Karlsruhe Institute of Technology

# Seamless Engineering Milestone 1 - Requirements Template

**Team number: Seamless Engineering Group 06**

## Requirements

If there is one aspect a project must have in order not to be doomed to failure, it is a reasonable and comprehensive repository of both functional and non-functional requirements. A project's requirements must be well considered, balanced and easily understood by all team-members, but perhaps most importantly, they must not be dropped or compromised during half of the project.

Fill out the following templates according to your requirements.

### Definition

Source: Projektmanagement in der Entwicklung von Produkten für sicherheitskritische Anwendungen, Prof. Nolle, ITIV, 2021

| | |
|---|---|
| Must | Overriding requirements that must be complied with by both the purchaser and the supplier (by law, standard, rules, ...). |
| Shall | Indispensable requirements; a deviation is not permitted without formal agreement between buyer and supplier. |
| Should | Recommendation or indication of the implementation of a requirement; a deviation is only allowed in justified cases. |
| Will | Statement of intent in connection with a requirement. |
| May | Permitted execution or deviation, no requirement. |

Karlsruhe Institute of Technology

## Functional Requirements

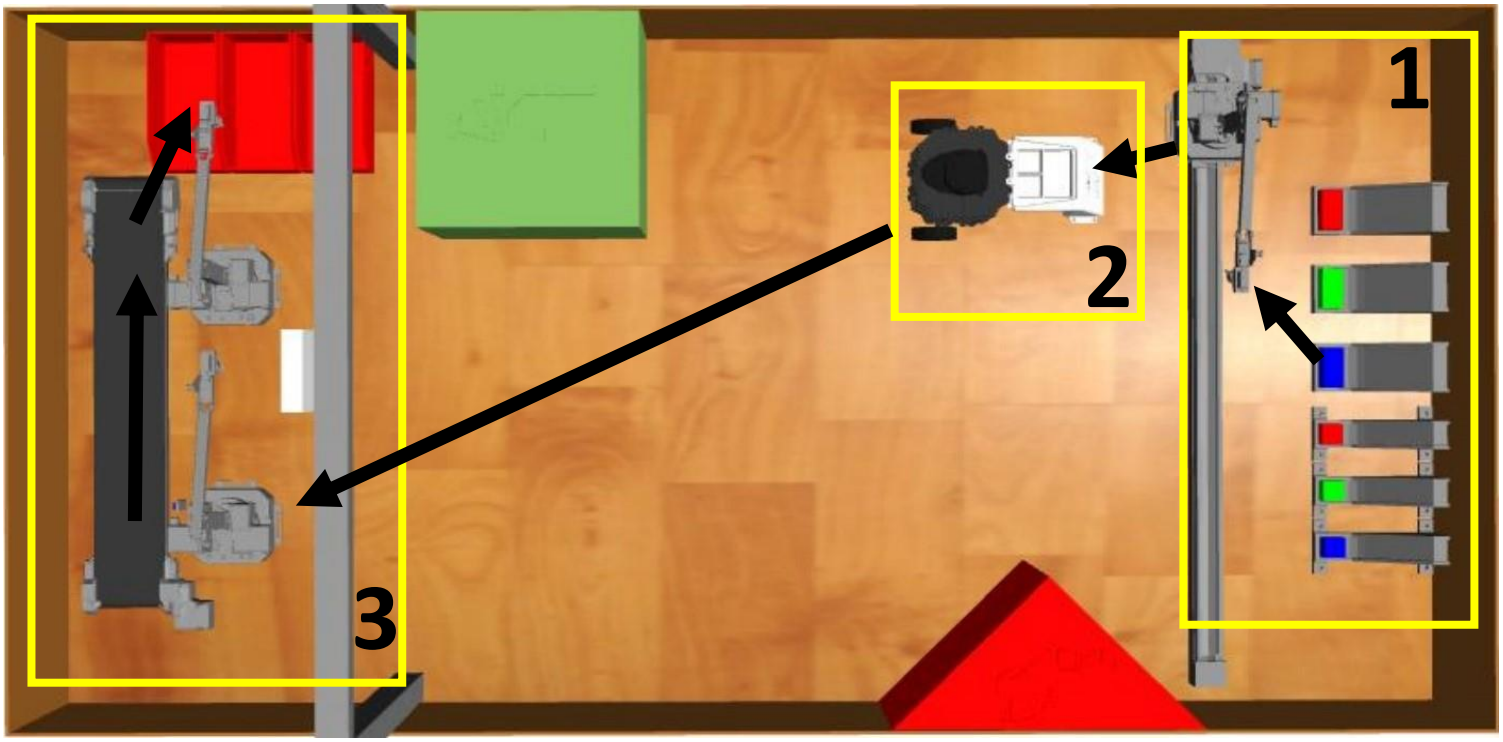
| Product Requirement | | System Requirement | | | |
|---|---|---|---|---|---|
| **ID** | **Title** | **ID** | **Description** | **Priority** | **Remarks** |
| F1 | Throughput | F1.1 | The throughput of the system shall be 2 cubes per minute. | Shall | Minimum throughput |
| | | F1.2 | The throughput of the system will be 6 cubes per minute. | Will | Desired throughput |
| F2 | Reliability | F2.1 | Objects A and B shall only go in their respective slots on the transport platform. | Shall | |
| | | F2.2 | The objects shall not be stacked upon one another. | Shall | |
| | | F2.3 | Objects shall not be dropped. | Shall | |
| | | F2.4 | The system shall operate event driven. | Shall | As opposed to being time driven. |
| F3 | Customer Satisfaction | F3.1 | The system shall keep track of all moving objects. | Shall | Every part of the system shall know at any point in time which object is where and can thus assure that it doesn't end up in the wrong delivery box. |
| F4 | Emergency Stop | F4.1 | The system must stop all movements when the emergency stop button is pressed. | Must | |
| | | F4.2 | The system must not drop any objects. | Must | Vacuum gripper remains active. |

Karlsruhe Institute of Technology

## Non-Functional Requirements

| Product Requirement | | System Requirement | | | |
|---|---|---|---|---|---|
| **ID** | **Title** | **ID** | **Description** | **Priority** | **Remarks** |
| NF1 | Easy usability of the system | NF1.1 | We will provide a customer-friendly user interface. | Will | |
| | | NF1.2 | We may provide a graphical user interface. | May | |
| NF2 | Portability of the System | NF2.1 | We shall use ROS as a coding platform, so the system will be easily portable. | Shall | |
| | | NF2.2 | Embedded systems shall not be modified. | Shall | E.g. we shall not install additional packages on the TurtleBot. |
| NF3 | Central settings area | NF3.1 | All settings shall be loaded by a single roslaunch file. | Shall | |
| | | NF3.2 | All settings shall be contained in a single folder. | Shall | |
| | | NF3.3 | Settings may be contained in a single file or split into multiple files. | May | Multiple files may be clearer than a single large file. |
| NF4 | Robustness against external interferences | NF4.1 | The system shall be able to resist against external interference. | Shall | |
| NF5 | Optimized routes | NF5.1 | The TurtleBot should follow the optimal path. | Should | |
| | | NF5.2 | The TurtleBot may deviate by up to 10% from the optimal path. | May | |
| NF6 | Efficient use of sensors and actuators | NF6.1 | The system should try to minimize the number of roundtrips of the TurtleBot. | Should | |
| | | NF6.2 | The subsystems shall enter a low power state if possible. | Shall | E.g. the conveyor belt shall stop moving if no object is placed upon it |
| NF7 | Scalability | NF7.1 | The system shall be developed using ROS which assures scalability. | Shall | |
| NF8 | Reusability | NF8.1 | Documentation shall be provided. | Shall | |

# Material Flow Description




1. uArm1 gets objects from slides

    a) TurtleBot moves to loading bay / calibrates its position.
    b) uArm1 slides in front of the desired slide.
    c) uArm1 gets object and slides back in front of the TurtleBot.
    d) uArm1 puts object on the TurtleBot.
    e) If there are more objects left in the order and TurtleBot isn't full yet: repeat from step b).

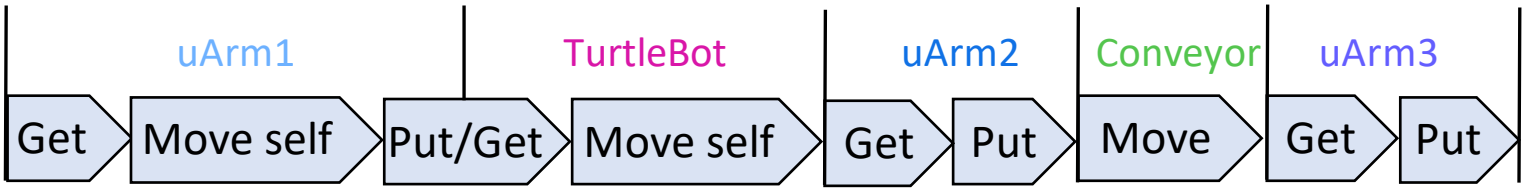
2. TurtleBot delivers object to collection area

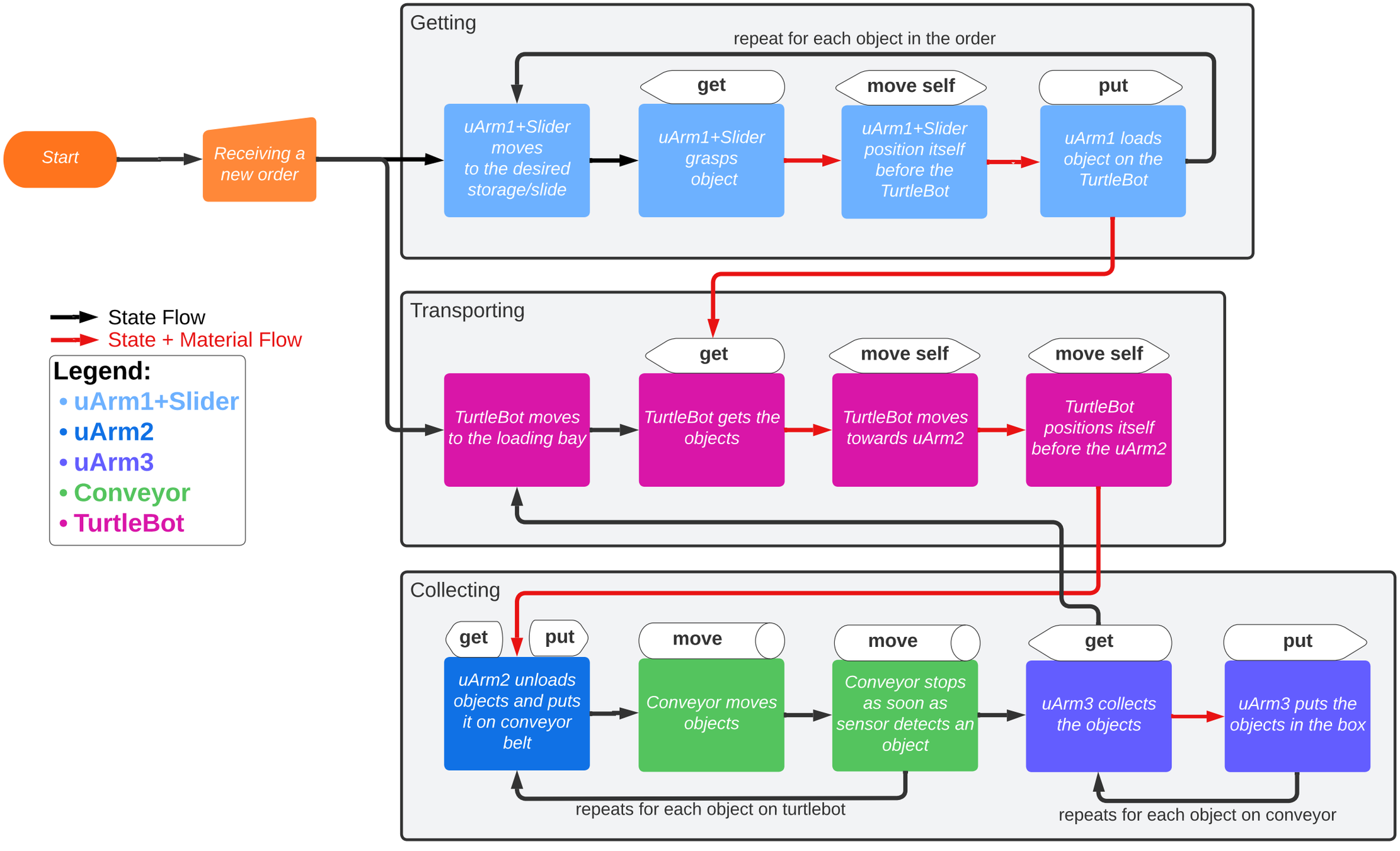
    a) TurtleBot moves itself to the uArm2.
    b) TurtleBot positions itself in front of uArm2.

3. Objects are collected and stored in the boxes

    a) uArm2 takes object from TurtleBot.
    b) uArm2 puts the object on the conveyor.
    c) Conveyor moves the object.
    d) uArm3 takes object from conveyor and puts it into the desired box.
    e) If there are more objects on the TurtleBot: repeat from step a)

## Material Flow based on Modular Material Handling (MMH):




Start
Receiving a new order
Getting
repeat for each object in the order
uArm1+Slider moves to the desired storage/slide
get
uArm1+Slider grasps object
move self
uArm1+Slider position itself before the TurtleBot
put
uArm1 loads object on the TurtleBot
State Flow
State + Material Flow
Legend:
• uArm1+Slider
• uArm2
• uArm3
• Conveyor
• TurtleBot
Transporting
TurtleBot moves to the loading bay
get
TurtleBot gets the objects
move self
TurtleBot moves towards uArm2
move self
TurtleBot positions itself before the uArm2
Collecting
get
put
uArm2 unloads objects and puts it on conveyor belt
move
Conveyor moves objects
move
Conveyor stops as soon as sensor detects an object
get
uArm3 collects the objects
put
uArm3 puts the objects in the box
repeats for each object on turtlebot
repeats for each object on conveyor

Seamless Engineering - Use case diagram (MS1)

Group 6