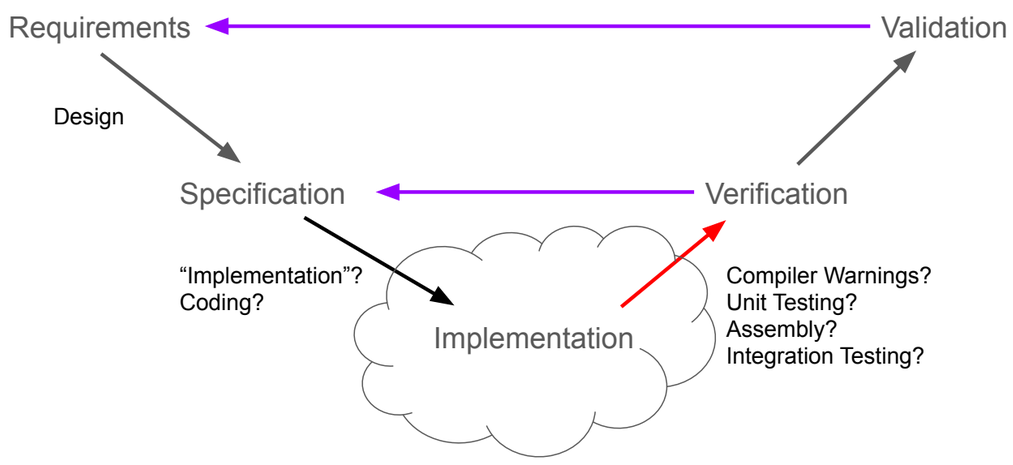
COM4506/6506: Testing and Verification in Safety Critical Systems

Dr Ramsay Taylor

Contents

After Implementation?



“Static” vs “Dynamic”

Static = not running

- observe the code and the system, see if you can see where it *might* go wrong

Dynamic = running

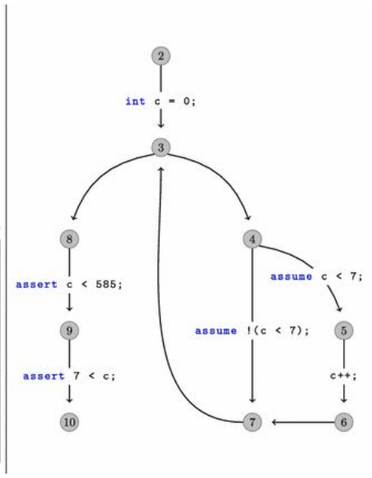
- try it and see if it *goes* wrong!

Static Analysis

- Compilers!
- Data Flow analysis
- Spec Checking
- Abstract Interpretation

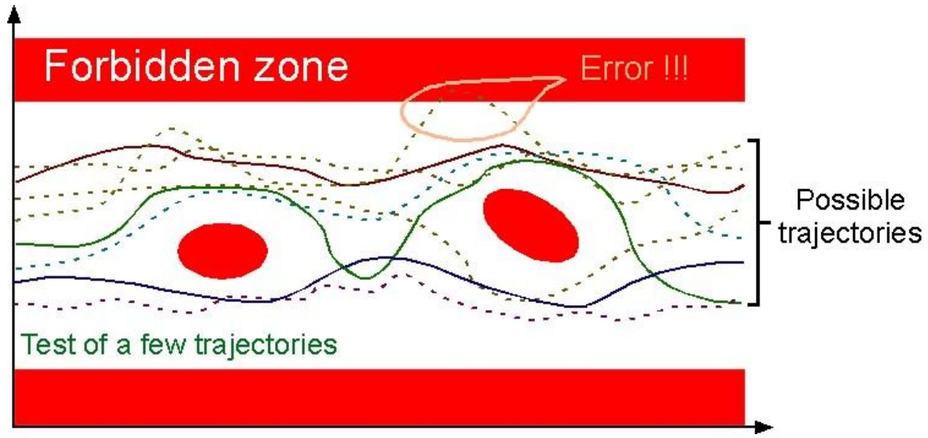
```
1  public void M() {
2    int c = 0;
3    while (*) {
4      if (c < 7) {
5        c++;
6      }
7    }
8    assert c < 585;
9    assert 7 < c;
10 }
```

Label	Interval state for c	
	Intermediate	Final
2	$\bot$	$\bot$
3	$[0,1]$	$[0,\infty]$
4	$[0,1]$	$[0,\infty]$
5	$[0,1]$	$[0,\infty]$
6	$[1,2]$	$[1,\infty]$
7	$[1,2]$	$[0,\infty]$
8	$[0,1]$	$[0,\infty]$
9	$[0,1]$	$[0,\infty]$
10	$[0,1]$	$[0,\infty]$



Christakis M., Wüstholz V. (2016) Bounded Abstract Interpretation. In: Rival X. (eds) Static Analysis. SAS 2016. Lecture Notes in Computer Science, vol 9837. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-53413-7_6

Static Analysis



https://www.di.ens.fr/~cousot/AI/IntroAbsInt.html

Dynamic Analysis - a.k.a Testing!

- “Does it Actually work?”

Dynamic Analysis - a.k.a Testing!

- ~~“Does it Actually work?”~~
- “Does it do what the Spec says it should?”

Dynamic Analysis - a.k.a Testing!

- ~~"Does it Actually work?"~~
- ~~"Does it do what the Spec says it should?"~~
- "If I put in 4 and 7, will I get 11?"

The Oracle problem...

".... well, it didn't crash…"

Dynamic Analysis - a.k.a Testing!

- ~~"Does it Actually work?"~~
- ~~"Does it do what the Spec says it should?"~~
- ~~"If I put in 4 and 7, will I get 11?"~~
- "If I put in 4 and 7, will I *always* get 11?"

The Oracle problem...

```
(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

Test Sets

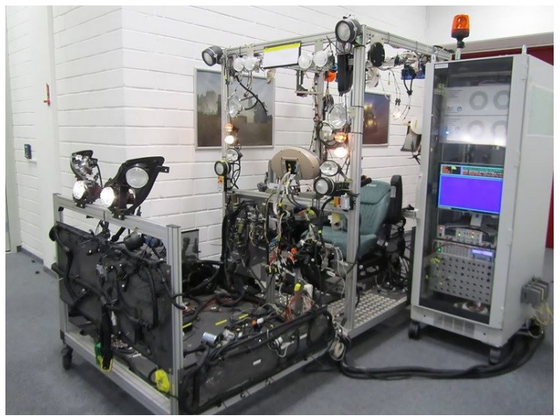
```
(4,4,4) -- Vote: 4, flag: 0
(2,4,4) -- Vote: 4, flag: 0
(2,2,4) -- Vote: 2, flag: 0
(2,3,4) -- Vote: 2147483647, flag: 0
```

A **Test Set** is … a set of tests!

Usually for a particular function or *unit*.

Test should be organised and modular for all the same reasons as the code!

Test environments



Does the software depend on interactions with hardware?

Or with other software components?

Does it behave differently in different *environments*?

Summary

- Static Analysis involves checking code without running it.
- Testing involves running the code - but hopefully with some objective!
- *How* we run the code matters, and we want to consider different conditions as well as different inputs.