

GOOD MORNING!

早上好!

안녕하세요!

PROJECT INTRODUCTION



DAY I (DONE)

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

DAY 2 (DONE)

- YOLOv8 기반 데이터 수집/학습/deploy (Security Alert)
 - 감시용 데이터 수집(bus, truck, tank 등)
 - 감시용 데이터 라벨링
 - YOLOv8 기반 학습
 - YOLOv8 Object Detection
- Porting to ROS
 - Create Security 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 (?)

- 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
 - 알람 기능 구현
- Porting to ROS
 - Create Sysmon Node
 - Receive Image/Data Topic from Security Alert Node and display on the SysMon webpage

DAY 3 (?)

- SQLite3를 이용한 데이터베이스 구축 및 연동 (System Monitor)
 - SQLite3 기본 기능 구현
 - DB 기능 구축
 - 알람이 울리는 경우 DB에 저장하는 기능 구현
 - 저장된 내용 검색하는 기능 구현
- Porting to ROS
 - Update Sysmon Node code
 - Update the database with received Obj. Det. Data from Security Alert Node
 - Display the content of DB on Security Monitor web page

DAY 4

- 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/telops
- Porting to ROS
 - Create AMR Controller Node
 - Create and send Obj.Tracking Image and data to Sysmon

DAY 5

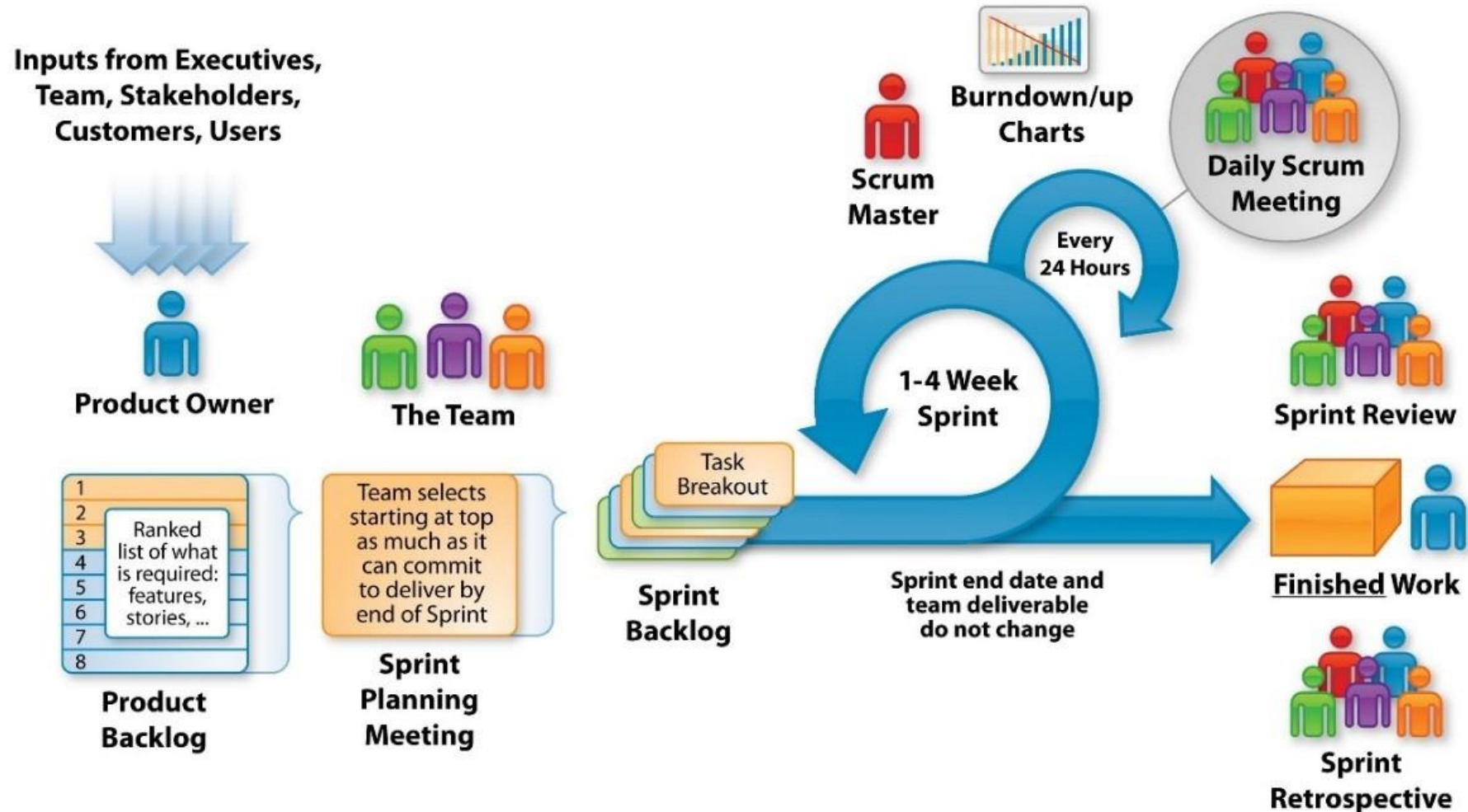
- 감시시스템 통합 구현
 - - 전체 시스템 통합 운용
- Team Demo & Presentation
- 평가 시간

프로젝트 RULE NUMBER ONE!!!

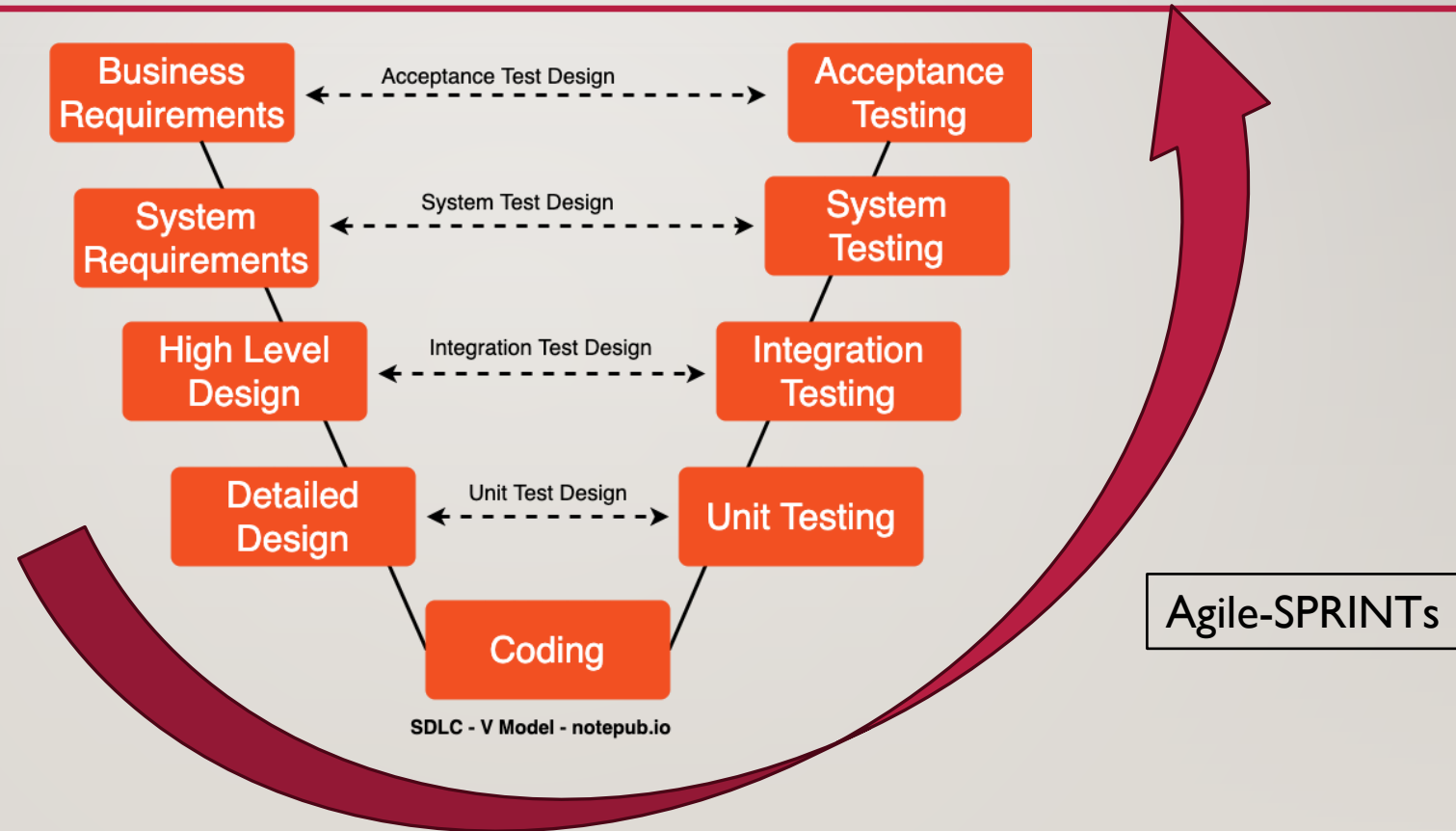
Have Fun Fun Fun!



The Agile - Scrum Framework



SW DEVELOPMENT PROCESS



PROJECT SPRINTS

- Security Alert
 - Object Detection

- System Monitor
 - Display Security Camera and info
 - Display AMR Camera and info
 - Store, display, and report Alerts

- AMR Controller
 - Movement (SLAM)
 - Target Acquisition (Obj. Det.) and Tracking

EXPECTED OUTCOME

- Security Alert and SysMon able to pass topics to update video and data

TEAM EXERCISE 9

Perform integrate and test of System Monitor and Security Alert Modules

RESULTS & CODE REVIEW BY EACH TEAM

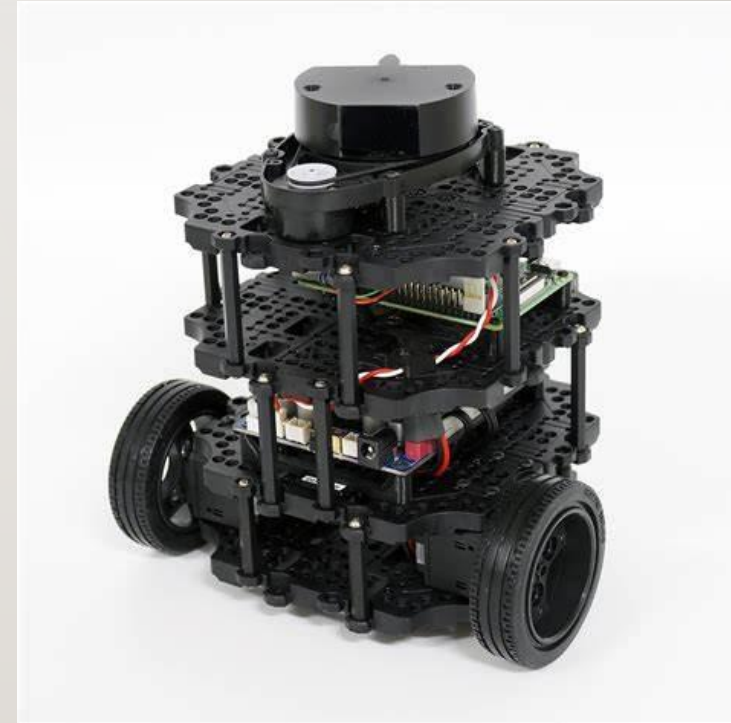
Show actual results against the expected results and explain the code written

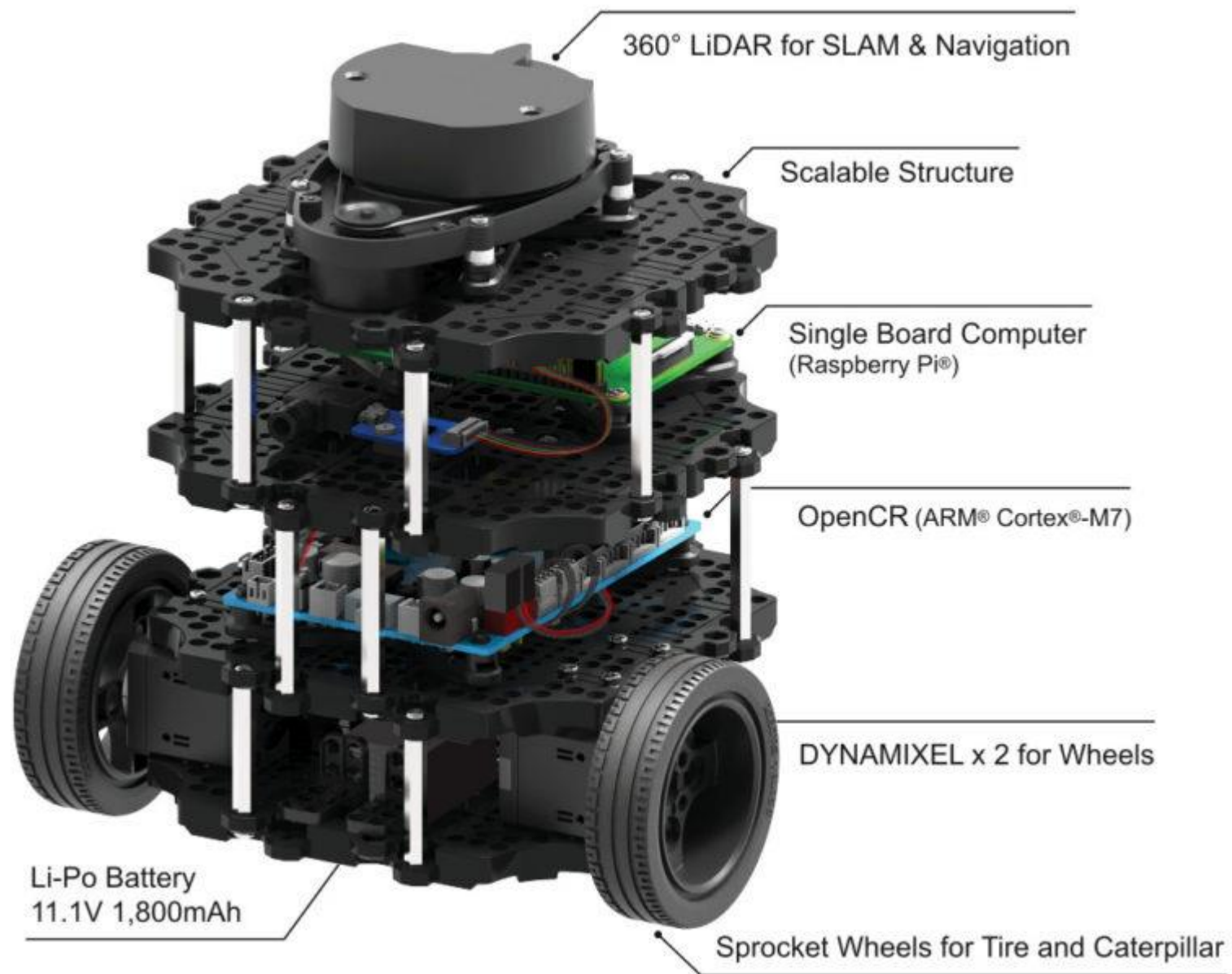
AMR CONTROLLER SPRINT



INTRODUCTION TO AMR

- [TurtleBot3](https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/)
- <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>





SETUP PC FOR AMR

INSTALL DEPENDENT ROS 2 PACKAGES

\$ sudo apt install ros-humble-gazebo-*

\$ sudo apt install ros-humble-cartographer

\$ sudo apt install ros-humble-
cartographer-ros

TURTLEBOT3

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

SETUP PC FOR AMR

INSTALL TURTLEBOT3 PACKAGES

\$ source ~/.bashrc

\$ sudo apt install ros-humble-dynamixel-
sdk

\$ sudo apt install ros-humble-turtlebot3-
msgs

\$ sudo apt install ros-humble-turtlebot3

SETUP PC FOR AMR

- If you want to download the source code

```
$ mkdir -p ~/turtlebot3_ws/src
```

```
$ cd ~/turtlebot3_ws/src/
```

```
$ git clone -b humble-devel  
https://github.com/ROBOTIS-  
GIT/DynamixelSDK.git
```

```
$ git clone -b humble-devel  
https://github.com/ROBOTIS-  
GIT/turtlebot3_msgs.git
```

```
$ git clone -b humble-devel  
https://github.com/ROBOTIS-GIT/turtlebot3.git
```

```
$ cd ~/turtlebot3_ws
```

```
$ colcon build --symlink-install
```

```
$ echo 'source  
~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc
```

```
$ source ~/.bashrc
```

SETUP ROS ID

PC

```
$ echo 'export ROS_DOMAIN_ID=I' >> ~/.bashrc
```

- I,2,3,4,5

```
$ source ~/.bashrc
```

```
$ env | grep ROS
```

HOW TO CONNECT TO AMR

AMR

- Get AMR IP address by physically connect by monitor and keyboard

\$ ifconfig

```
Wireless LAN adapter Wi-Fi 2:
```

```
Connection-specific DNS Suffix  . :  
Link-local IPv6 Address . . . . . : fe80::2bd0:50e1:9694:44%13  
IPv4 Address. . . . . : 192.168.10.14  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.10.1
```

\$ You can find then AMR ID rokey<n> in the linux prompt

CREATE YOUR WORKSPACE UNDER \$HOME DIRECTORY

- `mkdir ~/<my_dir>`
- Put all your file under this directory and remove at the end of the class
- Delete the directory at the end of the class

HOW TO CONNECT TO AMR

PC TERM1



SSH AMR TERM 1 (ONE TIME)

PC TERM2

```
$ dpkg -l | grep openssh
```

If not installed...

```
$ sudo apt install openssh-  
server -y
```

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

```
$ echo 'export  
TURTLEBOT3_MODEL=burger' >>  
~/.bashrc
```

```
$ echo 'export ROS_DOMAIN_ID=1  
#TURTLEBOT3' >> ~/.bashrc
```

```
$ source ~/.bashrc
```

```
$ ros2 launch turtlebot3_bringup  
robot.launch.py
```

```
$ echo 'export  
TURTLEBOT3_MODEL=burger'  
>> ~/.bashrc
```

```
$ source ~/.bashrc
```

```
$ ros2 run turtlebot3_teleop  
teleop_keyboard
```

*allows Vscode to run

SETTING UP VS CODE FOR REMOTE EDITING

- Install VS Code Remote - SSH Extension:
 - Open VS Code on your local machine.
 - Go to the Extensions view (Ctrl + Shift + X).
 - Search for "Remote - SSH" and install it.
- 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.
- Open a Remote Folder:
 - Once connected, VS Code 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 VS Code instance.

DIGITAL MAPPING

STEP1: SSH AMR TERM 1

```
$ source ~/bashrc
```

```
$ ros2 launch turtlebot3_bringup  
  robot.launch.py
```

STEP3: PC TERM 2

```
$ source ~/bashrc
```

```
$ ros2 run turtlebot3_teleop  
  teleop_keyboard
```

STEP2: SSH AMR TERM 2

```
$ source ~/bashrc
```

```
$ ros2 launch turtlebot3_cartographer  
  cartographer.launch.py
```

STEP4: SSH AMR TERM 3(AT THE END)

```
$ source ~/bashrc
```

```
$ ros2 run nav2_map_server  
  map_saver_cli -f ~/<my_dir>/map
```

IF MAPPING IS DONE OF PC MOVE MAP TO AMR

- scp map.yaml map pgm rokey<n>@<rokey IP>:\$HOME

NAVIGATION W/ MAP

STEP1: SSH AMR TERM 1

```
$ source ~/bashrc
```

```
$ ros2 launch turtlebot3_bringup  
  robot.launch.py
```

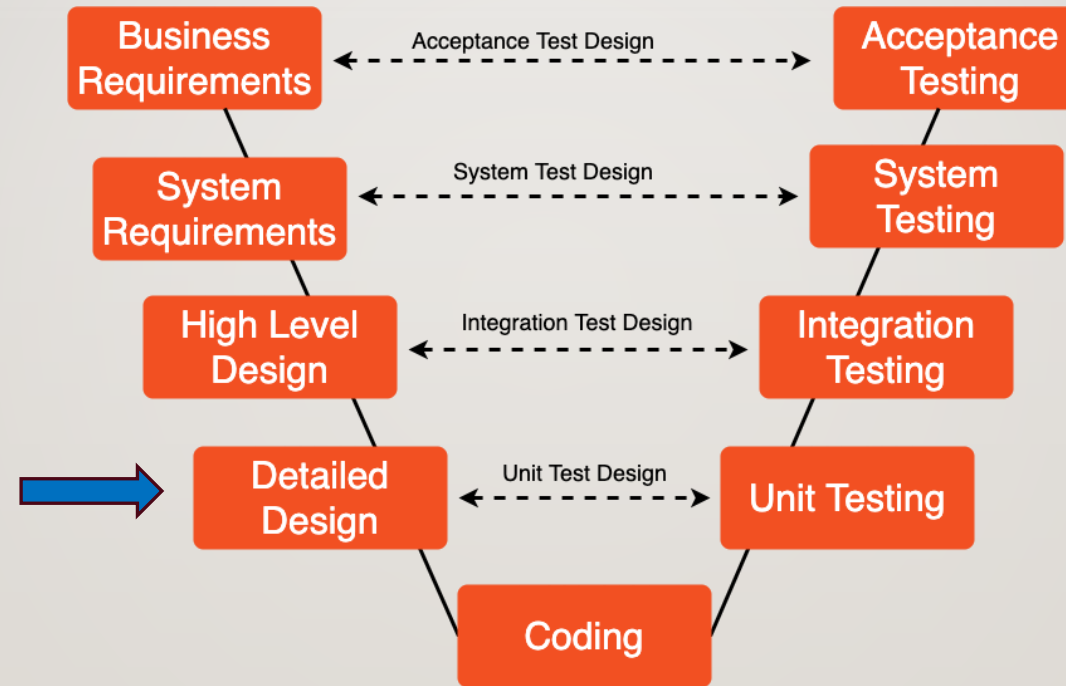
STEP1: SSH AMR TERM 3

```
$ ros2 run rviz2 rviz2
```

STEP2: SSH AMR TERM 2

```
$ ros2 launch turtlebot3_navigation2  
  navigation2.launch.py  
  map:=<map_file_path> (i.e:  
  $HOME/my_dir/map/map.yaml)
```

SPRINT 3 – AMR CONTROLLER



SDLC - V Model - notepub.io

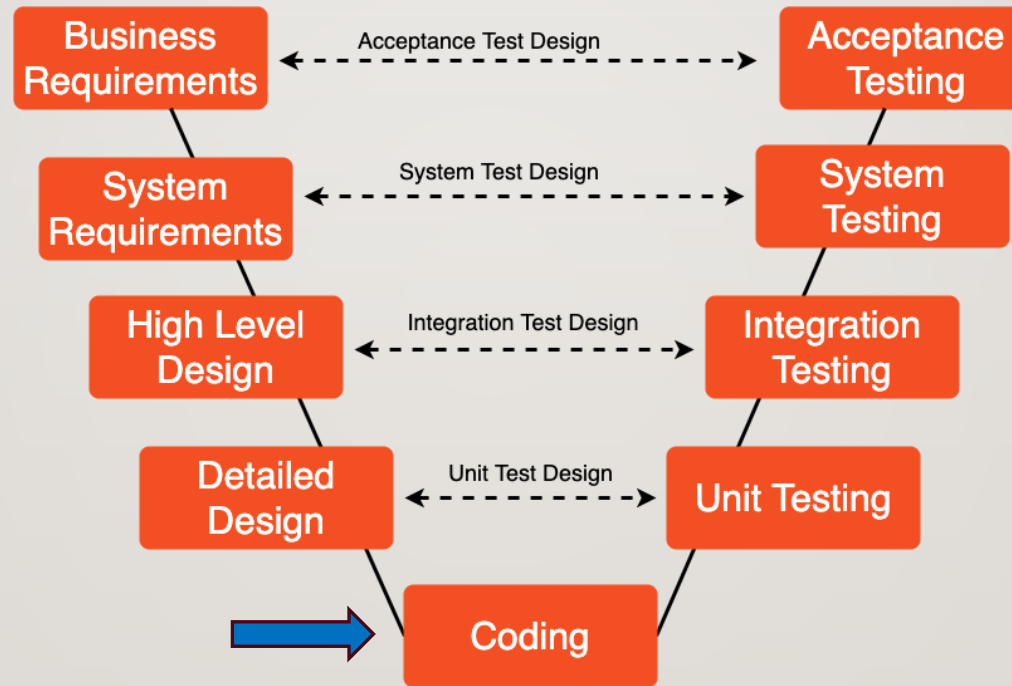
TEAM EXERCISE 10

Perform Detail Design of AMR Controller Module using Process Flow Diagram

DETAIL DESIGN REVIEW BY EACH TEAM

Using the process flow diagram present team's design

SPRINT 3 – AMR CONTROLLER



SDLC - V Model - notepub.io

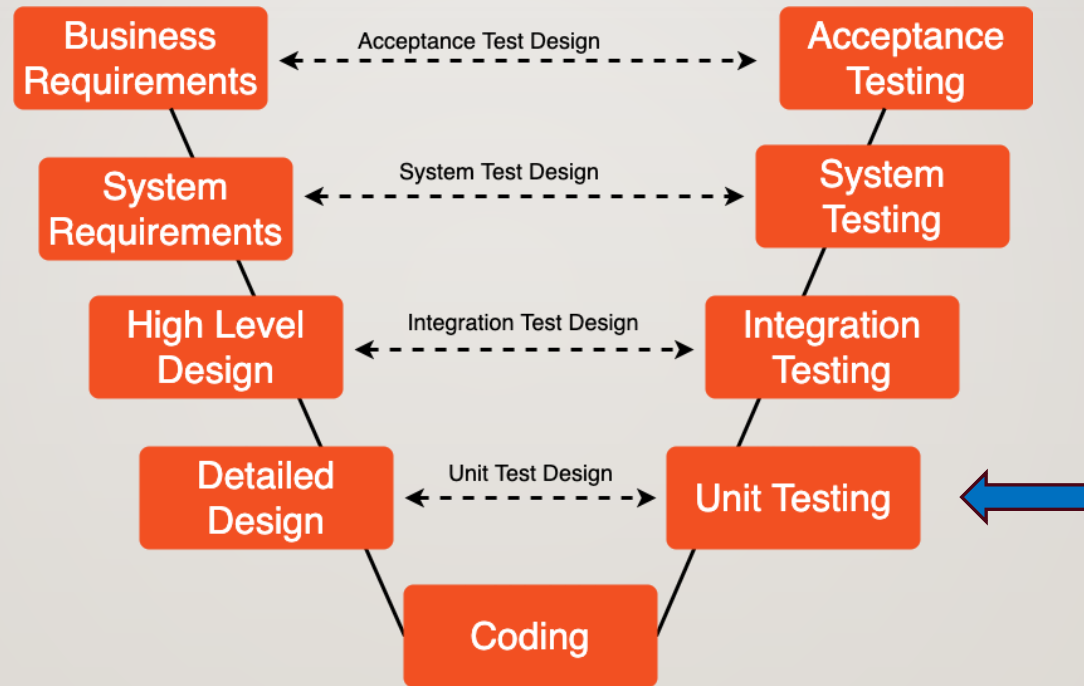
CODING HINT

- Initial Pose
 - nav2
 - rviz2 – 2D estimate pose
 - ros2 topic echo /initialpose
- Sending Goals
 - nav2
 - Rviz2 – send goals
 - ros2 topic echo /amcl_pose
- Stopping Navigation
NavigateToPose.cancel_all_goals_async()
- Sending multiple goals
 - ActionClient
 - /follow_waypoints

EXPECTED OUTCOME

AMR navigates to avoid obstacles, ignores dummies, track, and follow target

SPRINT 3 – AMR CONTROLLER



SDLC - V Model - notepub.io

TEAM EXERCISE I I

Perform coding and testing of AMR Controller Module

RESULTS & CODE REVIEW BY EACH TEAM

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

THE LAST DAY

- 9:30 – 4:00 p.m
 - System Integration & Test
 - Final Presentation Prep
- 4:00 – 5:40 p.m.
 - Live Demonstration 5 minutes
 - Presentation 15 minutes
- Equipment Return and Rap up

최종 프로젝트 발표



FINAL PRESENTATION MATERIAL PLANNING

- Solution Overview
 - Key Issues and Challenges
 - How did you overcome
 - Required Solution Improvements
 - Lessons Learned
 - Team Contribution
- 20 minutes

팀원 과 업무 책임



- 업무 책임
 - ...
- 숙련된 기술
 -