Quick heads up to get ready for our first lab

- Complete up to Checkpoint #1 before class
- Arrive laptop-ready

Prerequisites

CS3100 or equivalent is required

Honor system for now, but with a late verification happening next week

If you do not have meet prereq, drop the course or you can talk to me

Now videos...

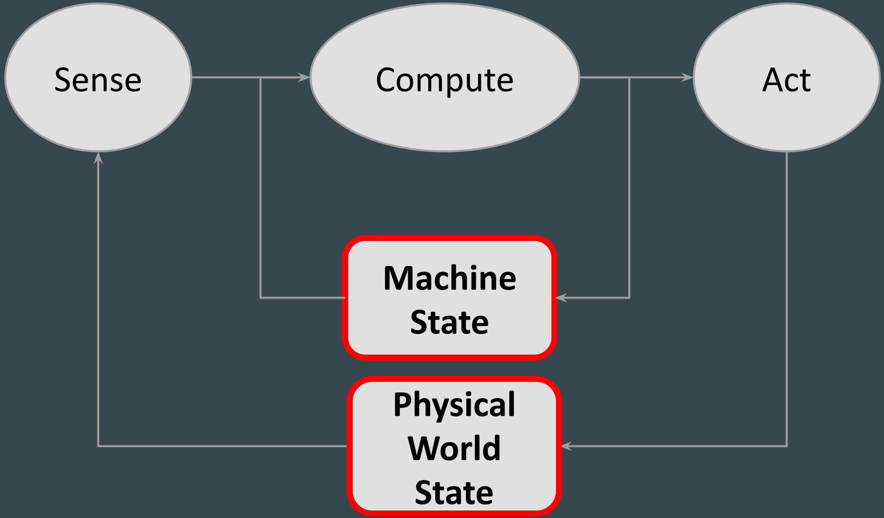
CS4501
Robotics for Soft Eng

•••

Robot Development - Cyber and Physical

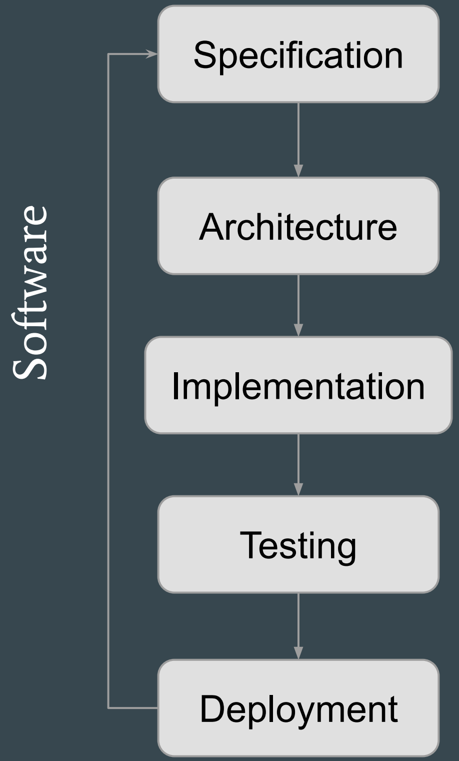
WIRED

www.amazonrobotics.com

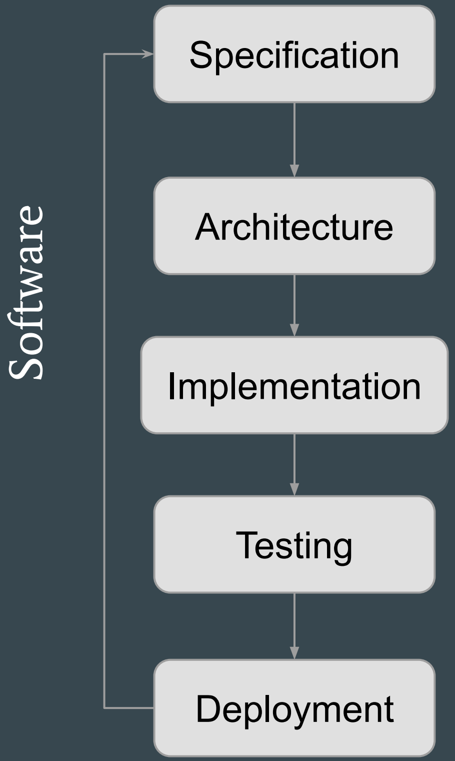


What does it sense?
What does it compute?
How does it act?
Machine state vs world state?

Development lifecycle



Development lifecycle



Specification relative to the World

Robotics

Physical attributes

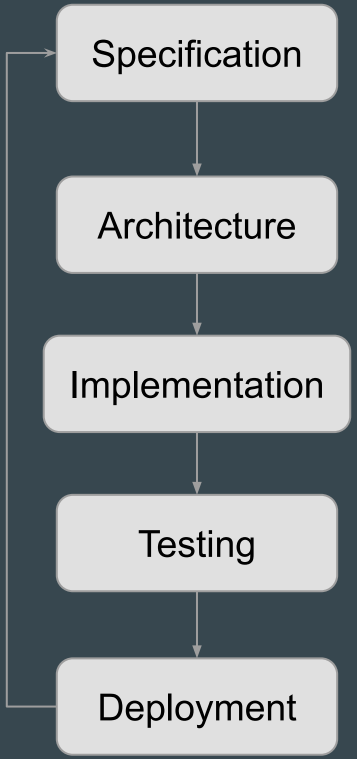
- *Size: 40” x 40” X 6”*
- *Lift 500 pounds*
- *Speed up to 2 mph*

World Behaviors

- *Follow ground markers*
- *No crashes against stationary objects*
- *...*

Development lifecycle

Software



Specification

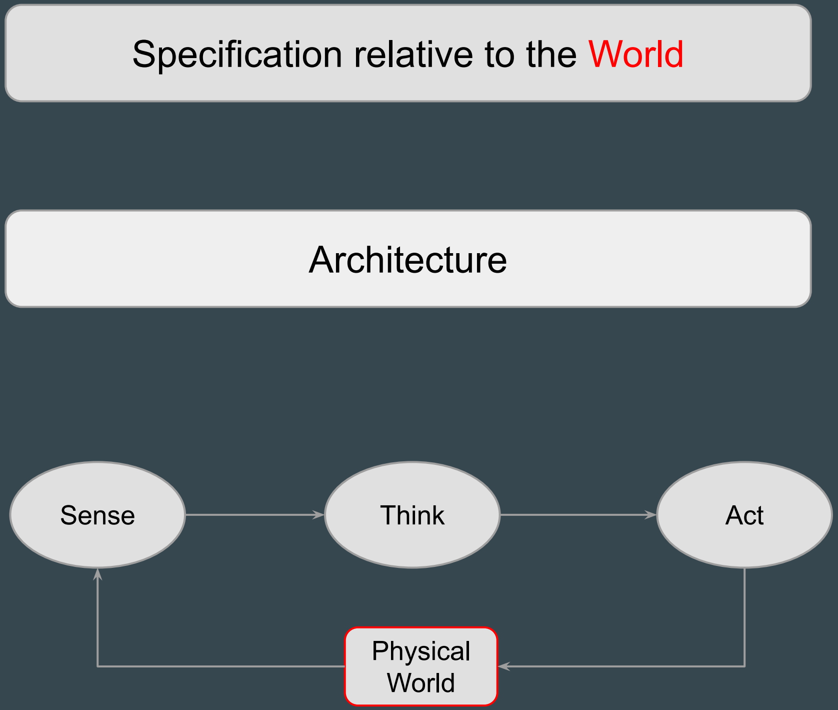
Architecture

Implementation

Testing

Deployment

Robotics



Specification relative to the World

Architecture

Sense

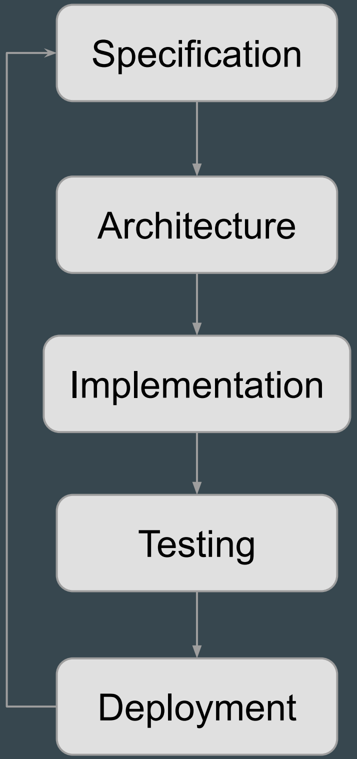
Think

Act

Physical World

Development lifecycle

Software



Specification

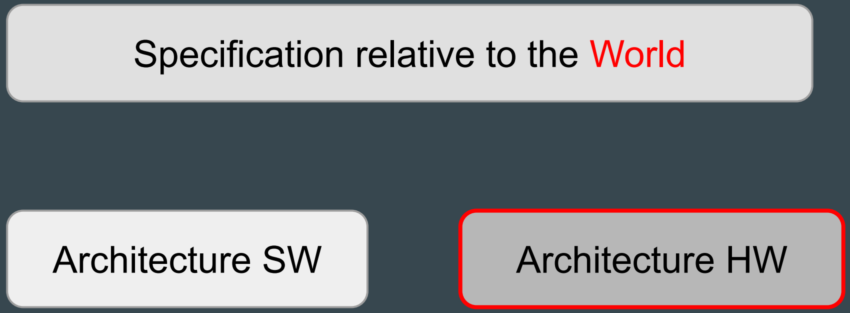
Architecture

Implementation

Testing

Deployment

Robotics

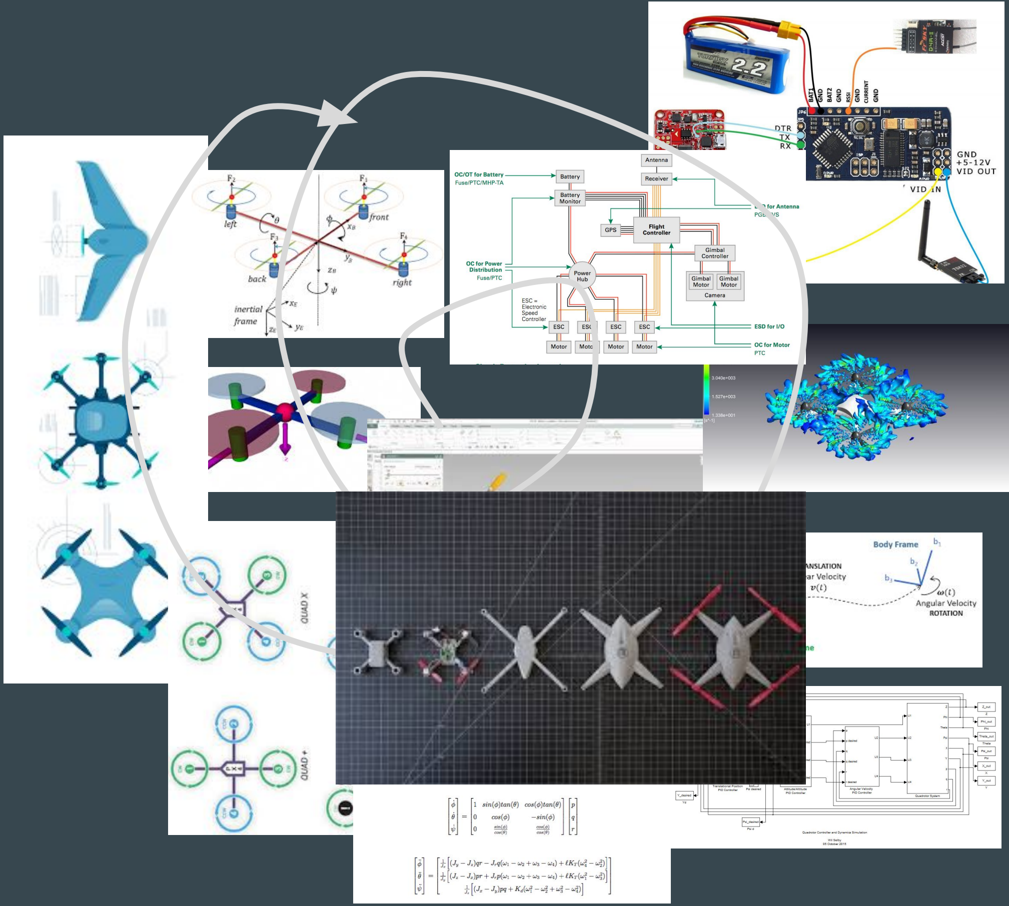


Specification relative to the World

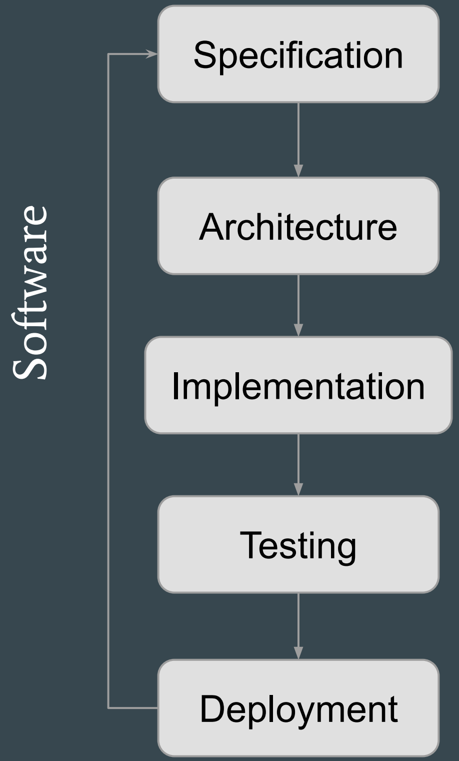
Architecture SW

Architecture HW

HW Architecture & Design



Development lifecycle



Software

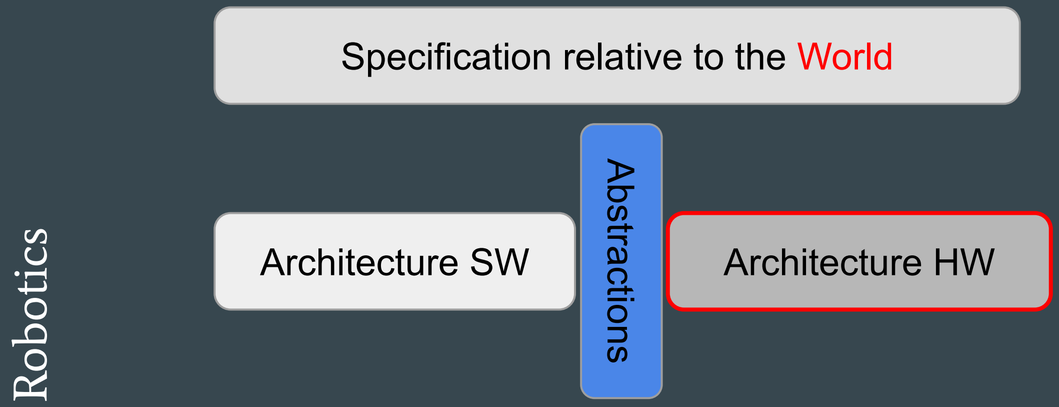
Specification

Architecture

Implementation

Testing

Deployment



Robotics

Specification relative to the World

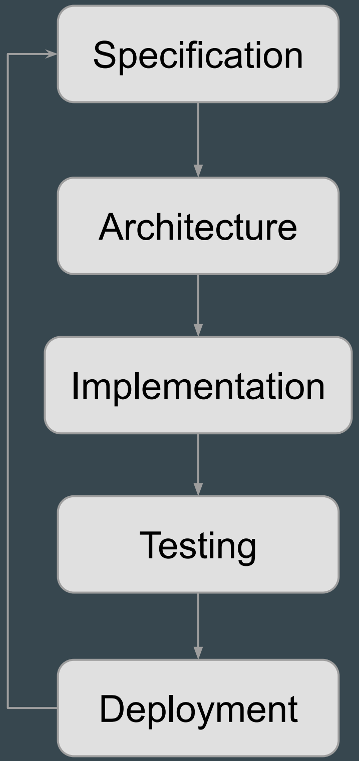
Architecture SW

Abstractions

Architecture HW

Development lifecycle

Software



Specification

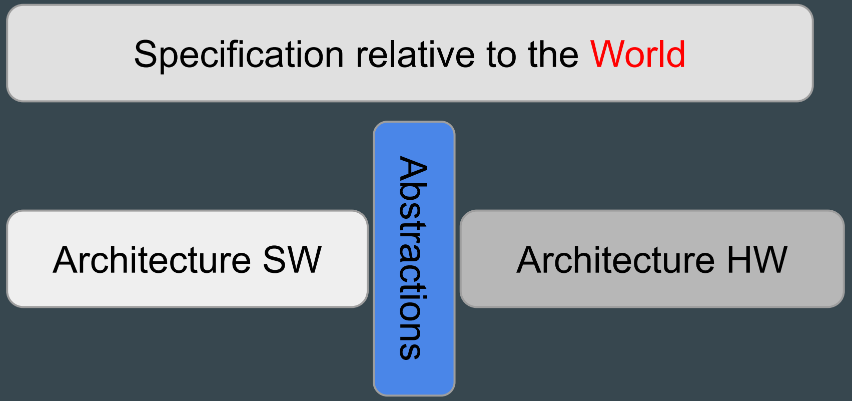
Architecture

Implementation

Testing

Deployment

Robotics

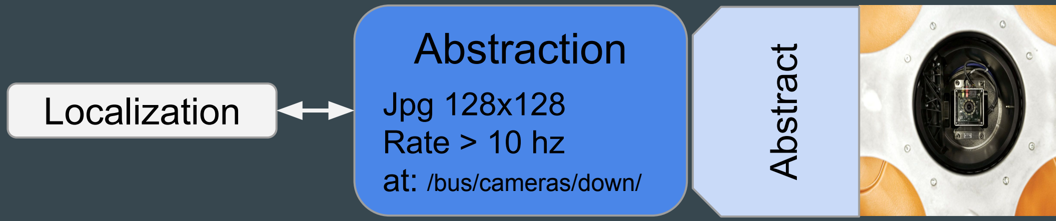


Specification relative to the World

Architecture SW

Abstractions

Architecture HW



Localization

Abstraction

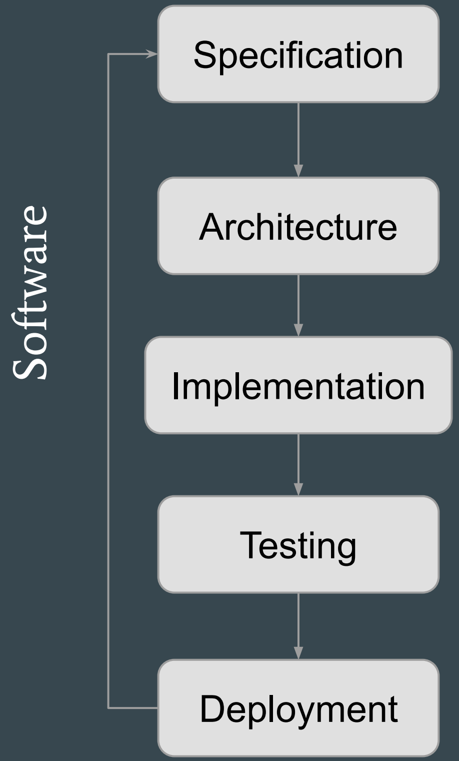
Jpg 128x128

Rate > 10 hz

at: /bus/cameras/down/

Abstract

Development lifecycle



Software

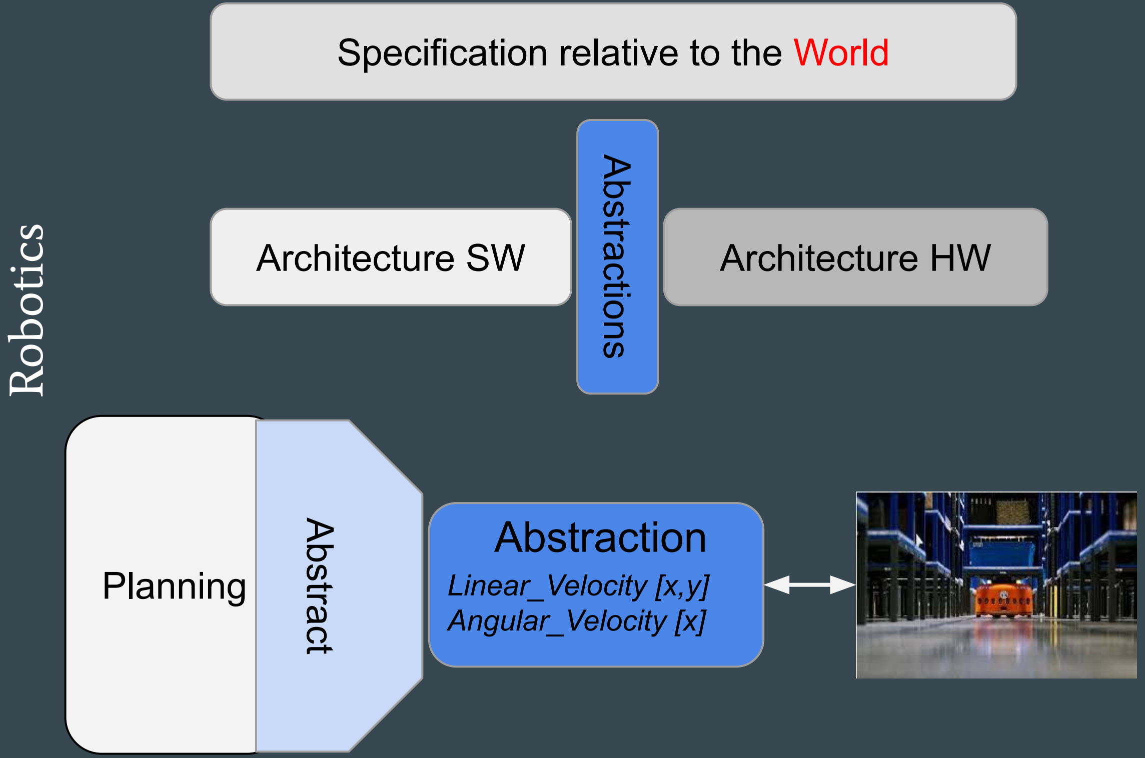
Specification

Architecture

Implementation

Testing

Deployment



Robotics

Specification relative to the World

Architecture SW

Abstractions

Architecture HW

Planning

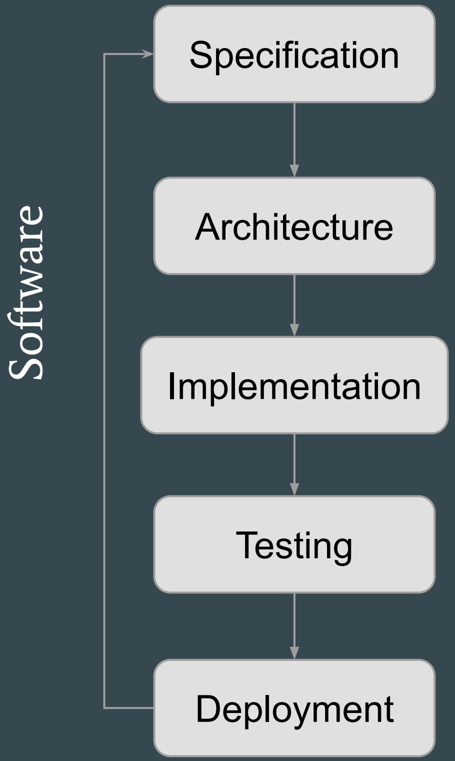
Abstract

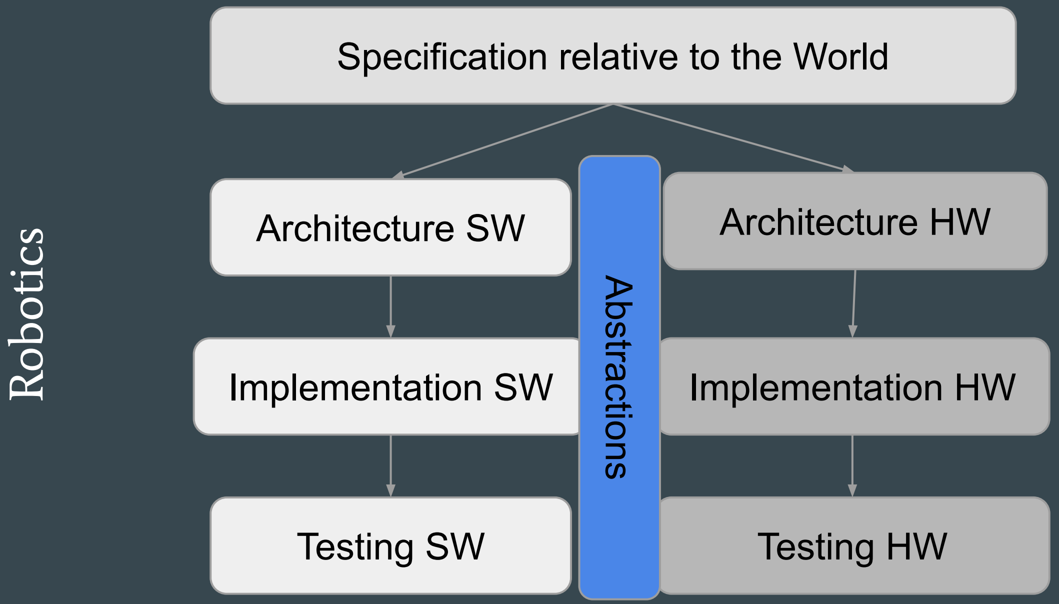
Abstraction

Linear_Velocity [x,y]

Angular_Velocity [x]

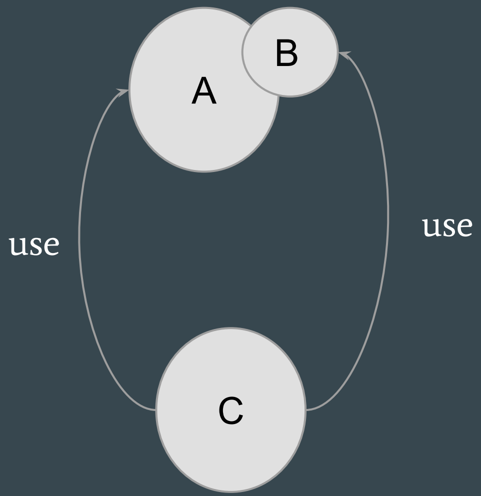
Development lifecycle



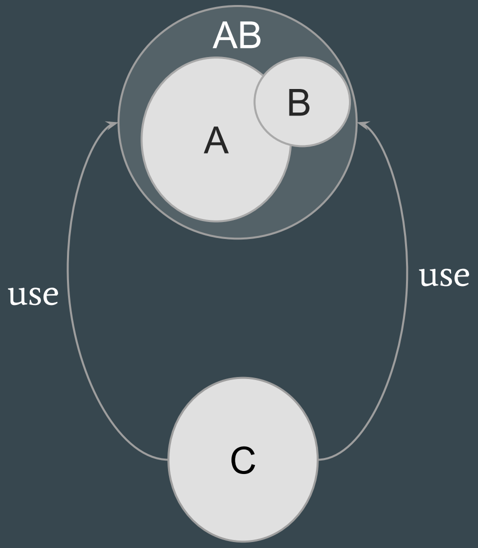


Law of Leaky Abstractions - G. Kiczales

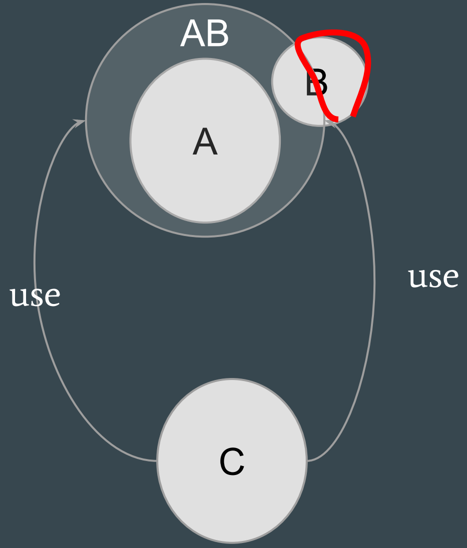
Noticeable between Cyber to Physical



C needs to understand
A and B



C needs to understand
abstraction AB



C needs to understand
abstraction AB + Leak

Law of Leaky Abstractions - J. Spolsky examples

- *Iterating direction on a 2D array does not matter*
- *Accessing virtual memory has a constant speed*
- *SMB are the same as local file*
- *SQL query with “where a=b and b=c and a=c” = “where a=b and b=c”*
- *VMs emulate an OS like it's running on real hardware*

What is leaking?

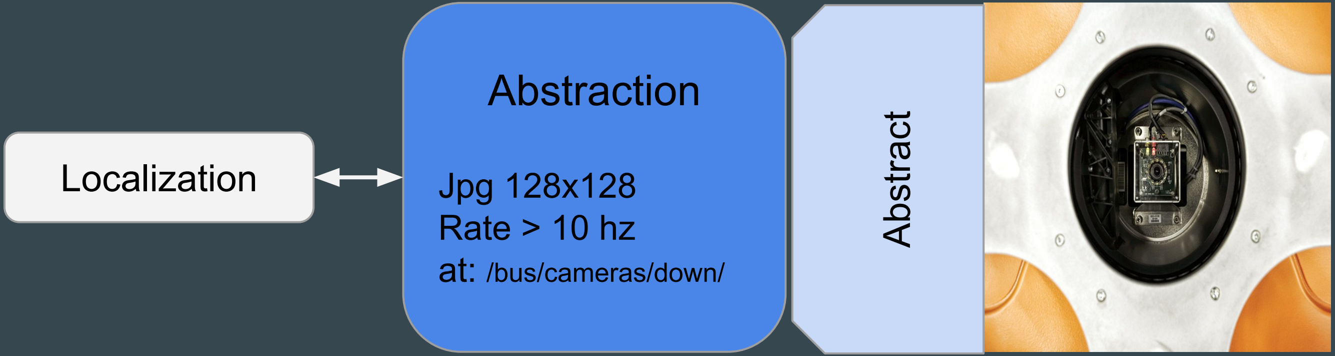
Law of Leaky Abstractions

- Abstractions makes us more efficient, until they leak
- All good abstractions leak
  - They have exceptional behaviors
  - They break underlying assumptions

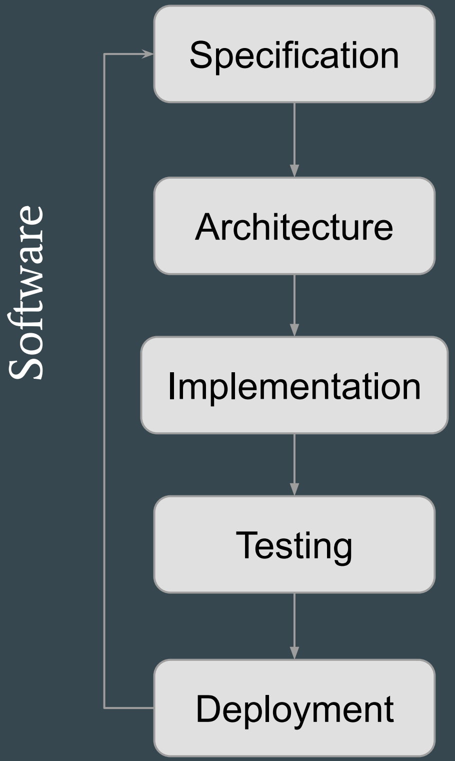
Abstractions of sensors, actuators, states, and comms are particularly Leaky!

Law of Leaky Abstractions

Where could it leak?



Development lifecycle



Software

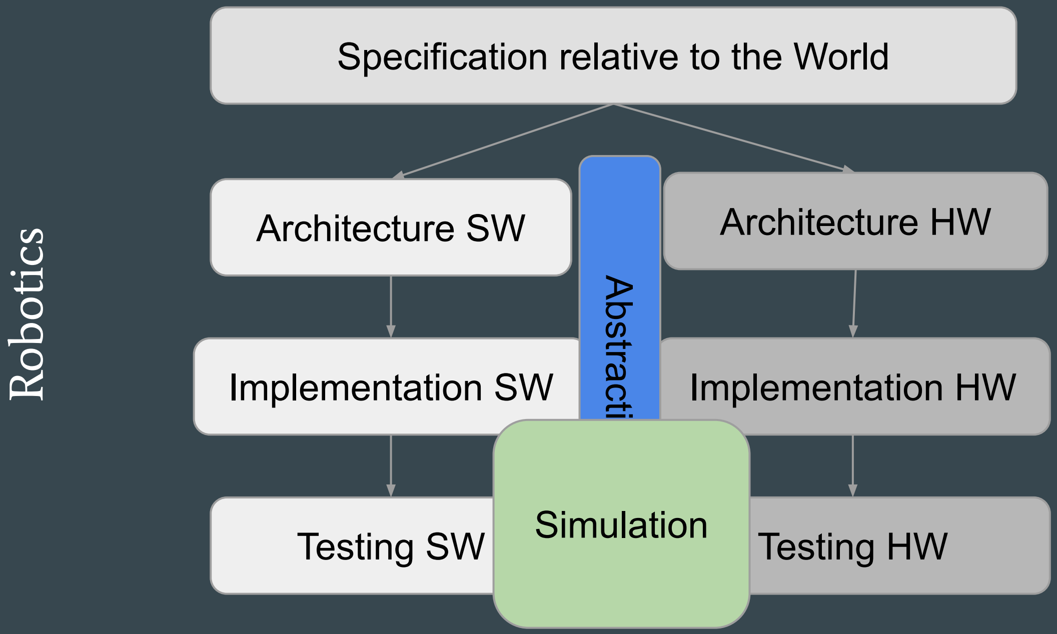
Specification

Architecture

Implementation

Testing

Deployment



Robotics

Specification relative to the World

Architecture SW

Architecture HW

Abstracti

Implementation SW

Implementation HW

Testing SW

Simulation

Testing HW

Simulation in Robotics

Developing Software

- Mock when
  - Relying on other components
    - Not available yet
    - Too complex/expensive
    - Functions to Databases

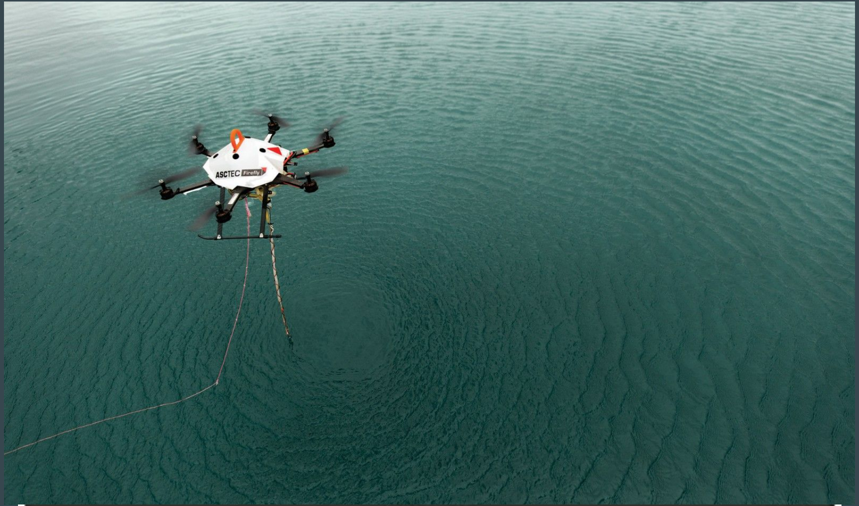
Developing Robots

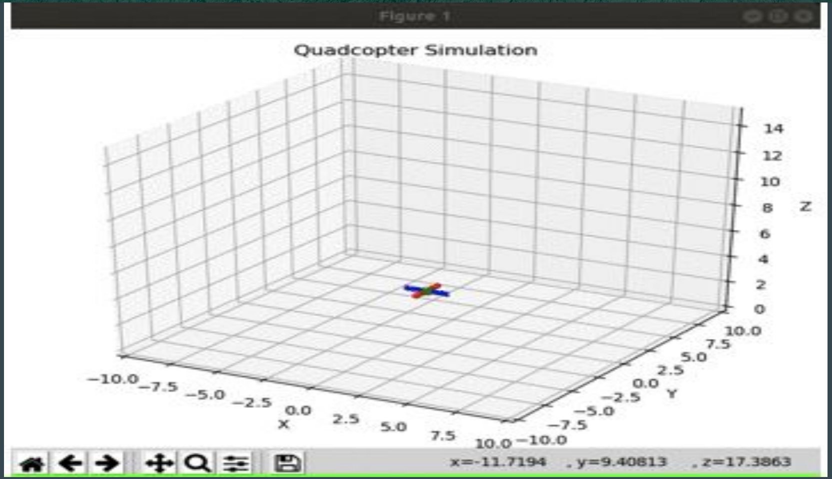
- Mock when
  - Relying on world
    - Too complex
    - Failures too expensive
  - Relying or other components
    - Sensors
    - Actuators
    - …
    - Software

Simulation in Robotics

Test hovering functionality

What do you mock?

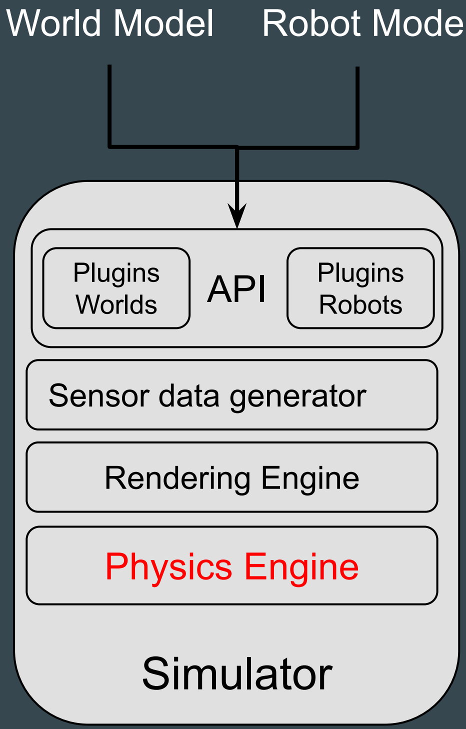


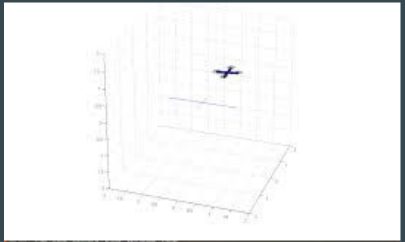


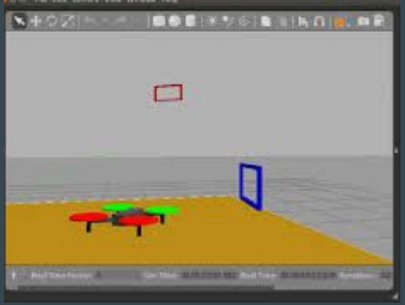
x=-11.7194 , y=9.40813 , z=17.3863

Simulation in Robotics

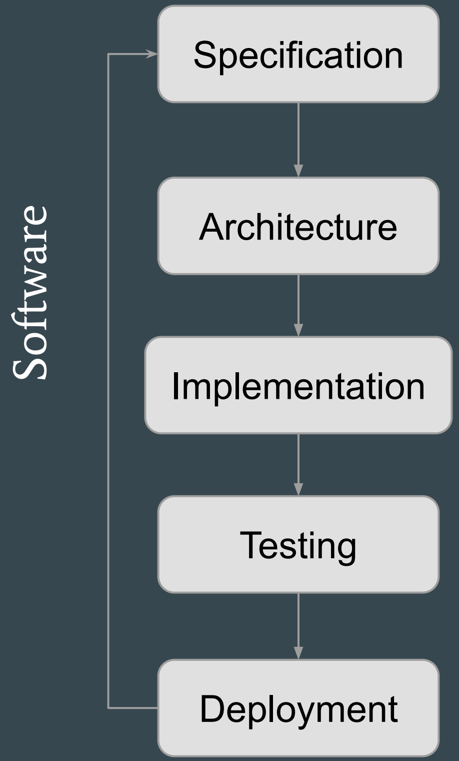
- Applications
  - Design Exploration
  - Testing
  - For SW/HW/Both
  - Training data
- Benefits
  - Accelerates development
  - Low cost







Development lifecycle



Software

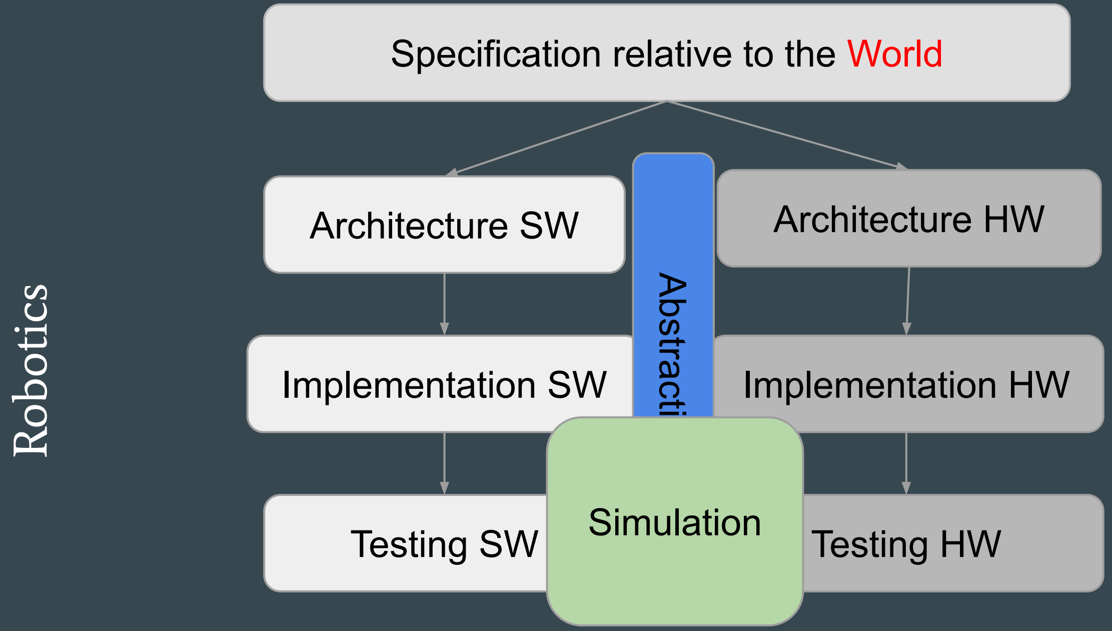
Specification

Architecture

Implementation

Testing

Deployment



Robotics

Specification relative to the World

Architecture SW

Architecture HW

Abstracti

Implementation SW

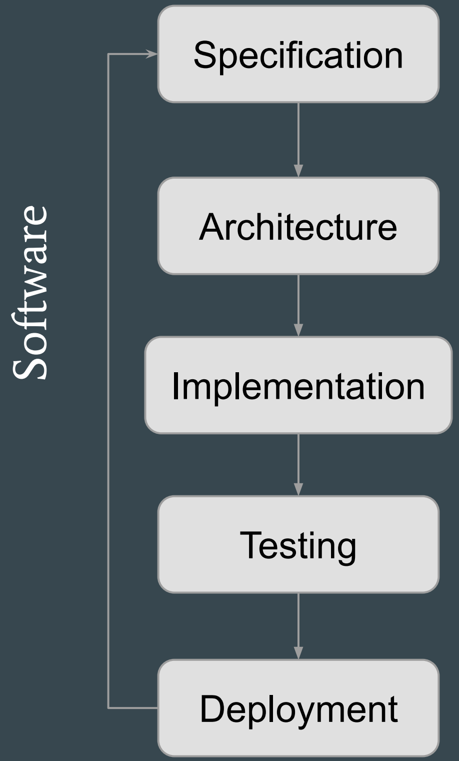
Implementation HW

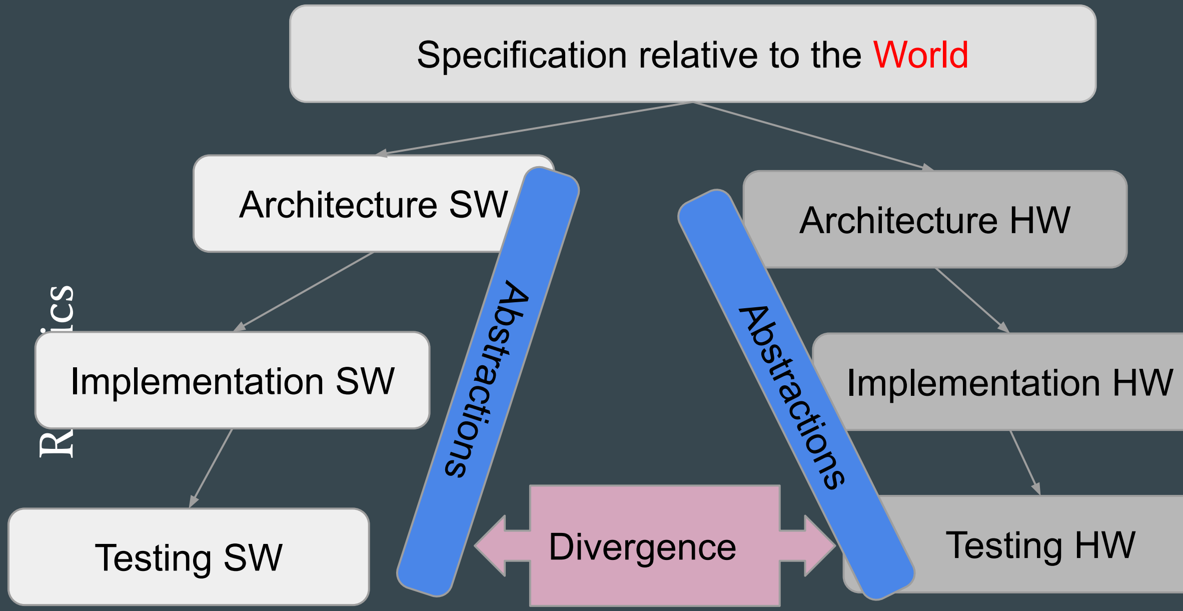
Testing SW

Simulation

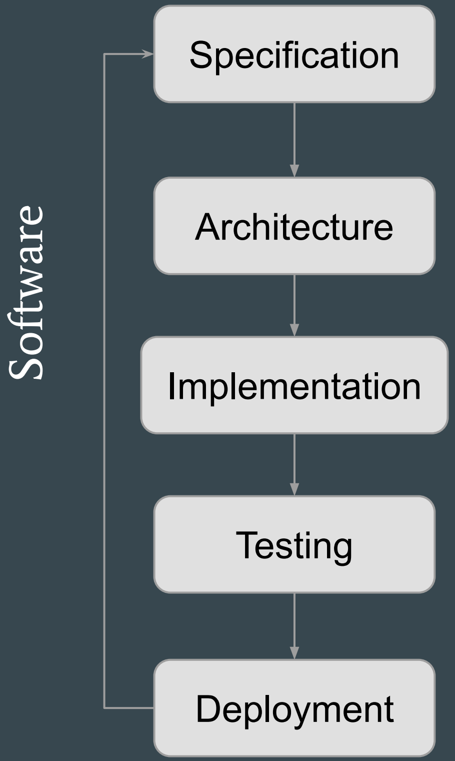
Testing HW

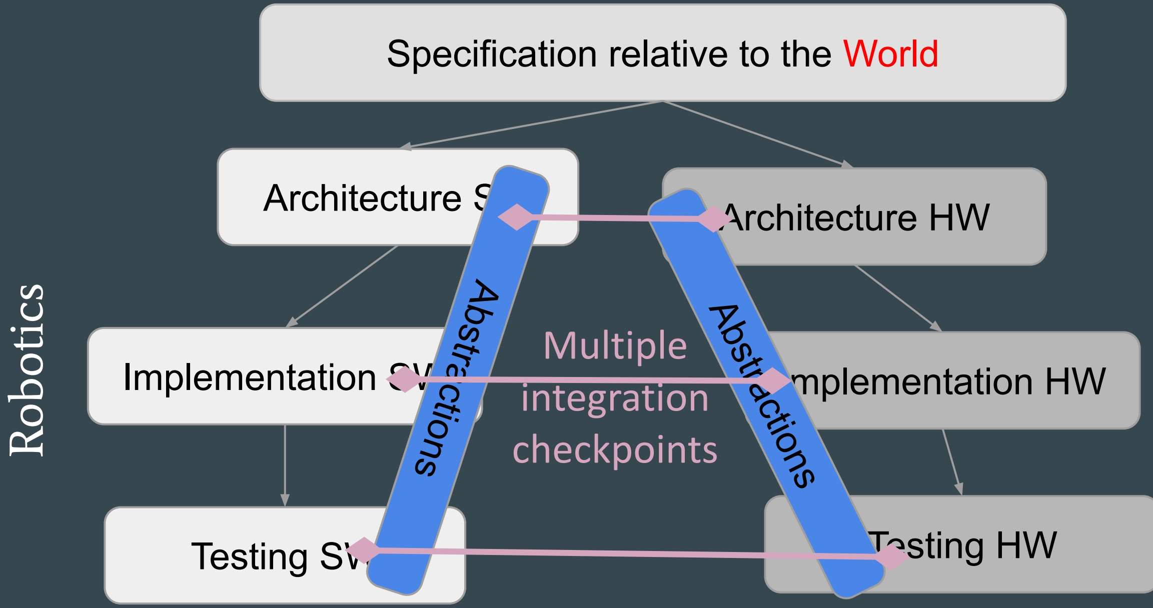
Development lifecycle



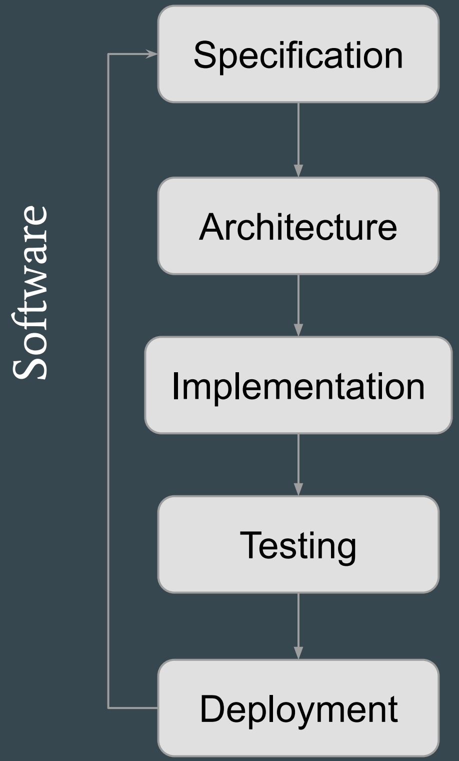


Development lifecycle





Development lifecycle



Software

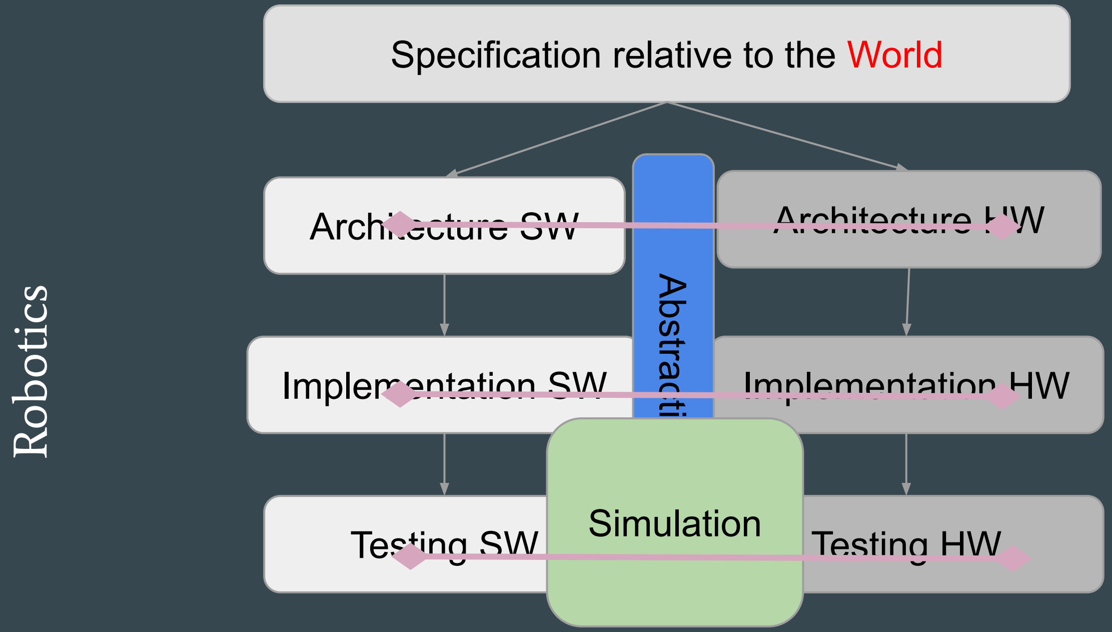
Specification

Architecture

Implementation

Testing

Deployment



Robotics

Specification relative to the World

Architecture SW

Architecture HW

Abstracti

Implementation SW

Implementation HW

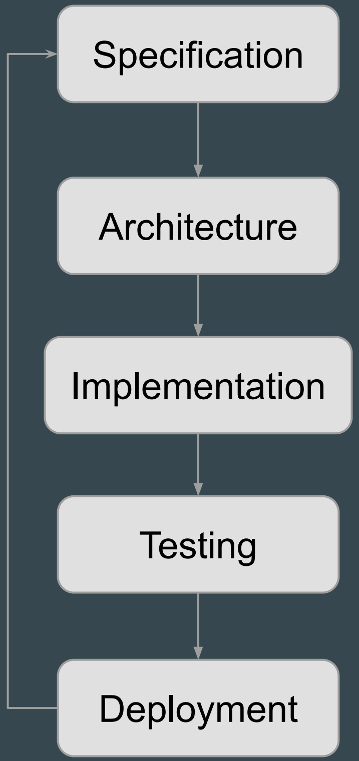
Testing SW

Simulation

Testing HW

Development lifecycle

Software



Specification

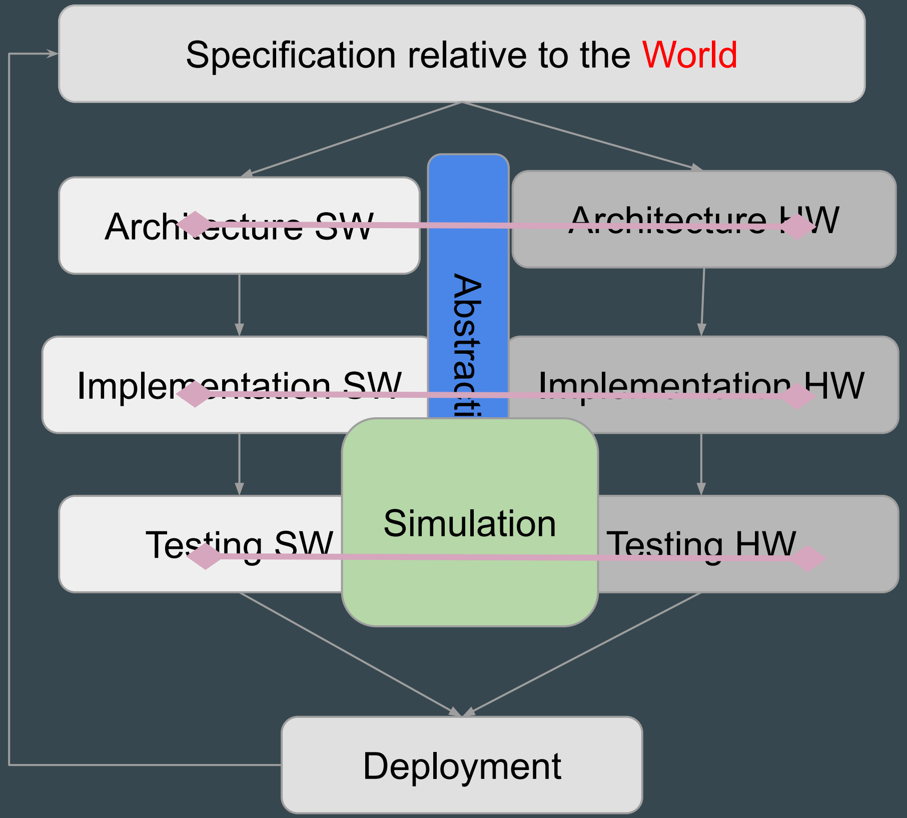
Architecture

Implementation

Testing

Deployment

Robotics



Specification relative to the World

Architecture SW

Architecture HW

Abstracti

Implementation SW

Implementation HW

Simulation

Testing SW

Testing HW

Deployment

Robot deployment

- Define acceptable initial states
- Multiple distributed processes
- Thousands of configuration parameters
- Optimization for scenarios

Programming the deployment

Robot development lifecycle

- Physical Requirements
- Multi-level and likely leaky Abstractions
- Parallelize synchronized SW/HW development
- Simulation is key tool
- Decomposition is interleaved with discovery
- Highly-multidisciplinary
  - Richer vocabulary
  - Higher opportunity for innovation
  - Higher opportunity for breakdowns

More Complex Development Process

Next - First Lab

- Complete up to Checkpoint #1 before class
- Arrive laptop-ready