ECSE-211
Design Principles and Methods

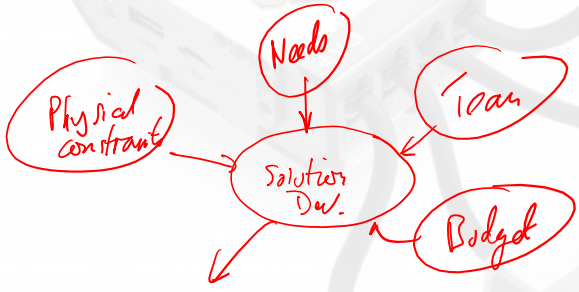
Lecture 7: Validation – Designing and Implementing Tests

Date: 1 February 2023

1

Lecture 6 - Questions

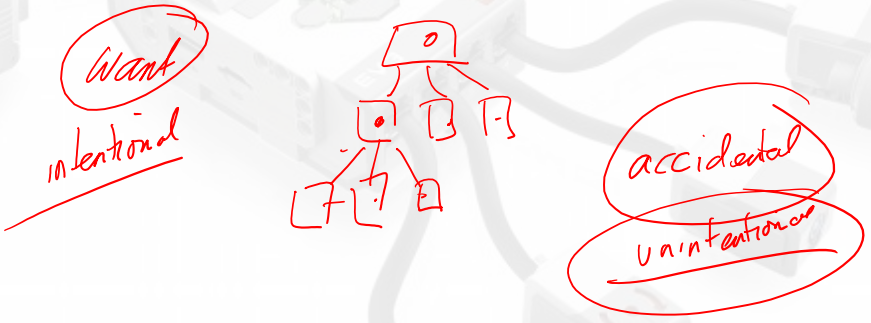
- Question 1
- What are the main inputs to solution development?



2

Question 2

- Why is it important to identify possible interactions between components early in the process?



3

Question 3

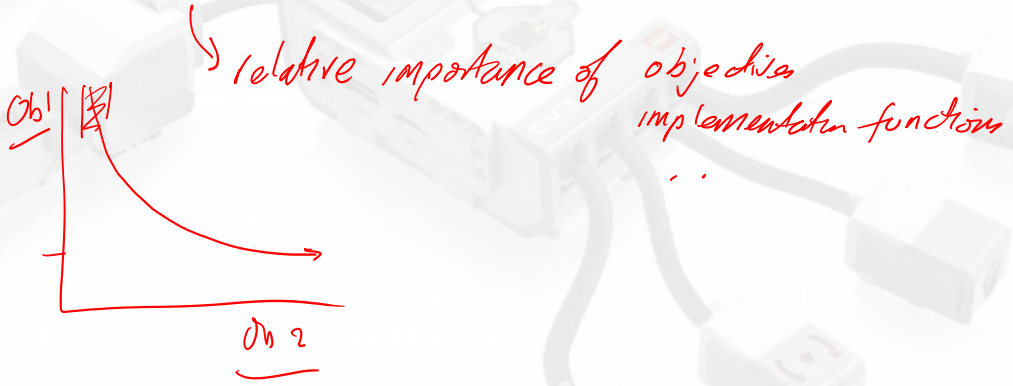
- What is a Tolerance and why is it important?

having a range on a value
↳ conforms some of the solutions
↳ allows practical implementations

4

Question 4

- What is a Tradeoff? Why do Tradeoffs happen in Engineering Design?



5

The Rationale for more than One Candidate

- Having multiple possible choices is the basis of "Set Based Design"
- With 3 potential candidates, each can be evaluated against the System Model and the Requirements Document
- The evaluation should include estimates of the "best chance of success" in the EDP
- One candidate is then chosen, both the system and the components, for further development
- NOTE – this process and the decisions together with the rationale for the decisions and any test results MUST be DOCUMENTED

6

A Candidate Decision

- Having chosen a solution for further development – do not throw away the others!

DO NOT THROW AWAY THE OTHER IDEAS AND WORK

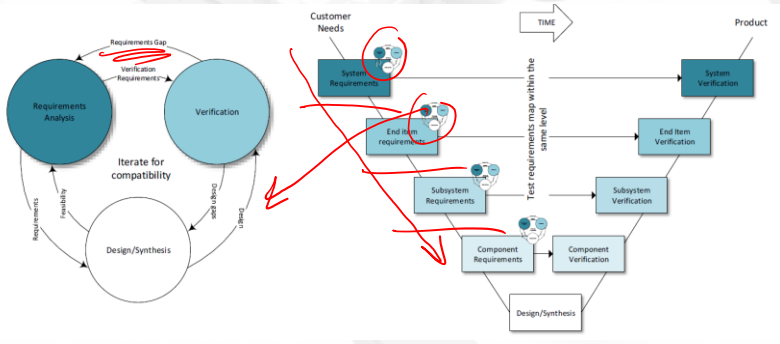
File the documents in case they are needed in the future

In a Design Process you need to be able to Restore a Previous State…

7

Validation

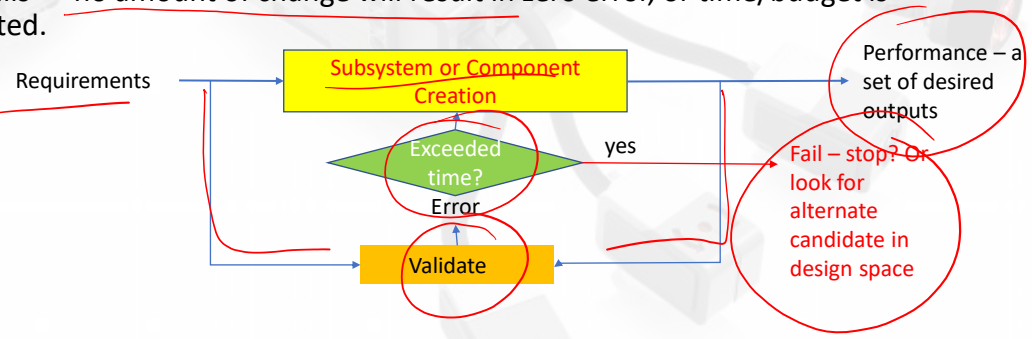
- The process described can only terminate through validation of each step
  - Each idea is tested against the Requirements and refined until the Requirements Gap is removed – or the idea is abandoned…



8

Validation

- The process of creating or "identifying" a potential candidate for a subsystem of component is iterative
- The process terminates when
  - It "validates" – zero error
  - It "Fails" – no amount of change will result in zero error, or time/budget is violated.



9

What is "Validation"?

- A Reminder:
  - *Validation* is an exercise largely conducted with external clients and stakeholders to provide assurance that a product meets (or will meet) the needs
  - *Verification* is an internal operation which evaluates whether a product, service or system complies with a requirement (which may be a regulation, a specification, etc.)
- The method by which these are implemented is "*Testing*"

10

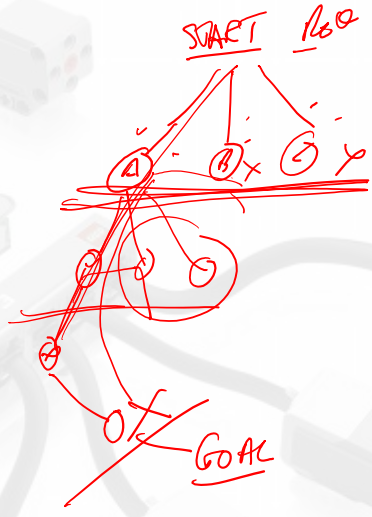
Previous Lecture - Ideas

- Idea Generation
- Tolerances and Impact on Design Choices
- Multiple Possible Designs
- Documentation

11

Contents

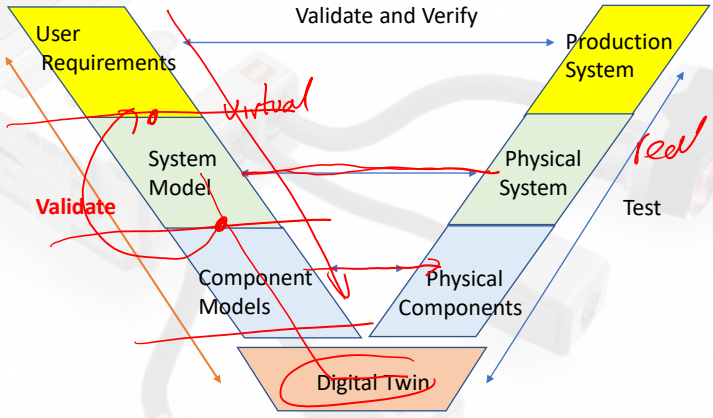
- Validation – a critical component
- What makes a good Test?
- Component and System Testing
- Implementing a Test
- Documenting a Test



12

Validation

- The part of the EDP which allows the process to progress
- At each step, the output(s) are validated against the inputs (Requirements)
- But how is validation done?



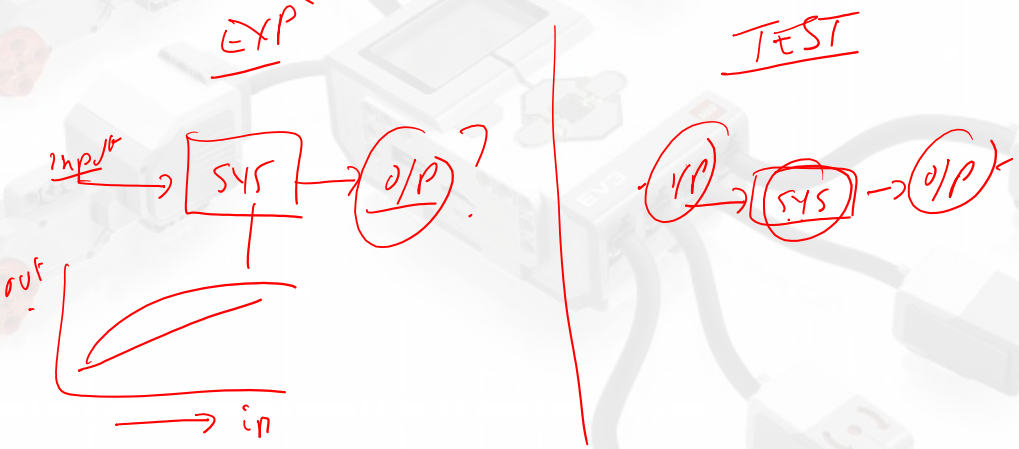
13

Validation

- Validation is a form of testing
- The goal is to determine if the concepts implemented through models perform to meet the requirements
- Testing is often related to physical systems
- For DPM (and design in general), validation and testing are considered equivalent

14

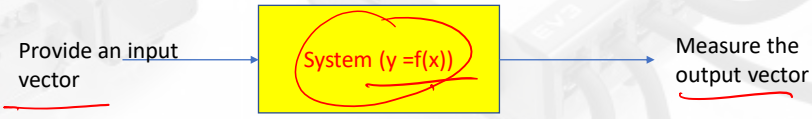
Experiments versus Tests?



15

What is a "Test"?

- Let's start with "what is an experiment"?
- Given a system, or plant, how does it behave?
- The behavior allows a Model to be constructed

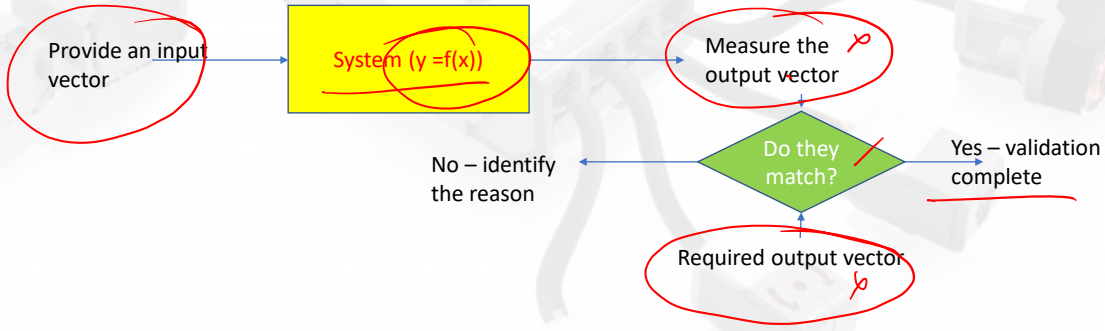


- When you need to get answers to questions, design experiments
  - How well can the robot localize itself?
  - What is the error in position keeping?
    - This defines *a tolerance* that can be achieved

16

What is a "Test"?

- A Test will determine if the System or Component actually behaves the way it was expected to.



17

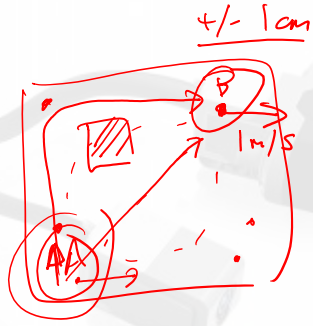
Testing

- What makes a good test?
  - Testing the performance of a component or a system should involve a coverage of the entire performance space
    - The performance space is essentially the response of the system to a particular stimulus – the input vector
  - There may be several performances, or outputs, for a given component
    - This the performance space could be multi-dimensional

18

Testing

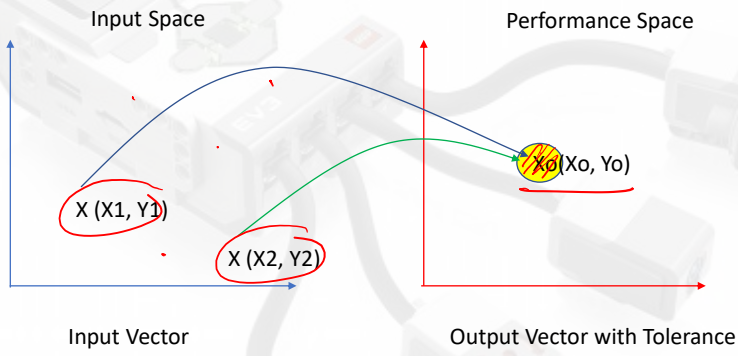
- What makes a good test?
  - For example, the input to a navigation system might be
    - The current position (in 2D or 3D space)
    - The current heading (as an angle in 2D or 3D space)
    - The current speed
  - The "performance" output might be:
    - A specified 2D or 3D position
    - A specified heading
    - A specified speed
    - Along with acceptable tolerances on all these



19

Testing

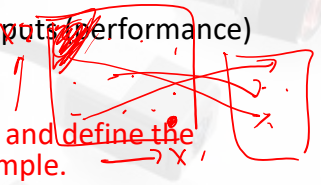
- Example for navigation:
- Independent of the starting point, the robot must arrive at the same ending point
- (for getting through a bridge…)



20

Testing

- Each component has a set of specifications or requirements that it should meet
- Each component has a set of inputs – which may be related to the performance (output) of a linked component
- To test the component (or system), a set of inputs needs to be considered which "cover" the input space
- The requirements for the component specify what the outputs (performance) must be to be acceptable
- So…
  - A good test should effectively sample the input space and define the expected performance (output) for each sample.
  - The performance must be met for the test to be considered acceptable



21

Testing

- Once component tests have been completed, the components can be integrated into the System
- Tests of a prototype serve to confirm the validity of the entire design
- Can the system fail if all the components work?

  Yes!

- Why or How?
  - Components can interact in ways not envisaged in the System Model
  - Example:
    - The navigation system works perfectly
    - The obstacle avoidance tests all passed
    - Now the two systems are integrated
    - The number of threads executing increases and the processor response reduces
    - Result – the motor thread misses commands and the motor spins out of control – navigation is broken!

22

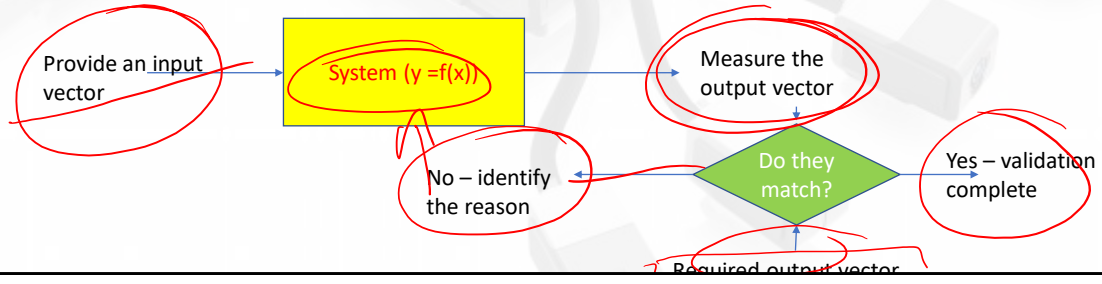
A Test Plan

- Developing and implementing an effective test plan is critical for the success of the EDP
- A test plan should include the tests to be completed to validate the performance of each component
  - If the test fails, then the component design needs to be reviewed and, potentially modified
- Once the component has been tested it can be integrated with other components and the partial system tested – so a test would need to be designed for the combination…
- Finally, the full system needs to be validated
- A test plan should contain the list of tests and a timeline for when they should be conducted.

23

The Structure of a Test

- A Test is created to Validate a particular stage of the EDP
- If it does not do this, it is of little use – in fact, it may have a negative impact.
- A Test is defined through a test document – which has a structure
- As with the other documents, there must be a header with a title, person responsible, etc.
- The main body of the document is structured around the test itself:



24

The Structure of a Test

- Contents of a Test Document
  - Statement of specific purposes of the test.
    - What design or implementation decisions will the test validate?
  - Specific test objectives
    - What needs to be evaluated or measured during the test?
  - A step-by-step procedure for conducting the test
    - Attention should be paid to the variables to be controlled or monitored
  - An outline of the expected results
    - A data sheet with the predicted data outcomes
    - If you don't know what should happen – how do you know if it is wrong?

25

Testing - Details

- So…
- A Test document should be created which describes each test
- In designing the input test vectors, they should cover the parameter space – in particular they need to address "edge cases" – points where there are likely to be large errors
  - E.g.
    - Does the ultrasonic sensor provide the readings predicted?
    - Can the robot find the goal, with an accuracy of positioning that is expected in the design for it to work?
- Edge cases can be identified from the experiments that are run and the modeling performed within the design process

26

Testing - Details

- Tests should be designed to check extreme cases of the specifications
  - E.g. what happens if the ambient light is extremely high?
  - What happens if the robot is traveling fast and hits a wall or an object?
  - What is the worst case situation that can be encountered during the operation of the robot?
- But – don't forget the routine cases…

27

A Test Document Example

- Goal: Determine if the robot can avoid obstacles
- Result: It does
- Actions: None

- Is this a good test document? No
- What is missing?

28

A Test Document

- *Date:* 11 February 2015
- *Tester:* John Smith
- *Author:* John Smith
- *Hardware version*: robot version 2.1
- *Software version:* 3.2
- *Goal:* Determine if the robot can avoid obstacles

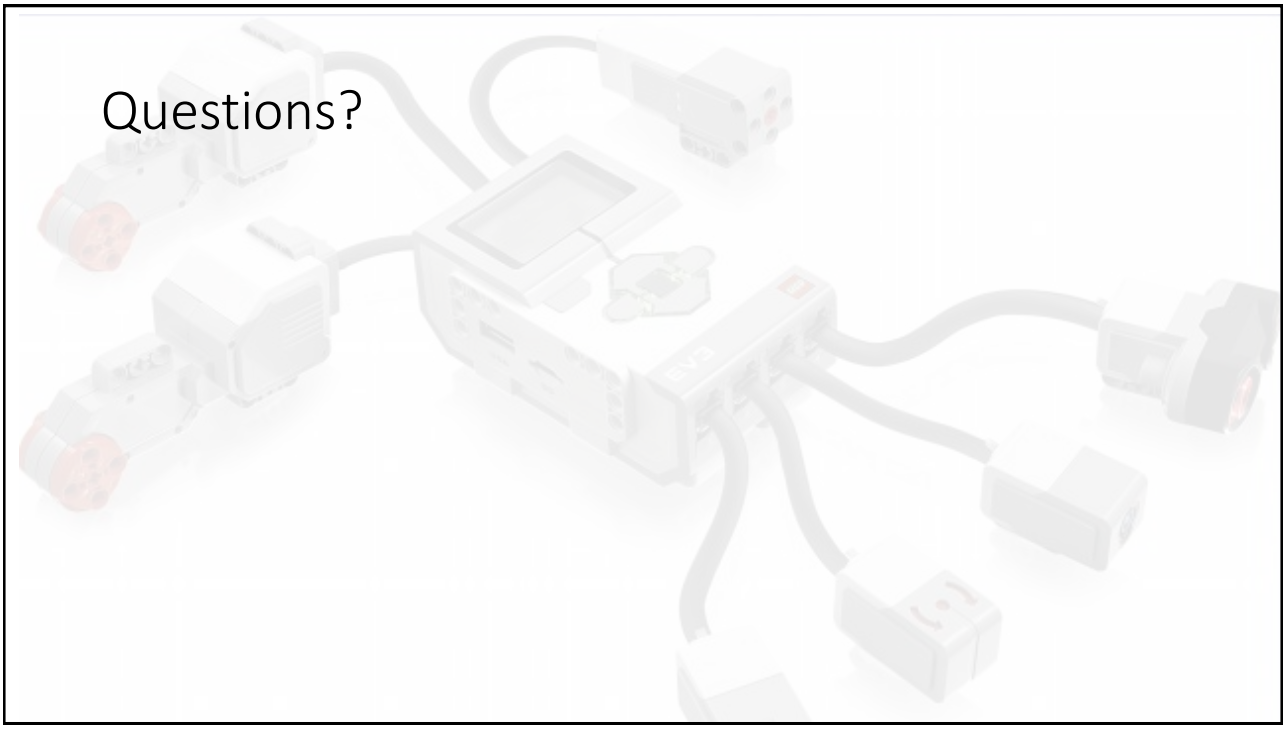
29

A Test Document

- *Procedure:* The robot should be placed at the origin of the grid facing North. Obstacles will be placed at grid locations (0,1), (1,0), (2,2). The robot will be instructed to travel to location (4,4). The test should be performed at least 10 times.
- *Expected Result:* The robot should identify any obstacles in its path and avoid them by at least 5cm. The robot LCD display should indicate that it has seen an obstacle. It should arrive at the destination with an accuracy of better than 5 cm.

30

Questions?



31