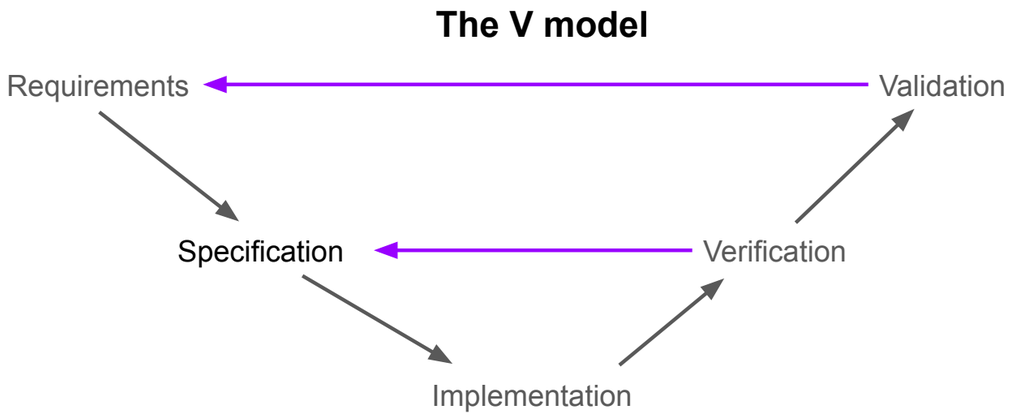
COM4506/6506: Testing and Verification in Safety Critical Systems

Dr Ramsay Taylor

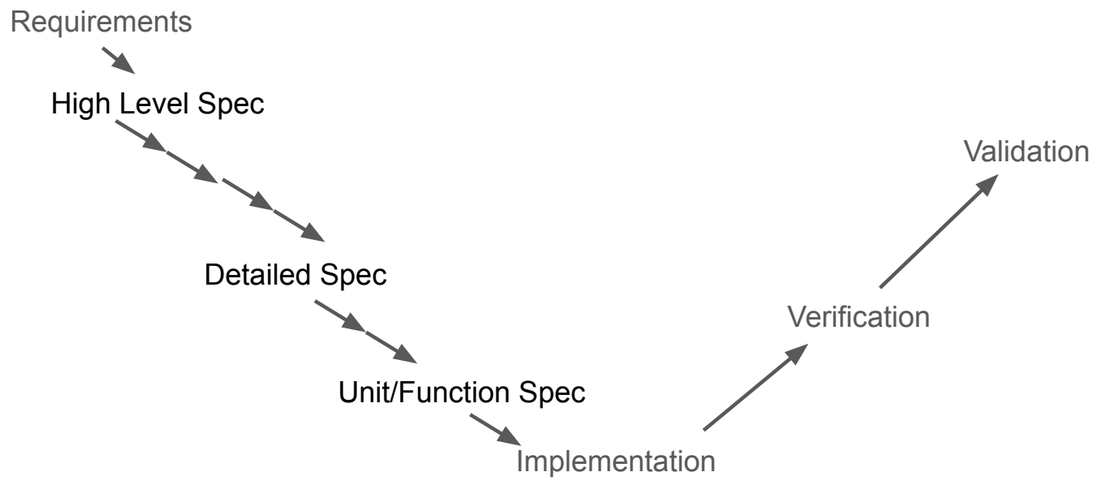
Contents

- When does Unit Testing stop being Unit Testing?
- Why does this change anything?
- How does this fit with our overall development?

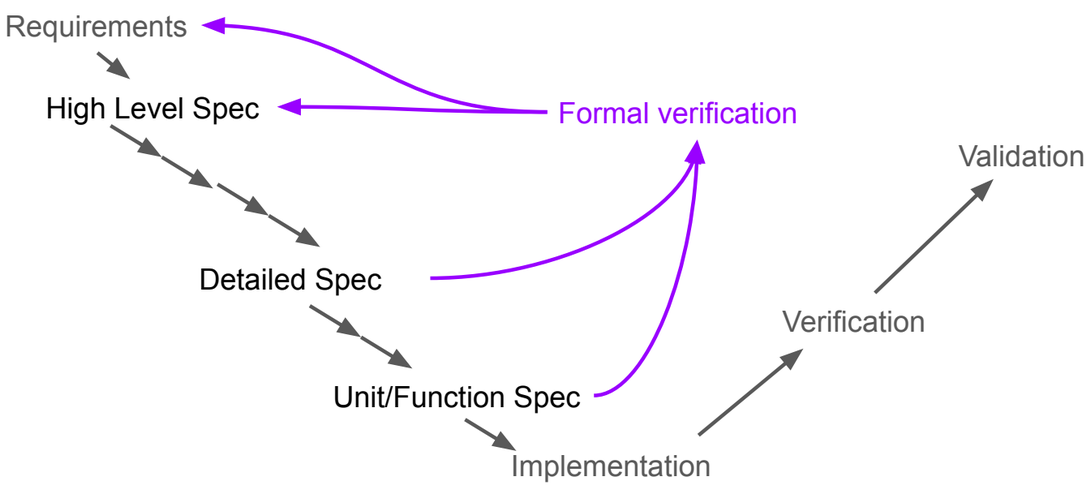
The other side of the V



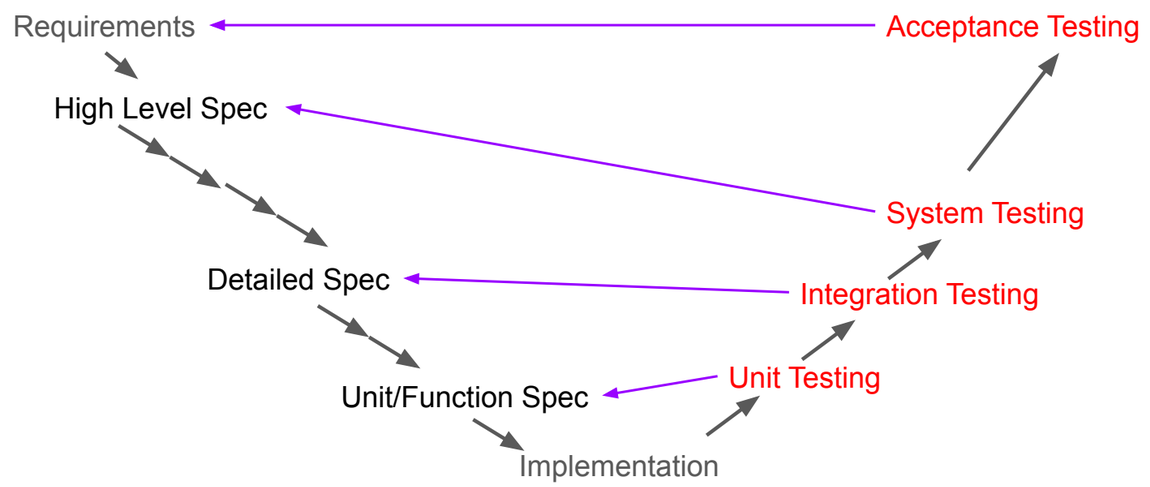
The other side of the V



The other side of the V

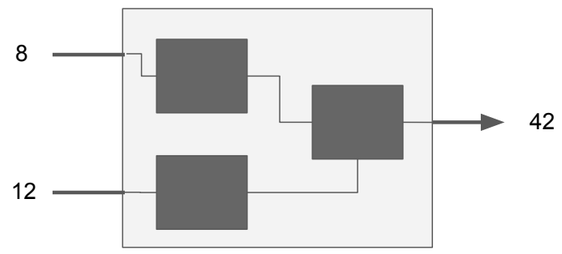


The other side of the V



When is Unit Testing actually Integration Testing?

When is Unit Testing actually Integration Testing?



When is Unit Testing actually Integration Testing?

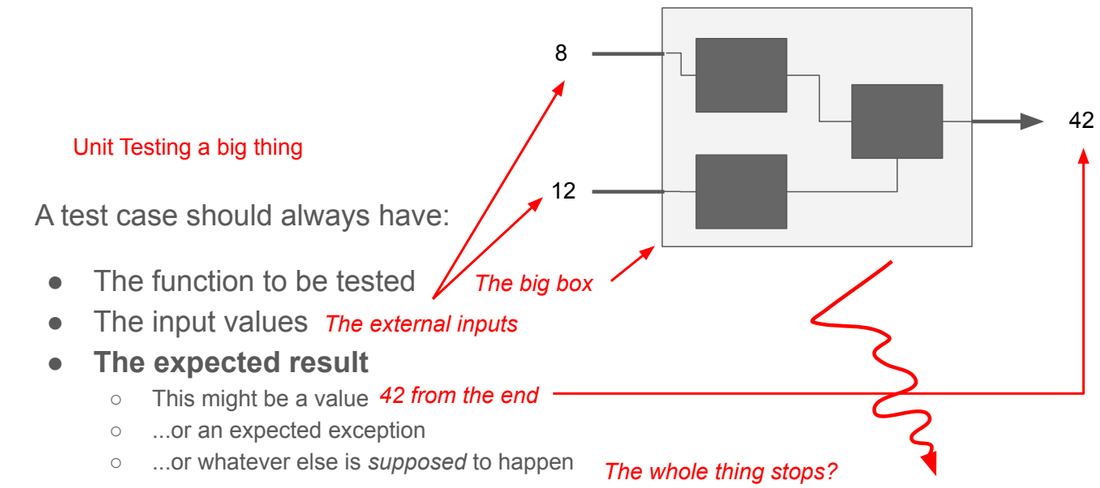
```
(4,4,4) -- Vote: 4, flag: 0
(2,4,4) -- Vote: 4, flag: 0
(2,2,4) -- Vote: 2, flag: 0
(2,3,4) -- Vote: 2147483647, flag: 0
```

A test case should always have:

- The function to be tested
- The input values
- **The expected result**
  - This might be a value
  - ...or an expected exception
  - ...or whatever else is *supposed* to happen

When is Unit Testing actually Integration Testing?

Unit Testing a big thing

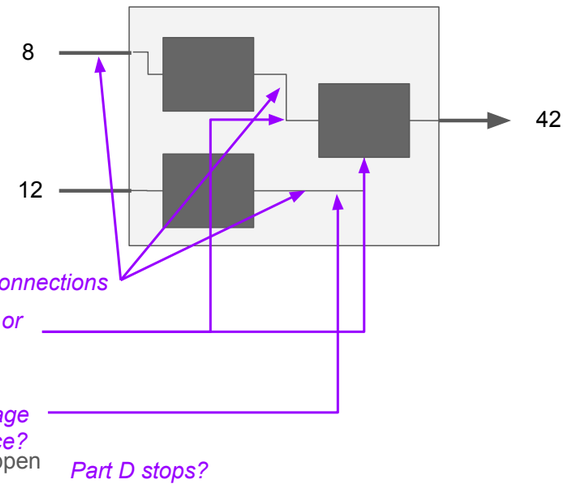


A test case should always have:

- The function to be tested
- The input values
- **The expected result**
  - This might be a value
  - ...or an expected exception
  - ...or whatever else is *supposed* to happen

When is Unit Testing actually Integration Testing?

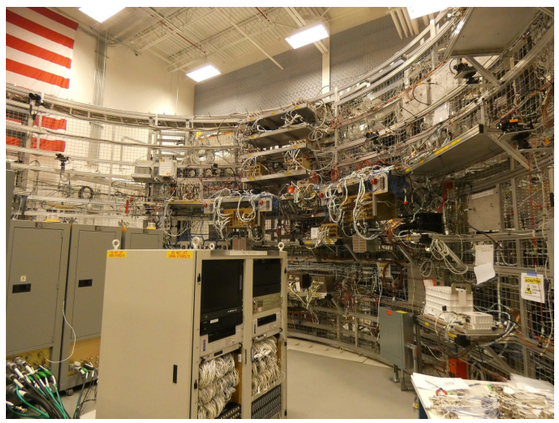
Integration Testing a set of things



A test case should always have:

- The function to be tested
- The input values
- **The expected result**
  - This might be a value
  - ...or an expected exception
  - ...or whatever else is *supposed* to happen

Integration Testing



- Physically separated components
- Reused components
- COTS components
- *Context* - DB? API?

System Testing

- “Whole” system
- External factors
- *Multiple* external factors at different points in the system

Acceptance Testing

- Testing directly against requirements
- Preferably by final or representative users
- Often where “User Experience” testing happens

Summary

- There is testing beyond Unit Testing.
- Where the line is can be a matter of opinion…
- Some of the test generation and test adequacy issues are still relevant.
- There are often other “interesting” factors - timing, sequencing, interaction patterns.
- System and Acceptance testing should involve “real world” situations and weird combinations of inputs (e.g. users!).