GOOD MORNING!
早上好!
안녕하세요!

PROJECT INTRODUCTION

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 (DONE)

- 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 (DONE?)

- 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
- 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

DAY 3 (DONE?)

- Porting to ROS
  - Update Sysmon Node code
  - Update the database with received Obj. Det. Data from Detection Alert Node
  - Display the content of DB on System Monitor web page

- And finally, Integration and Test of Detection Alert & System Monitor

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/ obj. following (teleop)
- Porting to ROS
  - Create AMR Controller Node
  - Create and send Obj. Tracking Image and data to Sysmon

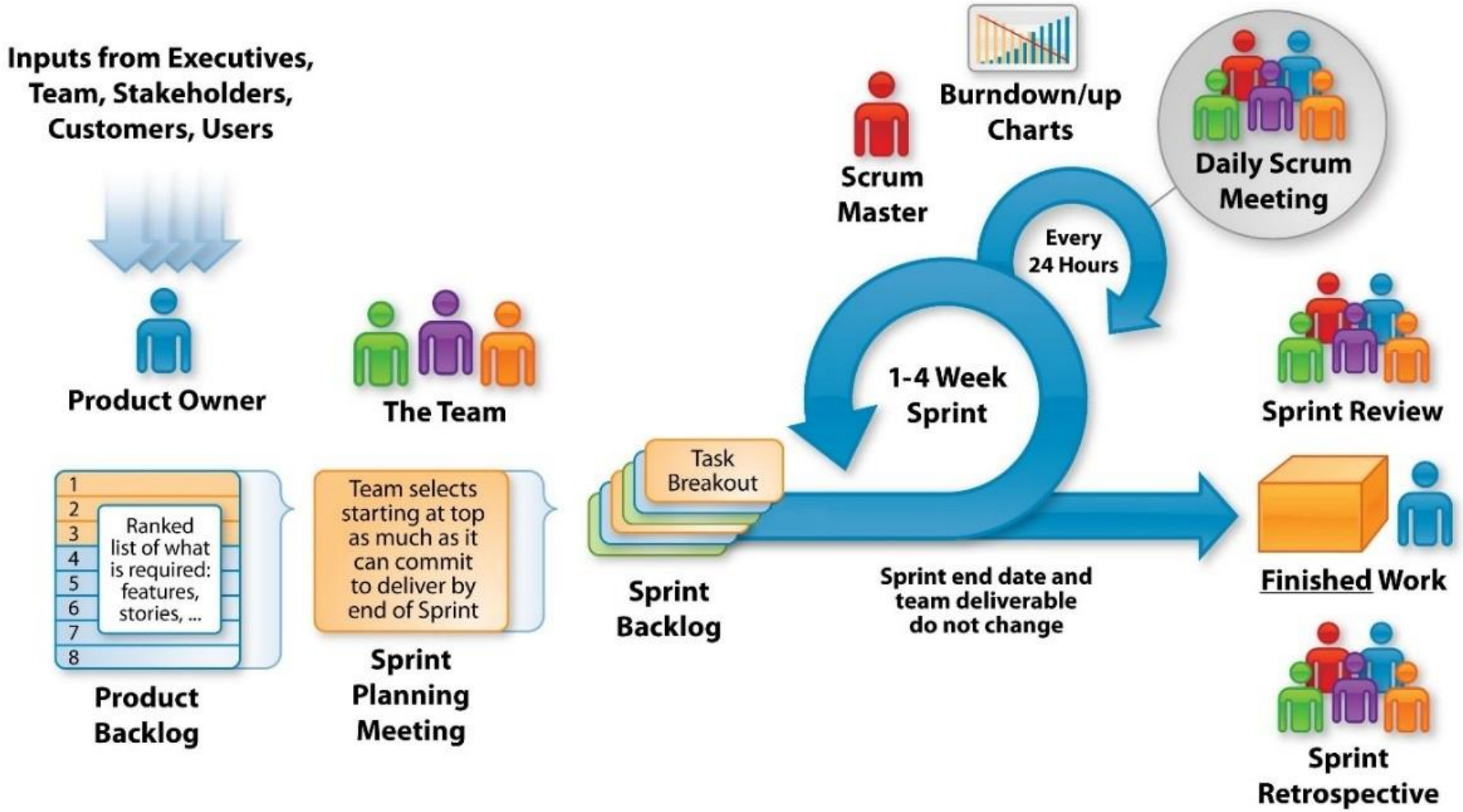
DAY 5

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

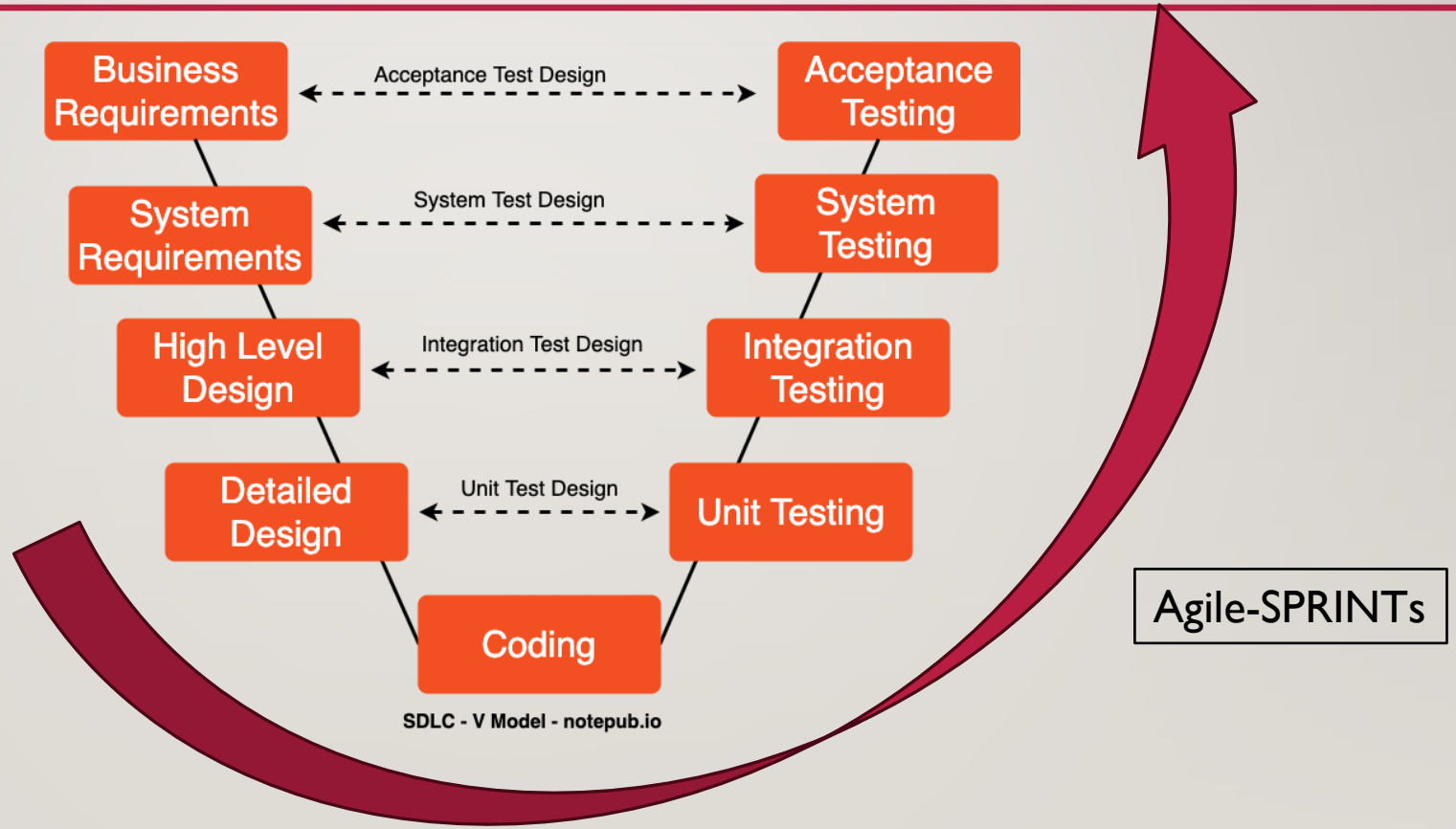
프로젝트 RULE NUMBER ONE!!!

Have Fun Fun Fun!

The Agile - Scrum Framework



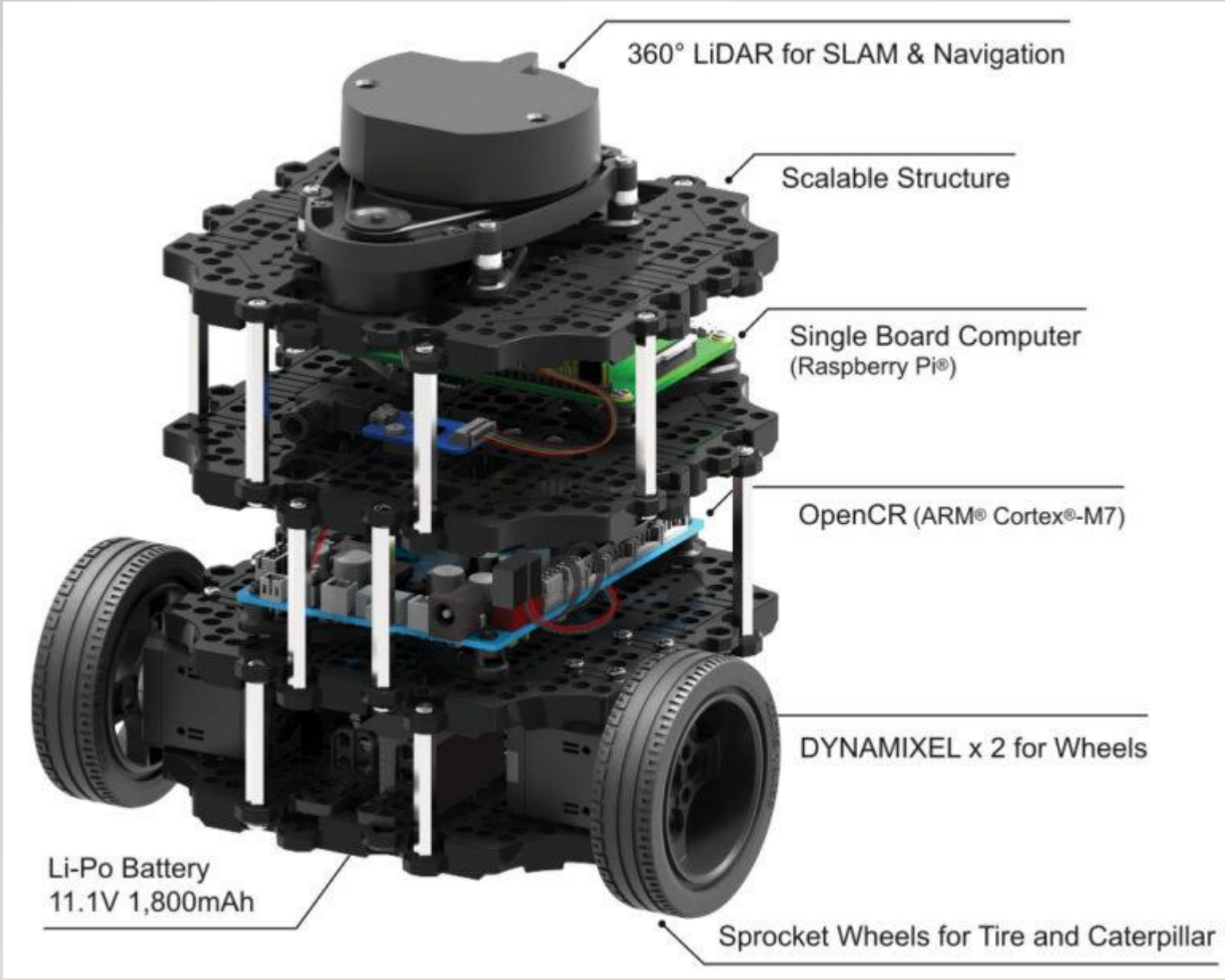
SW DEVELOPMENT PROCESS



Agile-SPRINTs

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

SETUP PC FOR AMR

TURTLEBOT3

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

CHECK IF INSTALLED

$ apt list | grep gazebo

$ apt list | grep carto

INSTALL DEPENDENT ROS 2 PACKAGES

If not already installed,

$ sudo apt install ros-humble-gazebo-*

$ sudo apt install ros-humble-cartographer

$ sudo apt install ros-humble-cartographer-ros

SETUP PC FOR AMR

INSTALL DEPENDENT ROS 2 PACKAGES

$ Check if installed

    $ apt list | grep gazebo

    $ apt list | grep carto

If not,

$ 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

CHECK IF ALREADY INSTALLED

$ apt list | grep dynam

$ apt list | grep turtlebot3

INSTALL TURTLEBOT3 PACKAGES

If not already installed,

$ 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

$ cat ~/.bashrc

If ROS_DOMAIN_ID does not exist or is not correctly set,

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

(ID = 1,2,3,4,5 depends on your team number)

$ source ~/.bashrc

$ env | grep ROS

HOW TO CONNECT TO AMR

PC

$ sudo ufw status

$ sudo ufw disable

- disables firewall for ubuntu systems

HOW TO CONNECT TO AMR

PC/AMR

Connect to rokey or rokey_APTest

Must be same network for both

*rokey_APTest does not have internet access

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

HOW TO CONNECT TO AMR

PC TERM1

$ dpkg -l | grep openssh

If not installed…

$ sudo apt install openssh-server –y

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

*allows Vscode/Rviz to run

HOW TO CONNECT TO AMR

PC TERM1

$ ros2 run demo_nodes_cpp talker

SSH AMR TERM1

$ cat ~/.bashrc

If ROS_DOMAIN_ID is not correctly set,

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

(ID = 1,2,3,4,5 depends on your team number)

$ ros2 run demo_nodes_cpp listener

CREATE YOUR WORKSPACE UNDER AMR $HOME DIRECTORY

AMR

$ mkdir

~/rokey1_<grp_letter><grp_num>_ws

(i.e. mkdir ~/rokey1_A2_ws)

- Put all your file under this directory and remove at the end of the class
- Delete the directory at the end of the class

SETTING UP TO USE TURTLEBOT3

PC TERM 1

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

SSH AMR TERM 1

$

SETTING UP TO USE TURTLEBOT3

SSH AMR TERM 1

$ cat ~/.bashrc

Check if export command on the right is in the .bashrd, if not execute the command on the right

SSH AMR TERM 1

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

SETTING UP TO USE TURTLEBOT3

PC TERM 1

$ cat ~/.bashrc

Check if export command on the right is in the .bashrd, if not execute the command on the right

PC TERM 1

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

USING AMR TELEOP

SSH AMR TERM 1

$ source ~/.bashrc

$ ros2 launch turtlebot3_bringup robot.launch.py

….. running

PC TERM2

$ source ~/.bashrc

$ ros2 run turtlebot3_teleop teleop_keyboard

USING AMR TELEOP

PC TERM1

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

SSH AMR TERM 1

$ ros2 launch turtlebot3_bringup robot.launch.py

…..running

PC TERM2

$ ros2 run turtlebot3_teleop teleop_keyboard

…..running

DIGITAL MAPPING

STEP1 : SSH AMR TERM 1

```
$ source ~/bashrc
$ ros2 launch turtlebot3_bringup
  robot.launch.py
```

STEP2: SSH AMR TERM 2

```
$ source ~/bashrc
$ ros2 launch turtlebot3_cartographer
  cartographer.launch.py
```

DIGITAL MAPPING

STEP1: SSH AMR TERM 1

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

….. running
```

STEP2: SSH AMR TERM 2

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

….. running
```

STEP3: PC TERM 2

```
$ source ~/bashrc

$ ros2 run turtlebot3_teleop
  teleop_keyboard
```

DIGITAL MAPPING

STEP1: SSH AMR TERM 1

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

….. running
```

STEP2: SSH AMR TERM 2

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

….. running
```

STEP3: PC TERM 2

```
$ ros2 run turtlebot3_teleop
  teleop_keyboard

….. running
```

STEP4: SSH AMR TERM 3(AT THE END)

```
$ source ~/bashrc

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

NAVIGATION W/ MAP

STEP1 : SSH AMR TERM 1

```
$ source ~/bashrc
$ ros2 launch turtlebot3_bringup
  robot.launch.py
```

STEP2: SSH AMR TERM 2

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

NAVIGATION W/ MAP

STEP1 : SSH AMR TERM 1

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

….running
```

STEP2: SSH AMR TERM 2

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

….running
```

*Perform 2D Pose Estimate and Send Goal

IF MAPPING IS DONE OF PC MOVE MAP TO AMR

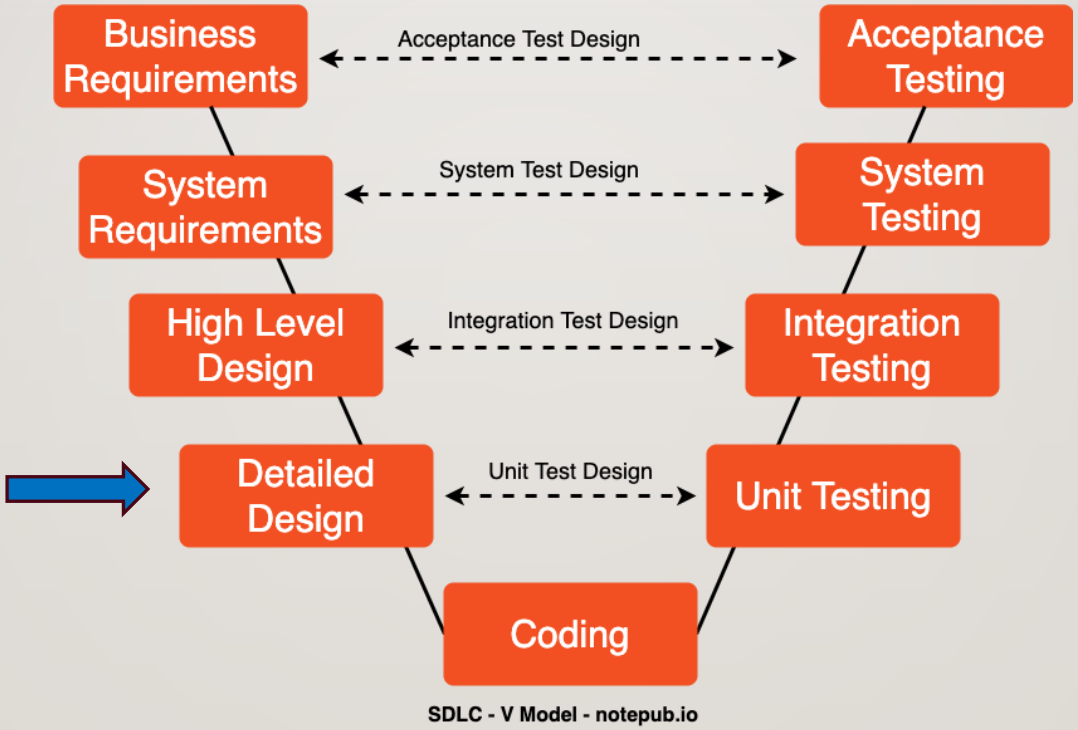
- scp map.yaml map pgm rokey<n>@<rokey IP>:$HOME

PROJECT SPRINTS

- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems
- System Monitor
  - Receive and Display Detection Camera and info
  - Receive and Display AMR Camera and info
  - Store, display, and report Information and Alerts
- 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

AMR CONTROLLER SPRINT

SPRINT 3 – AMR CONTROLLER



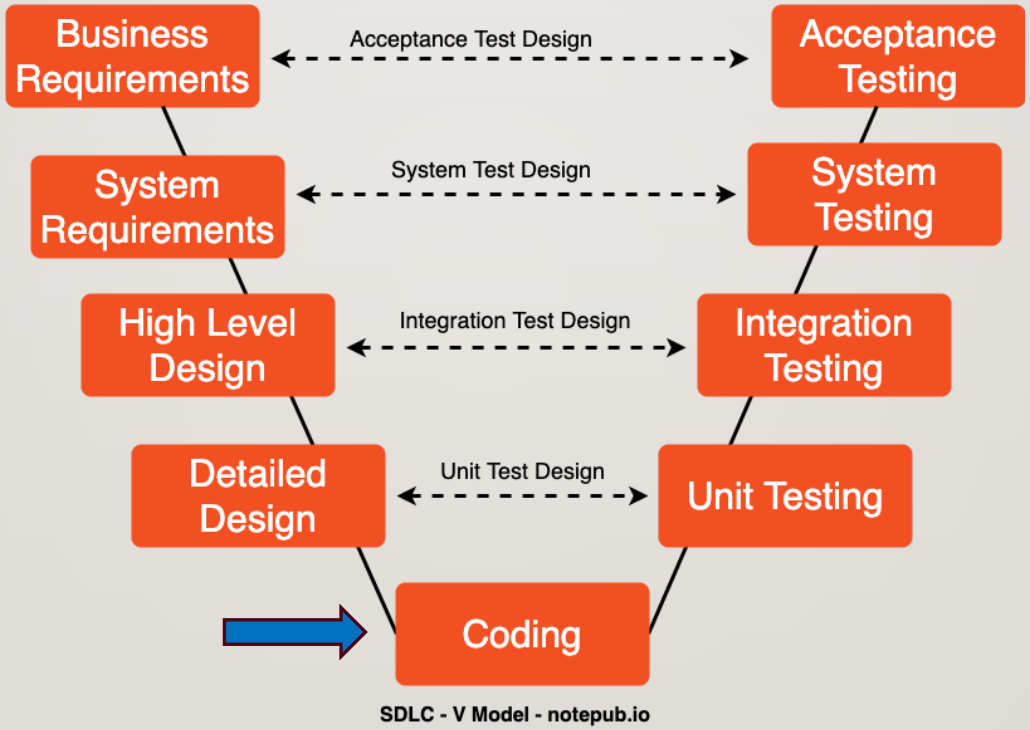
TEAM EXERCISE 10

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



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.
- 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, 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.

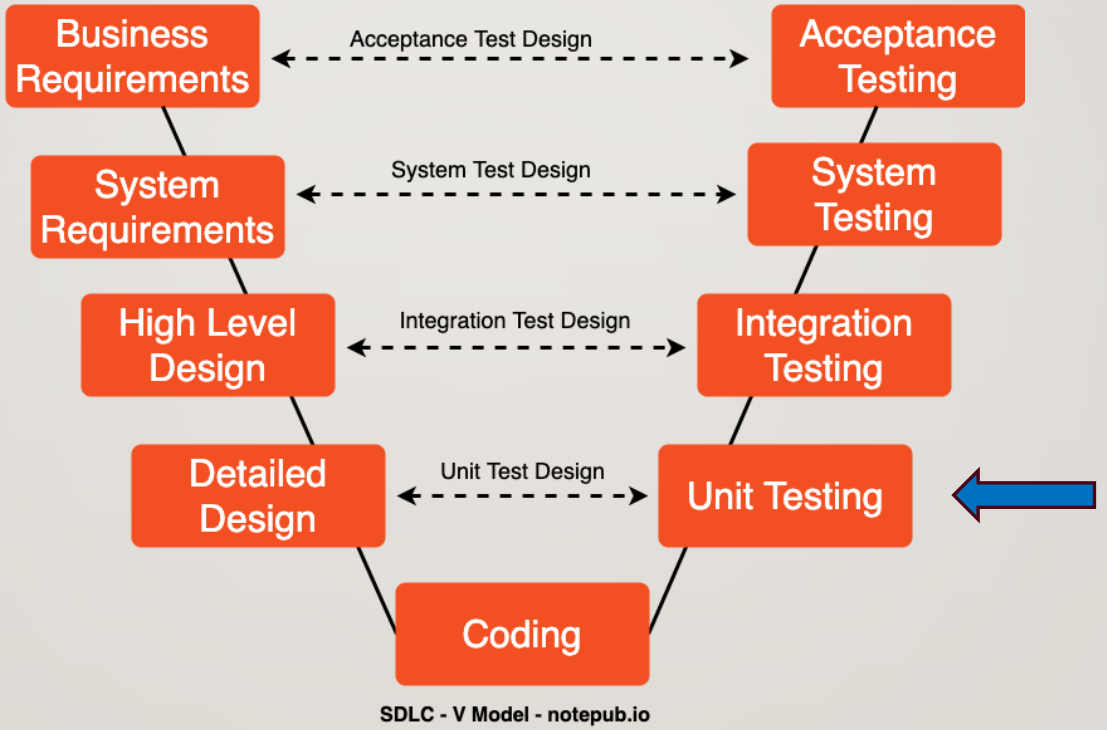
CODING HINT

- 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



TEAM EXERCISE 11

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
  - ….

Andreas Kim