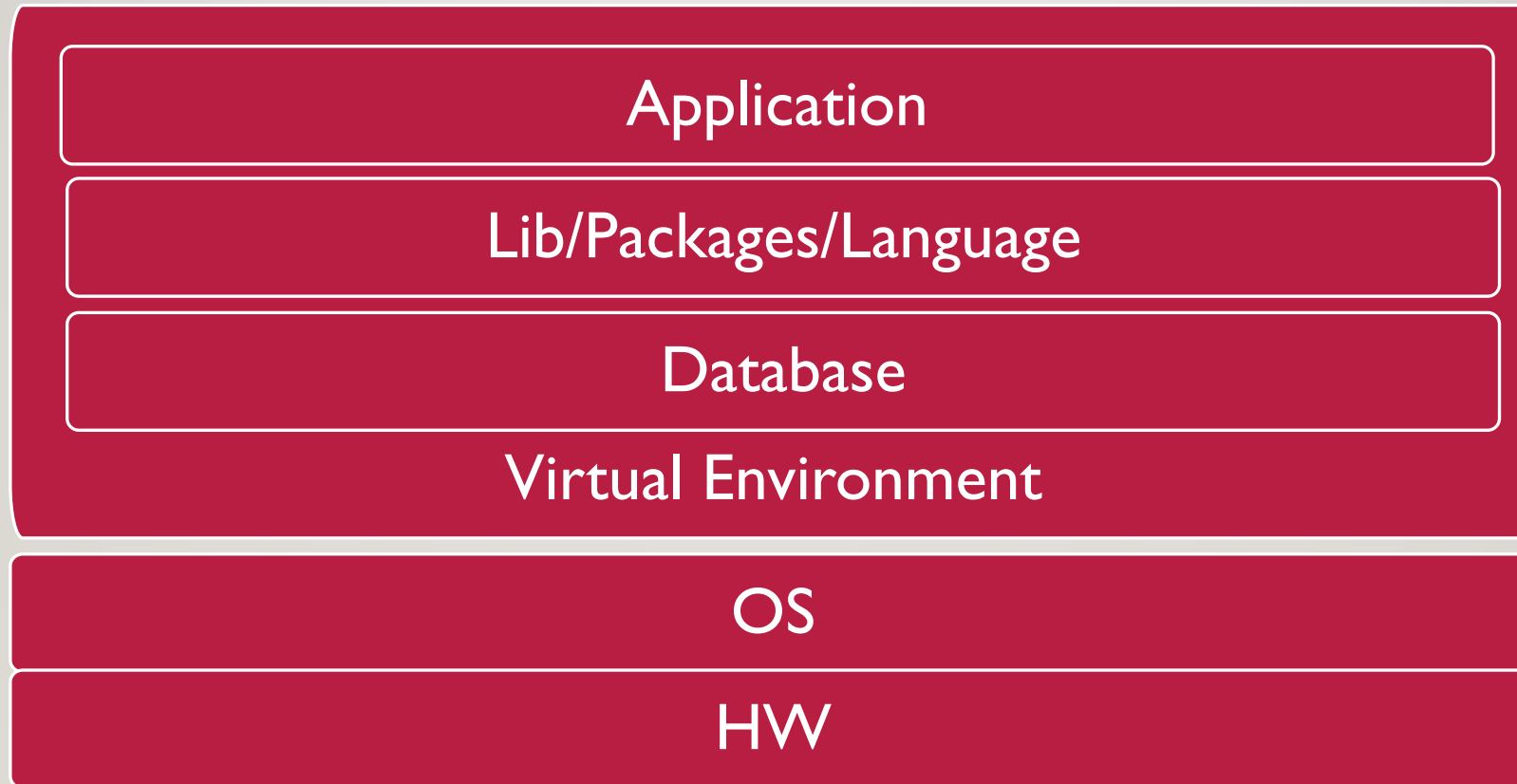


DUMMY



# EXAMPLE SYSTEM STACK

---



# BASE SYSTEM PACKAGES

---

## PC

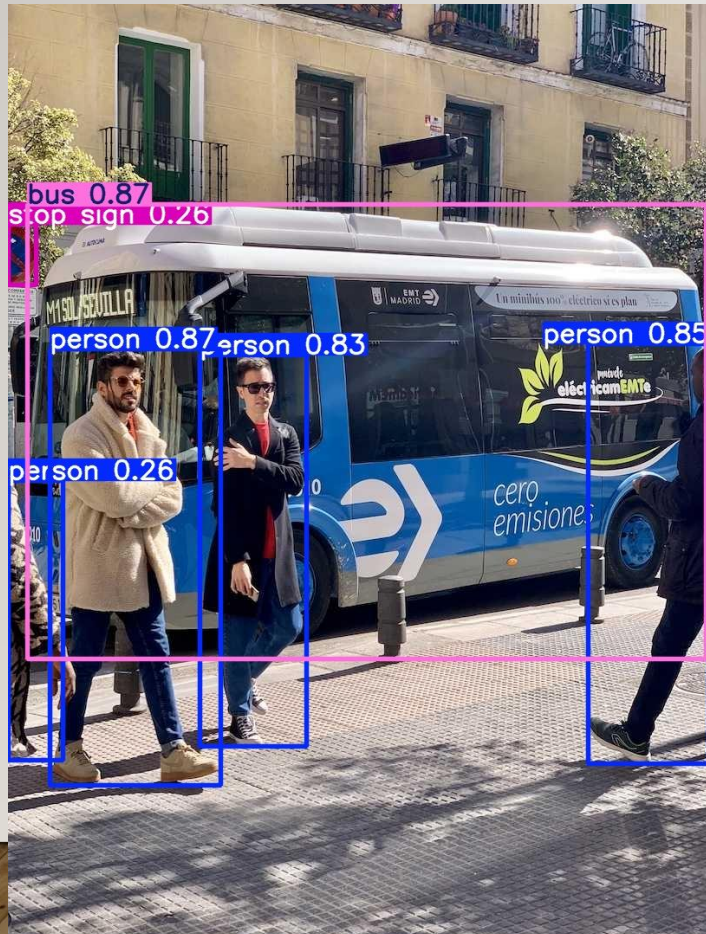
- Python3
- ROS2
- Opencv
- Ultralytics
- Flask
- SQLite3

## AMR

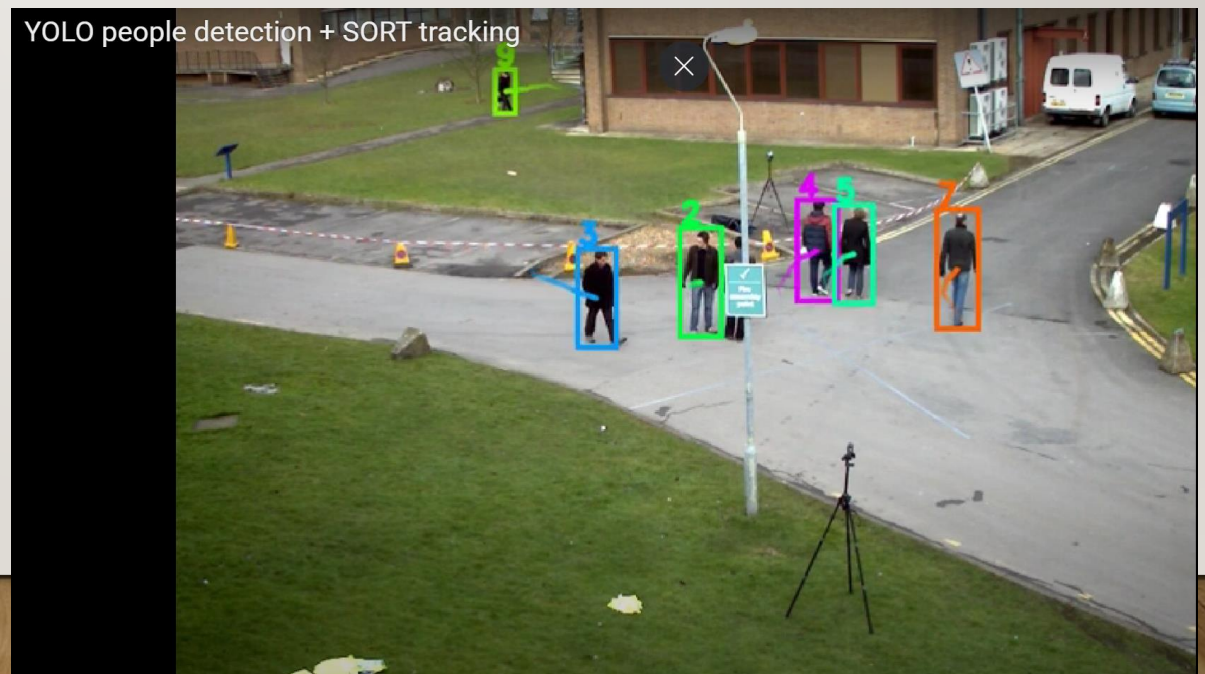
- Python3
- ROS2



# YOLO OBJ. DET. VS. YOLO TRACKING



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



# 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

# TEAM EXERCISE 2-2

---

Brainstorm **mini-project** System Requirement and document

Using the posted notes and flipchart as needed

Include where, when, what will be used



# SYSTEM REQUIREMENT PRESENTATION BY EACH TEAM (SELECTED)

---

Using the posted notes and flipchart as needed

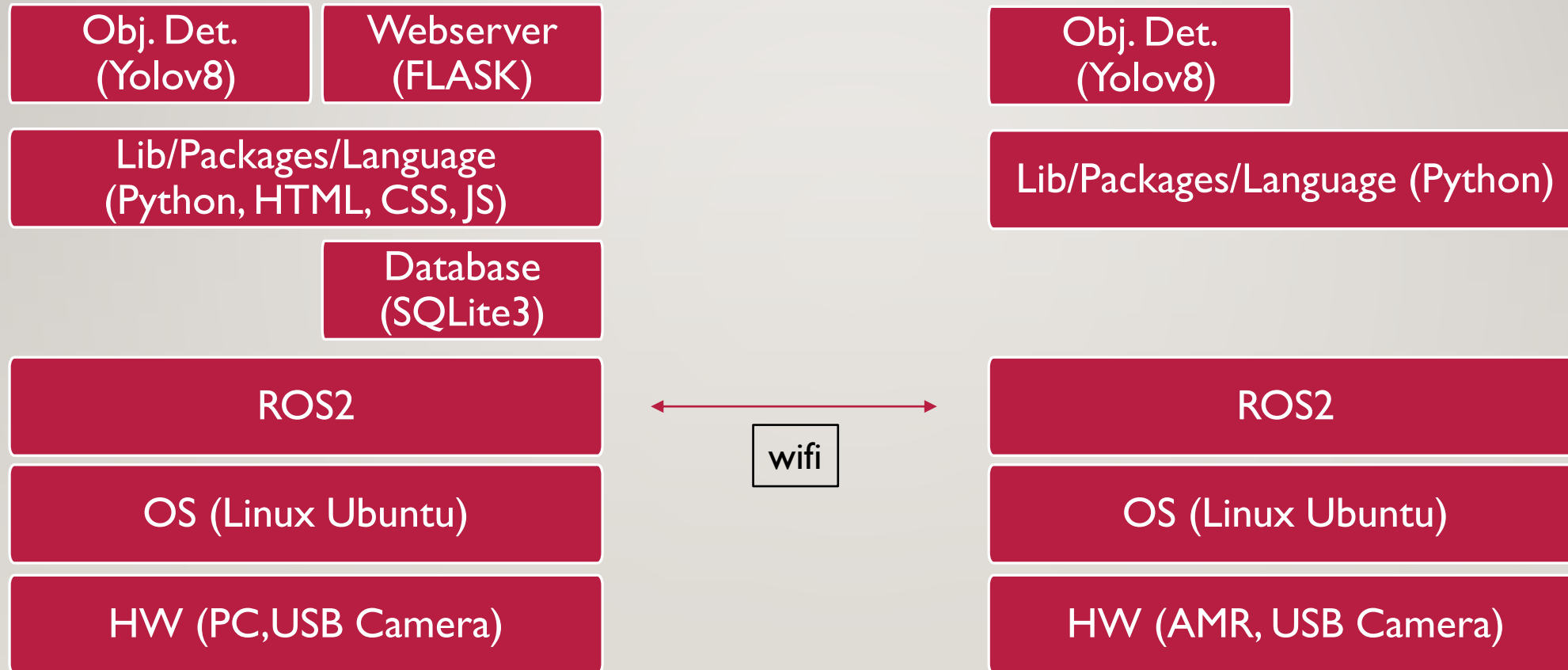


# SYSTEM AND DEVELOPMENT ENVIRONMENT SETUP

---



# PROJECT SW STACK



# PROJECT SW SETUP

---

● PC 환경 구성	● 완료
● Turtlebot4 SW 구성	● 완료
● User PC Single Robot Network Setup	● 완료
● PC .bashrc 구성 Example	● 완료

# OPERATING ROBOT (MANUALLY)

---

● Move Robot CLI	● 진행 중
● Turtlebot4 SSH Access	● 진행 중
● YOLO Setup	● 진행 중
● ROS Workspace Example	● 진행 중
● Homework_삐뽕삐뽕 소리내기 노드 만들기	● 진행 중



# CONNECTING TO ROBOT

---

● Move Robot CLI	● 진행 중
● Turtlebot4 SSH Access	● 진행 중
● YOLO Setup	● 진행 중
● ROS Workspace Example	● 진행 중
● Homework_삐뽀삐뽀 소리내기 노드 만들기	● 진행 중


# SETTING UP YOLO

---

● Move Robot CLI	● 진행 중
● Turtlebot4 SSH Access	● 진행 중
● YOLO Setup	● 진행 중
● ROS Workspace Example	● 진행 중
● Homework_삐뽀삐뽀 소리내기 노드 만들기	● 진행 중

# ROS WORKSPACE

---

● Move Robot CLI		● 진행 중
● Turtlebot4 SSH Access	 OPEN	● 진행 중
● YOLO Setup		● 진행 중
● ROS Workspace Example		● 진행 중
● Homework_삐뽀삐뽀 소리내기 노드 만들기		● 진행 중

# HOMEWORK

---





# REVIEW AMR (TURTLEBOT4) E-MANUAL

---

- [Features · User Manual](#)
- <https://turtlebot.github.io/turtlebot4-user-manual/overview/features.html>



# PLEASE REVIEW YOUR WORK FROM EARLIER ONLINE CLASS

---

- Yolo obj. Det. Vs. Yolo Tracking
  - [Object Detection - Ultralytics YOLO Docs](#)
  - [Track - Ultralytics YOLO Docs](#)
  - [Model Training with Ultralytics YOLO - Ultralytics YOLO Docs](#)
- Yolo
  - Data Labelling (ex: Labellmg/roboflow)
  - Data pre-processing for YoloV8 Training
  - YoloV8 training to create .pt file
  - Using .pt file to predict/inference
- ROS
  - colcon build
  - Node, Topic, Service, Action, Interface, etc. coding

# ROS EXERCISE I

---

Create a ROS2 Package with these publisher and subscribers

```
2_0_a_image_publisher.py
2_0_b_image_subscriber.py
2_0_c_data_publisher.py
2_0_d_data_subscriber.py
```

Try these CLI

```
$ 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 <topic_name>
$ ros2 interface list
$ ros2 interface show
  <package_name>/msg/<MessageName>
$ rqt --clear-config
```

# ROS EXERCISE 2

---

● YOLO Setup

● 완료

● Homework\_삐뽀삐뽀 소리내기 노드 만들기

● 완료



# BEFORE GOING HOME (VERY IMPORTANT)!!!

---

Remember to:

- Dock your robot
- Perform **soft** shutdown by
  - PC Terminal  
`$ ssh ubuntu@<AMR IP>`
  - AMR Terminal  
`$ sudo shutdown now`

Or

- Perform **soft** shutdown by
  - PC Terminal  
`$ ssh ubuntu@<AMR IP>`
  - AMR Terminal  
`$ sudo shutdown now`
- Undock your robot
- Power off Create 3



# 프로젝트 RULE NUMBER ONE!!!

---

Are we having  
Fun???

