GOOD MORNING!
早上好!
안녕하세요!

DAY 5

DAY 1

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

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 (IF NEEDED)

- 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

- 환경 구축
- 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

- Business Requirement Development
- System Requirement Development
- Time Management
- System(High Level) Design
- Begin Detail Design to Acceptance - Agile Development (SPRINTs)

DAY 4 (MINI PROJECT)

SYSTEM DESIGN

- Mini Project

DETAIL DESIGN

- Detection
- AMR Control

DAY 5 (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 6 (FINAL PROJECT)

- 시스템 설계 및 프로세스 정립
- 비즈니스 요구 사항 업데이트
- 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)

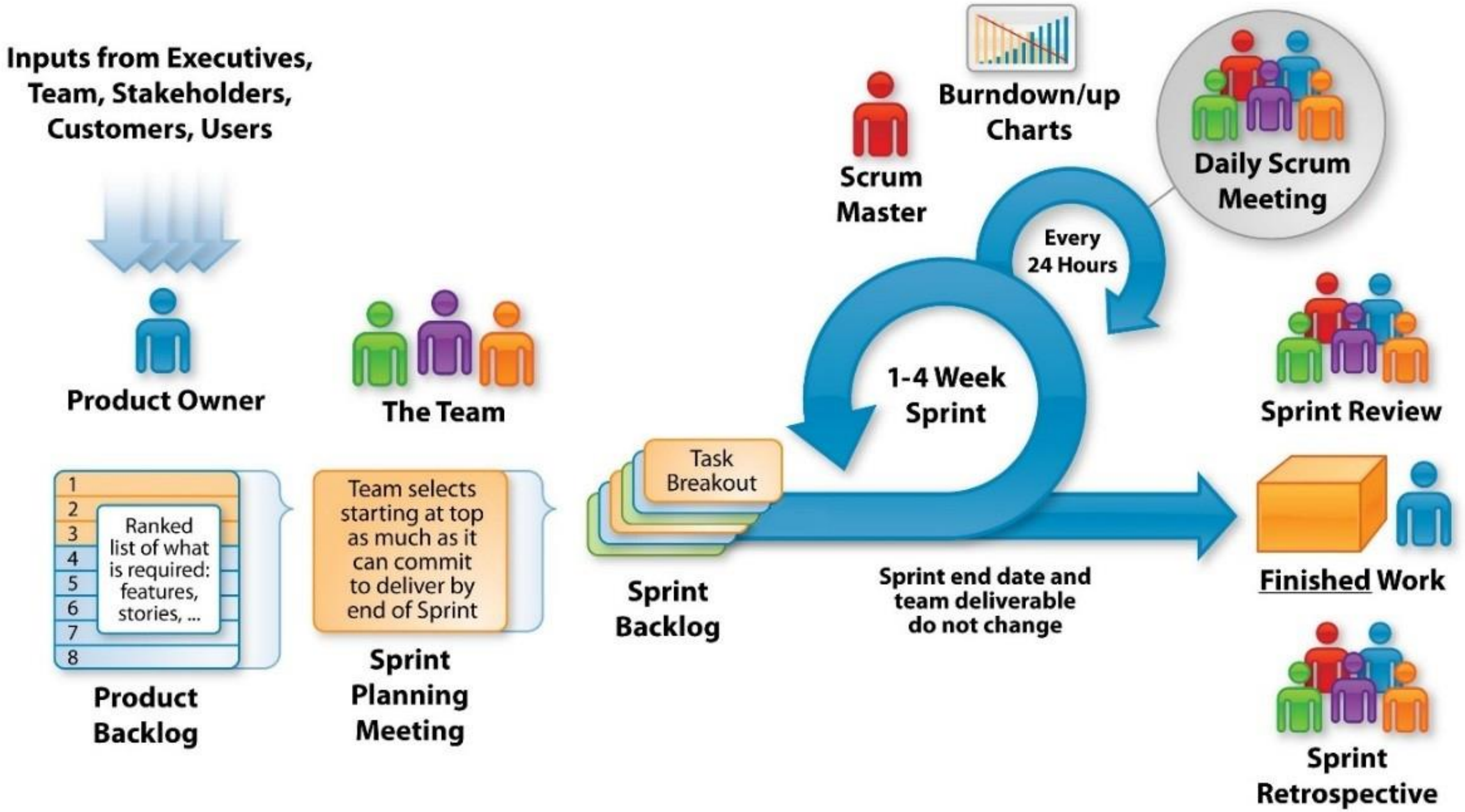
- Flask 를 이용한 웹 서버 구축 (System Monitor)
  - Flask/HTML Intro
  - Deploy YOLOv8 Obj. Det results to web
  - Log in 기능 구현
  - Sysmon 웹기능 구현

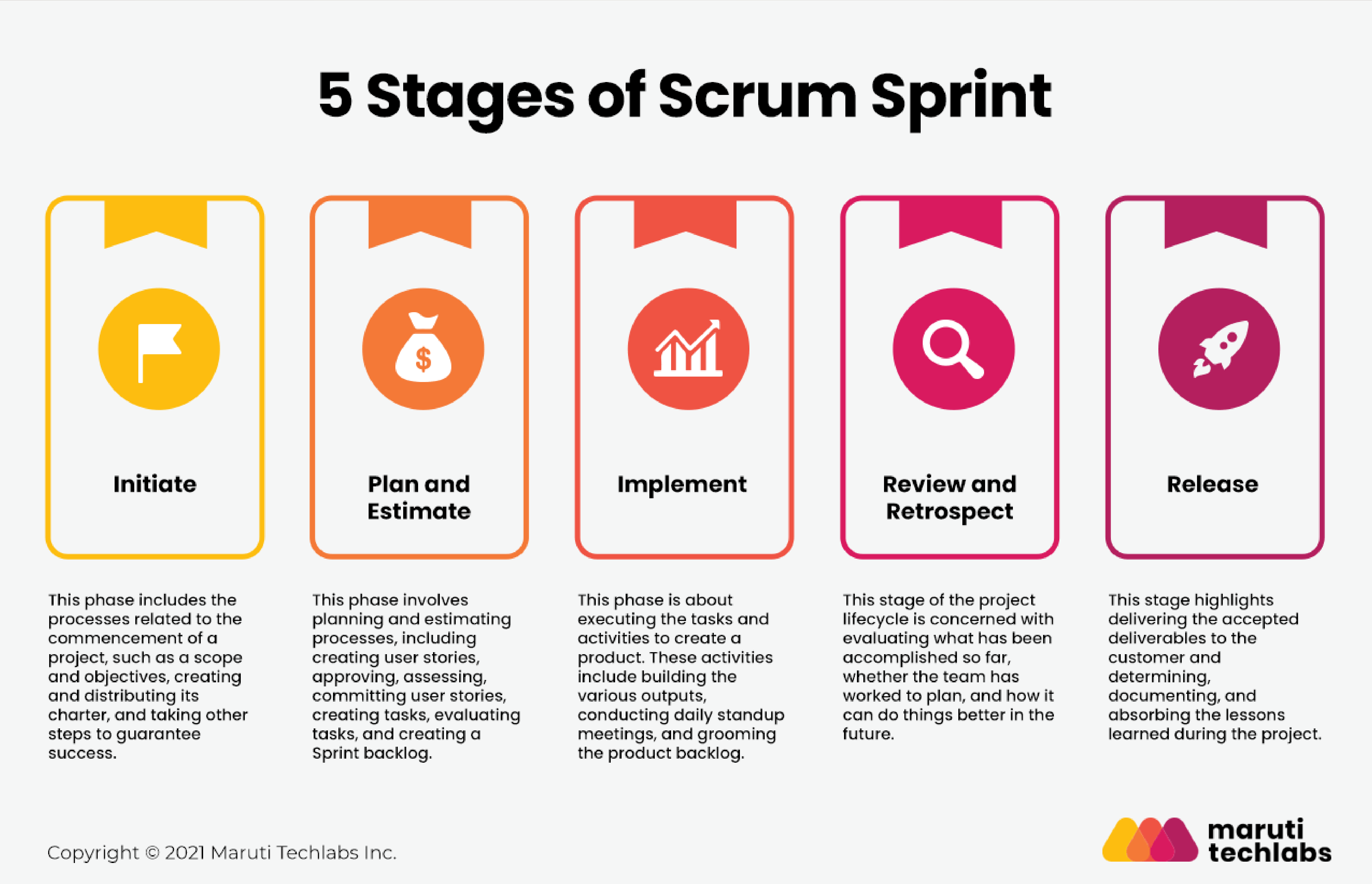
- SQLite3를 이용한 데이터베이스 구축 및 연동 (System Monitor)
  - SQLite3 기본 기능 구현
  - DB 기능 구축
  - 저장된 내용 검색하는 기능 구현

프로젝트 RULE NUMBER ONE!!!

Have Fun Fun Fun!

The Agile - Scrum Framework


Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint end date and team deliverable do not change
Sprint Review
Finished Work
Sprint Retrospective



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

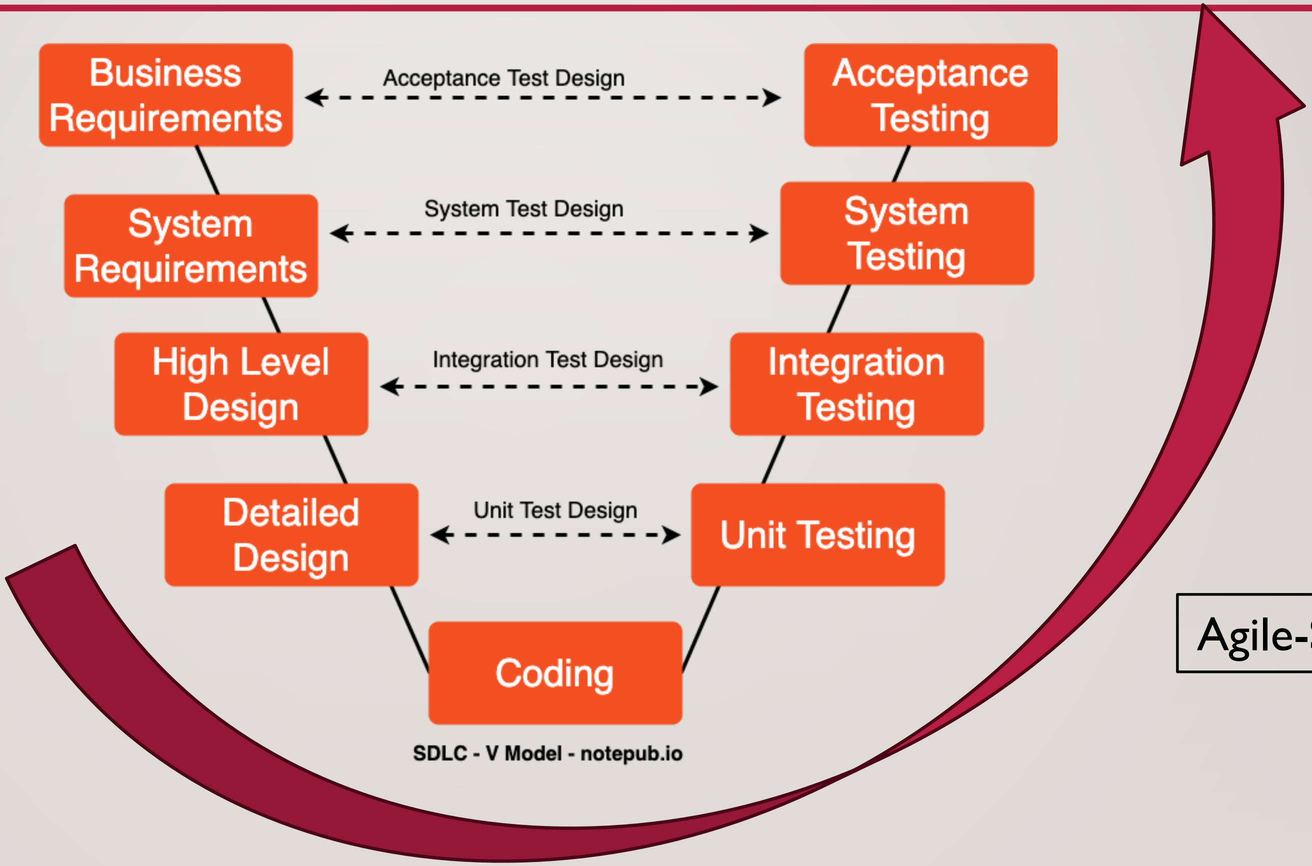
This stage of the project lifecycle is concerned with evaluating what has been accomplished so far, whether the team has worked to plan, and how it can do things better in the future.

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



SDLC - V Model - notepub.io

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

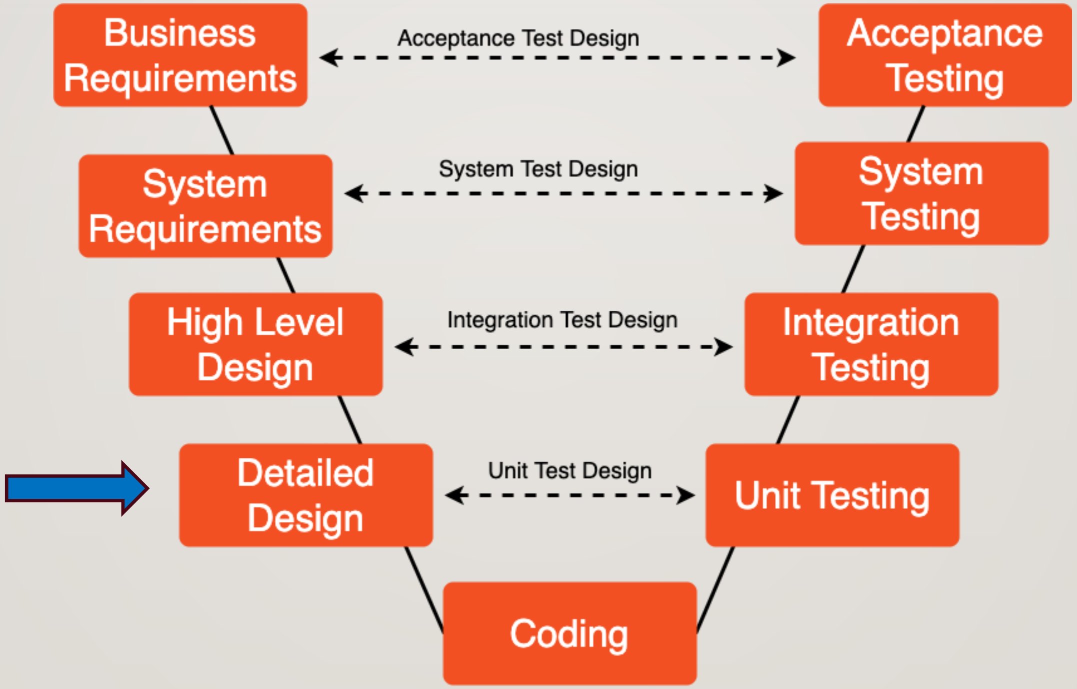
PROJECT SPRINTS

- 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

BEGINS SPRINTS
DETAIL DESIGN/CODING/TESTING

SPRINT 1 - DETECTION ALERT



SDLC - V Model - notepub.io

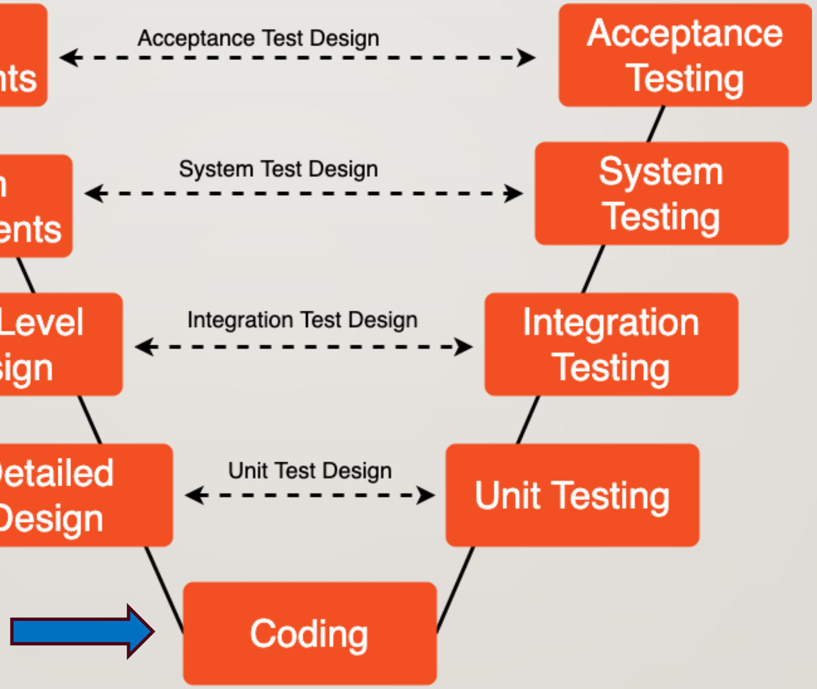
TEAM EXERCISE 4

Perform Detail Design of Detection Alert Module using Process Flow Diagram

DETAIL DESIGN REVIEW BY EACH TEAM

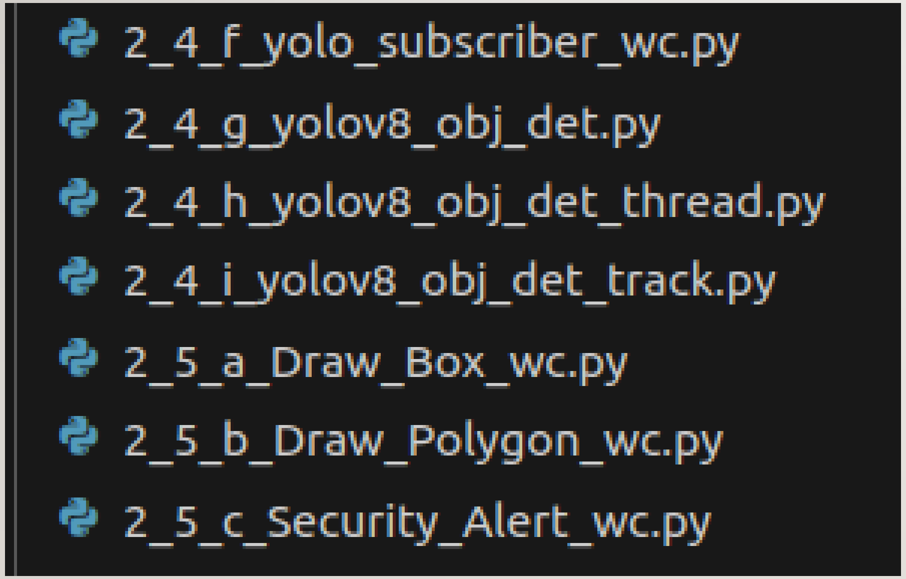
Using the process flow diagram present team's design

SPRINT 1 - DETECTION ALERT



SDLC - V Model - notepub.io

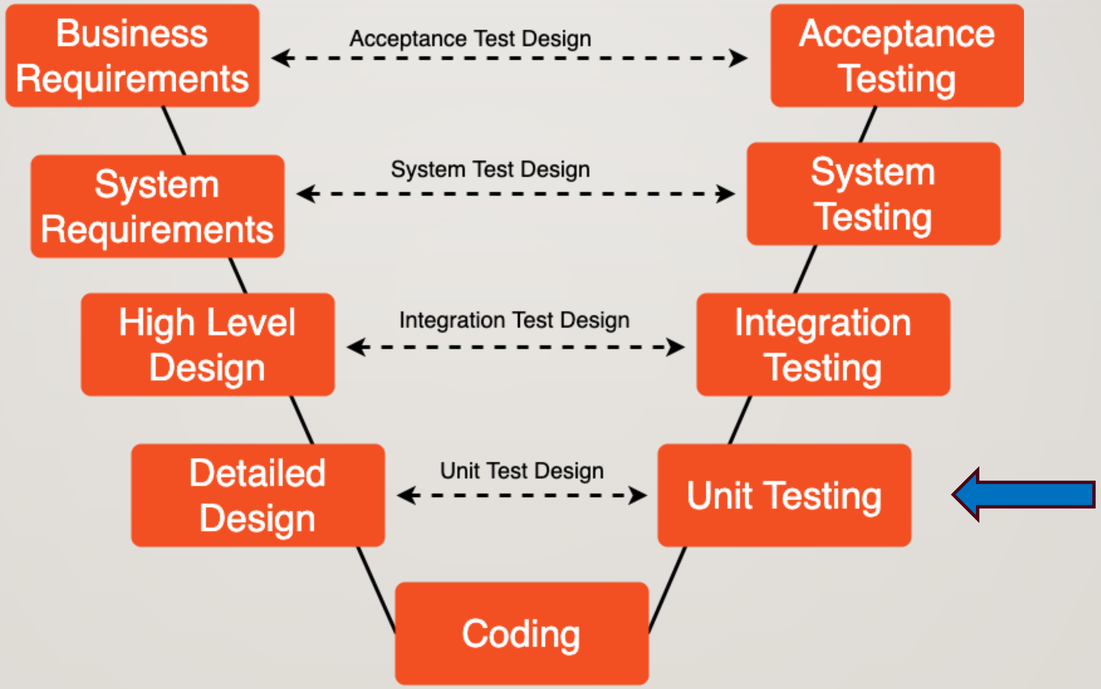
CODING HINTS



2_4_f_yolo_subscriber_wc.py
2_4_g_yolov8_obj_det.py
2_4_h_yolov8_obj_det_thread.py
2_4_i_yolov8_obj_det_track.py
2_5_a_Draw_Box_wc.py
2_5_b_Draw_Polygon_wc.py
2_5_c_Security_Alert_wc.py

- Create your detection/alert condition

SPRINT 1 - DETECTION ALERT



Business Requirements

Acceptance Test Design

Acceptance Testing

System Requirements

System Test Design

System Testing

High Level Design

Integration Test Design

Integration Testing

Detailed Design

Unit Test Design

Unit Testing

Coding

SDLC - V Model - notepub.io

TEAM EXERCISE 5

Perform coding and testing of Detection Alert Module

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

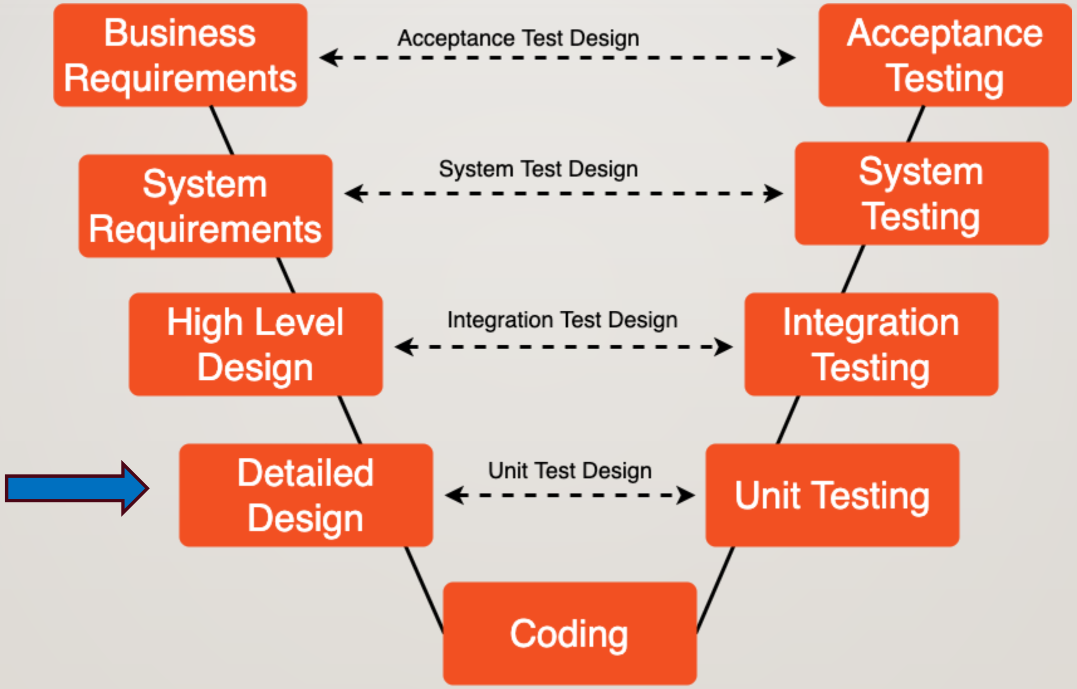
PROJECT SPRINTS

- 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

AMR CONTROLLER SPRINT

SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

TEAM EXERCISE 6

Perform Detail Design of AMR Controller Module using Process Flow Diagram

DESIGN QUESTIONS:

- How do you start the robot?
  - Initial Position
- How do you find AMR current position and orientation?
- How do you sending Goals?
  - Single goal
  - Multiple goals
- How do you manual control of AMR Odometry?
  - How to move forward, backward, left and right???

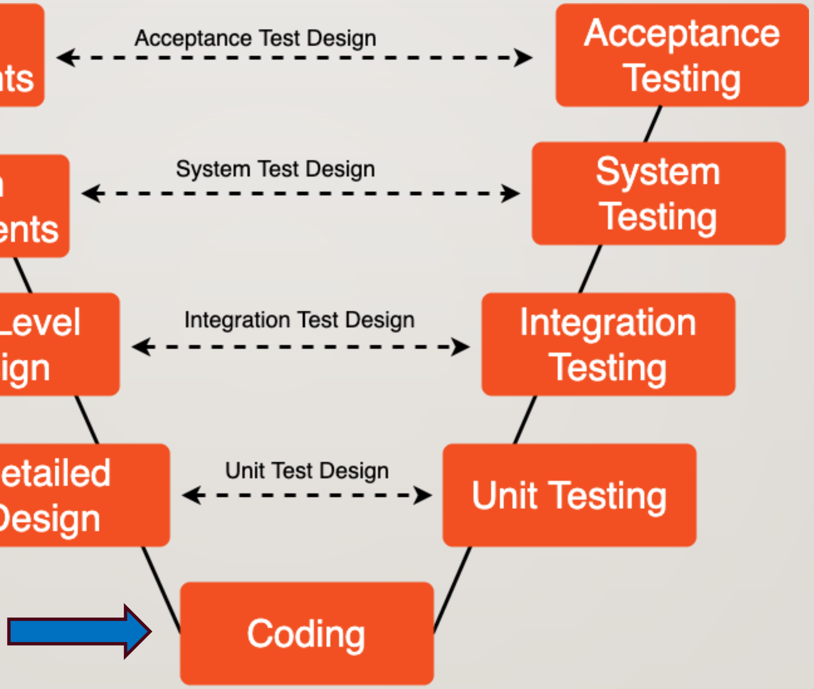
WHAT IS THE FOLLOW ALGORITHM?

- Left/Right?
- Forward/Backward?
- Velocity?
- Camera position?
- Depth? Local/Global Coordinate transform?

DETAIL DESIGN REVIEW BY EACH TEAM

Using the process flow diagram present team’s design

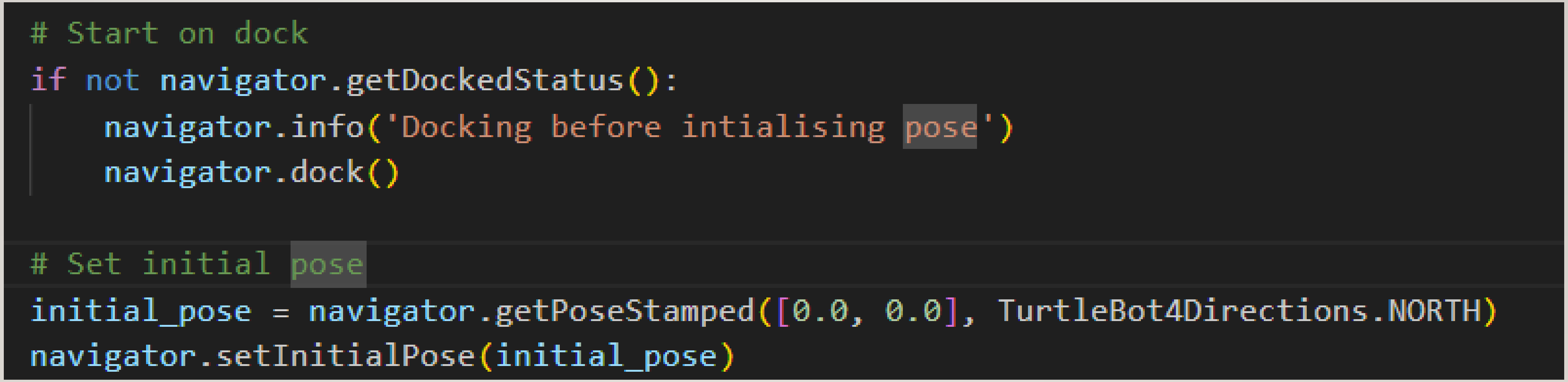
SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

CODING HINTS:

- How do you start the robot?



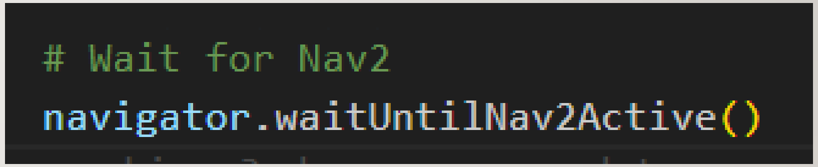
```python
# Start on dock
if not navigator.getDockedStatus():
    navigator.info('Docking before intialising pose')
    navigator.dock()

# Set initial pose
initial_pose = navigator.getPoseStamped([0.0, 0.0], TurtleBot4Directions.NORTH)
navigator.setInitialPose(initial_pose)
```

CODING HINTS:

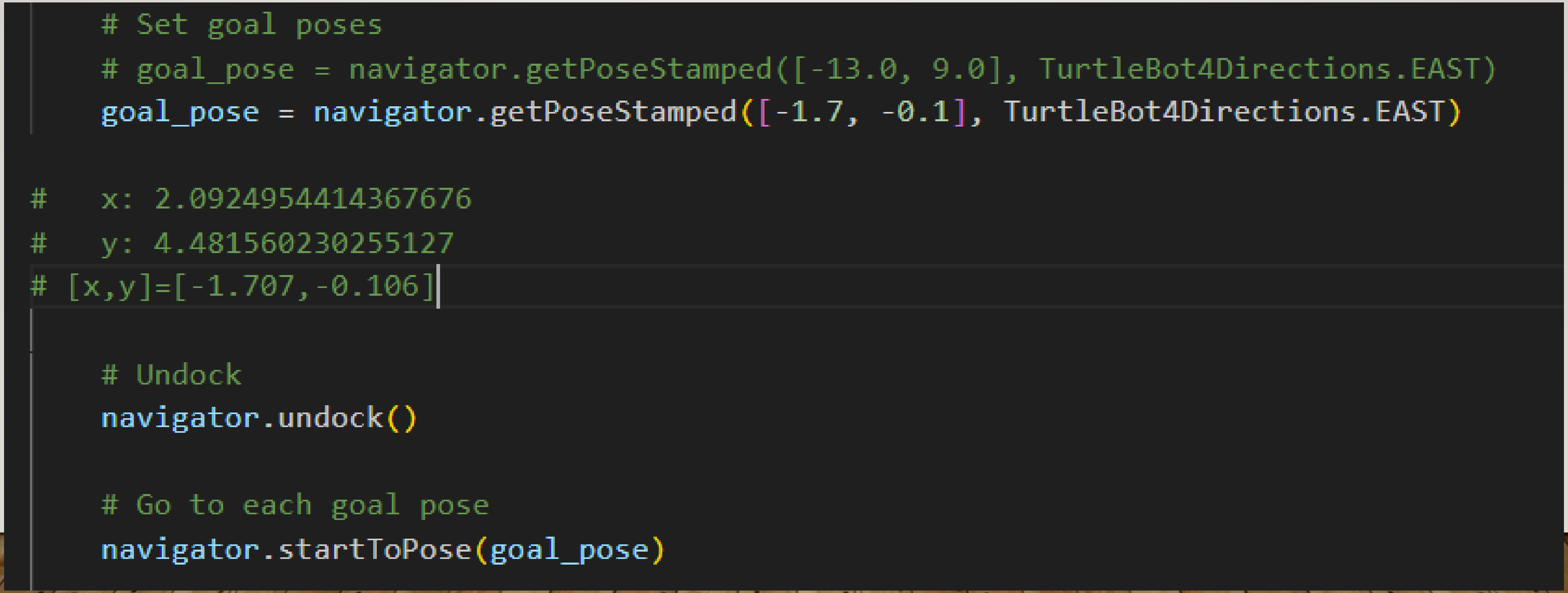
- How do you find AMR current position and orientation?
  - Echo topic: amcl_pose
  - Use Rviz Publish Points & Echo clicked_points
  - Etc…

CODING HINTS:

- How do you sending Goals?
  - Single goal



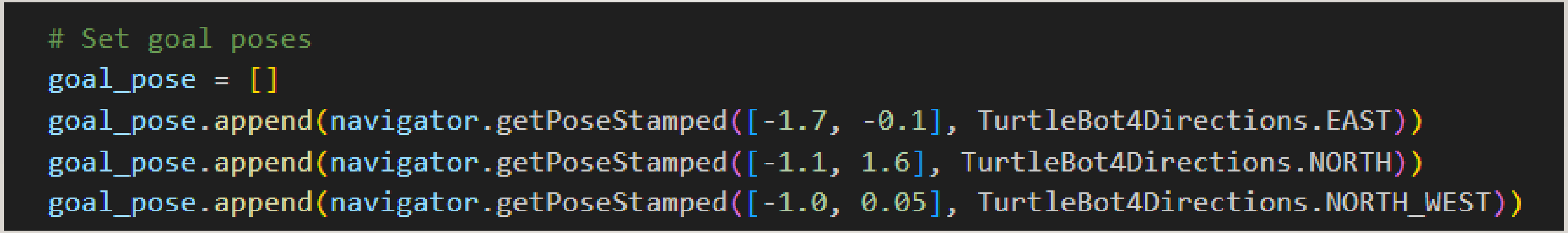
```
# Wait for Nav2
navigator.waitUntilNav2Active()
```



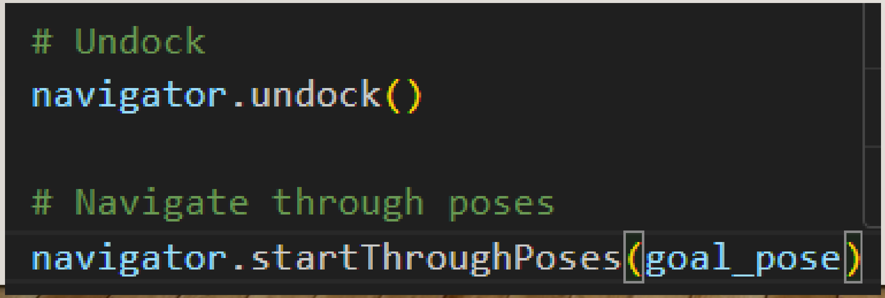
```
    # Set goal poses
    # goal_pose = navigator.getPoseStamped([-13.0, 9.0], TurtleBot4Directions.EAST)
    goal_pose = navigator.getPoseStamped([-1.7, -0.1], TurtleBot4Directions.EAST)

#   x: 2.0924954414367676
#   y: 4.481560230255127
# [x,y]=[-1.707,-0.106]

    # Undock
    navigator.undock()

    # Go to each goal pose
    navigator.startToPose(goal_pose)
```

CODING HINTS:

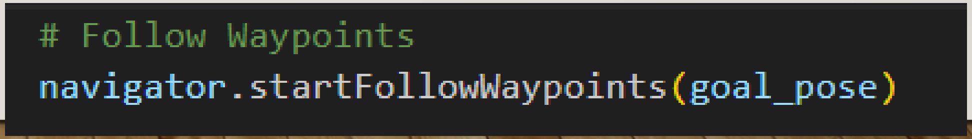
- How do you sending Goals?
  - Single goal
  - Multiple goals



```python
# Set goal poses
goal_pose = []
goal_pose.append(navigator.getPoseStamped([-1.7, -0.1], TurtleBot4Directions.EAST))
goal_pose.append(navigator.getPoseStamped([-1.1, 1.6], TurtleBot4Directions.NORTH))
goal_pose.append(navigator.getPoseStamped([-1.0, 0.05], TurtleBot4Directions.NORTH_WEST))
```



```python
# Undock
navigator.undock()

# Navigate through poses
navigator.startThroughPoses(goal_pose)
```



```python
# Follow Waypoints
navigator.startFollowWaypoints(goal_pose)
```

CODING HINTS:

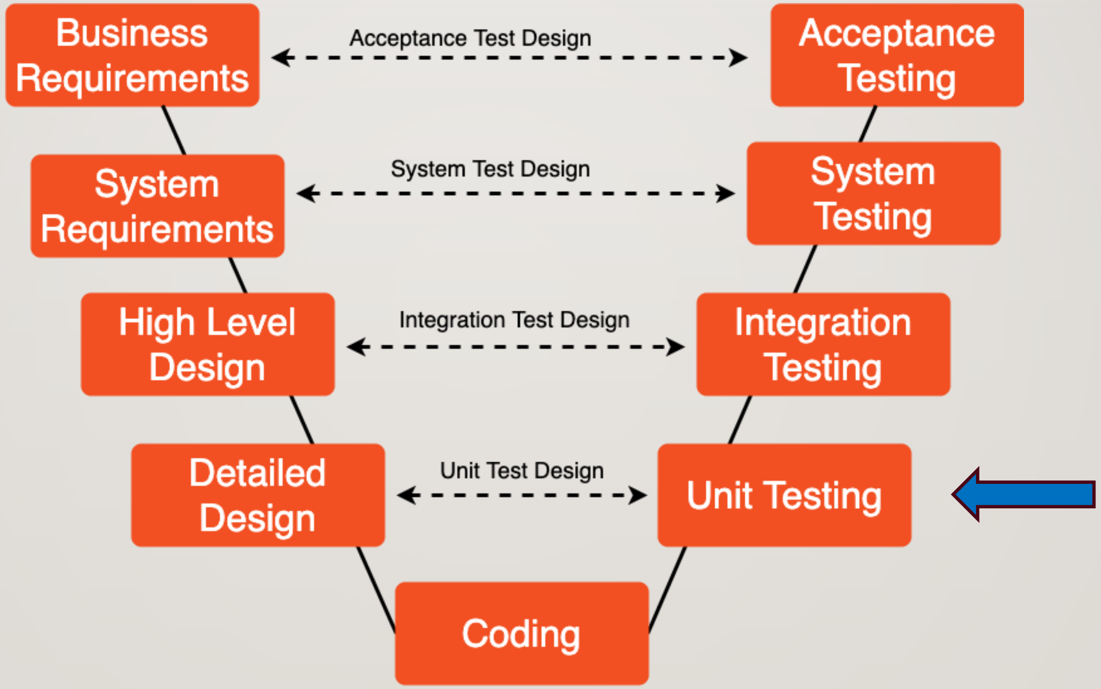
- How do you manual control of AMR Odometry?
  - How to move forward, backward, left and right???

- Use teleops to move the robots
  - Echo topic: cmd_vel
- geometry_msgs.msg
  - Twist
    - twist.linear.x = <+/-n>
    - twist.angular.z = <+/-n>
  - self.cmd_publisher.publish(twist)

EXPECTED OUTCOME

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

SPRINT 2 – AMR CONTROLLER



SDLC - V Model - notepub.io

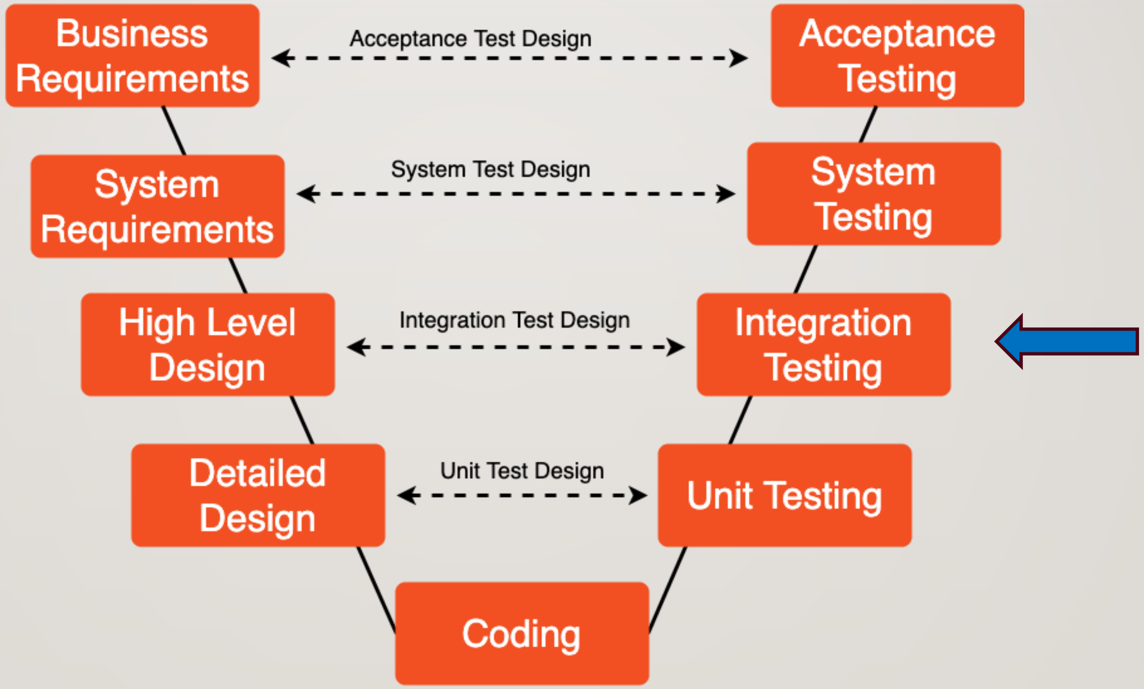
TEAM EXERCISE 7

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

SPRINT 1&2 – DETECTION ALERT/AMR CONTROLLER INTEGRATION & TEST



SDLC - V Model - notepub.io

EXPECTED OUTCOME

- Detection Alert and AMR Controller able to pass topics for necessary actions between

TEAM EXERCISE 8

Perform integrate and test of Detection Alert and AMR Controller Modules

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!