Prifysgol Abertawe
Swansea University

CS-230 Software Engineering

L20: Integration Testing

Dr. Liam O’Reilly

Semester 1 – 2020

Recap: Two Aspects to Exceptions

- There are two aspects to dealing with bad situations at run time:
  - Easier Part: Code must be written to detect errors:
    - Use if statements to test variables and decide if there is a bad situation.
    - If there is then we can throw an exception.
    - Most library methods will already do this.
  - Harder Part: Handling Errors:
    - You need to ‘catch’ each possible exception and resolve the bad situation.
    - You should only catch exception if “you” are in a position where you ca actually deal with them.

Recap: Throw Early – Catch Late

- Throw Early:
  - When a method detects a problem that it cannot solve, it is better to throw an exception, rather than try to come up with an imperfect fix.
- Catch Late:
  - Only catch errors if you (the code) are in a position where you can actually remedy the bad situation properly!
  - If you catch an error then the error should be dealt with and execution continues as normal.
  - If you are not in such a position, the best action is simply to have the exception propagate to its caller, allowing it to be caught by a competent handler.
- Better to crash the program than continue on with even more errors due to bad situations not actually being resolved correctly.

Previously in CS-230...

- You now have code
- With low bug counts (maybe)...
- How do we ensure that it satisfies the requirements and is not buggy

Unit and Integration Testing

Testing is Very Important

- Testing is one of the most important aspects of software engineering
- It is one of the most time consuming parts
- It can be costly if done not so well
- We have not given it adequate attention in the course
  - But we should have...

Testing

- In the end, you'd like to be sure that your program works
- Thus, you need to do some sort of testing to ensure that it works
- Testing involves three things
  1. Look at the requirements and specification
  2. Look at what your code does
  3. Do the two match up?

- There are two types of testing we review:
  - Do the blocks of code work? – unit testing
  - Do the blocks of code work together? – integration testing

Unit Testing

- Testing various units (methods / classes) in isolation.
- In CS-135 you learnt:
  - Single Fault Assumption
  - Black-Box Testing:
    - Boundary Value Analysis:
      - *“Focus on the boundaries of the input space"*
    - Equivalence Class Testing:
      - “Often many inputs behave the same”
  - White-Box Testing:
    - Code coverage techniques

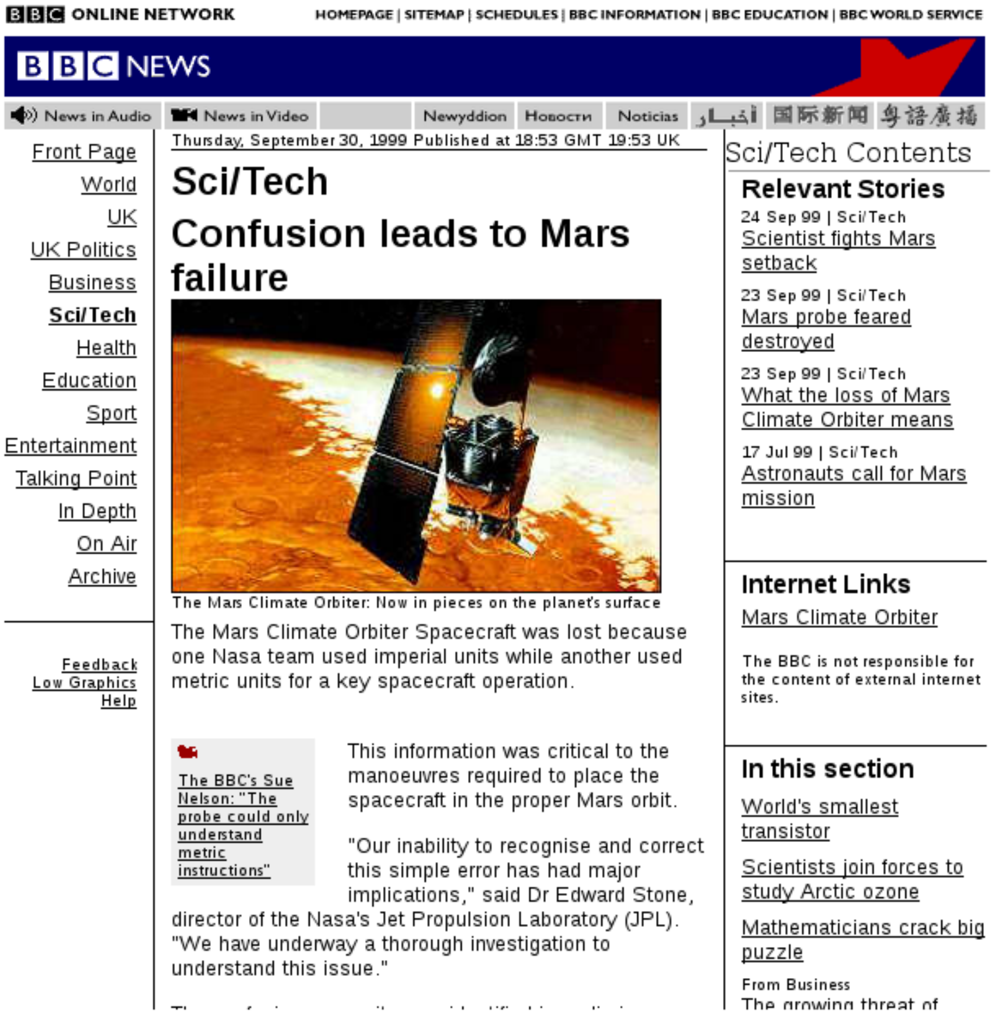
Integration Testing

Integration Testing

- Okay, assuming the classes individually work...
- What else can go wrong?
- Collaborations could have bugs
    - Known as integration testing
    - Test to make sure that there are no issues with method calls
    - Each “unit” could be fine but...
    - The units could be connected incorrectly.

Mars Climate Orbiter

- Mars Climate Orbiter Crashes into Mars
  - http://news.bbc.co.uk/1/hi/sci/tech/462264.stm
  - One NASA team used used imperial units while another used metric.



BBC ONLINE NETWORK — HOMEPAGE | SITEMAP | SCHEDULES | BBC INFORMATION | BBC EDUCATION | BBC WORLD SERVICE

BBC NEWS

News in Audio | News in Video | Newyddion | Новости | Noticias | أخبار | 国际新闻 | 粤語廣播

Front Page
World
UK
UK Politics
Business
Sci/Tech
Health
Education
Sport
Entertainment
Talking Point
In Depth
On Air
Archive

Feedback
Low Graphics
Help

Thursday, September 30, 1999 Published at 18:53 GMT 19:53 UK

Sci/Tech

Confusion leads to Mars failure

The Mars Climate Orbiter: Now in pieces on the planet's surface

The Mars Climate Orbiter Spacecraft was lost because one Nasa team used imperial units while another used metric units for a key spacecraft operation.

The BBC's Sue Nelson: "The probe could only understand metric instructions"

This information was critical to the manoeuvres required to place the spacecraft in the proper Mars orbit.

"Our inability to recognise and correct this simple error has had major implications," said Dr Edward Stone, director of the Nasa's Jet Propulsion Laboratory (JPL). "We have underway a thorough investigation to understand this issue."

Sci/Tech Contents

Relevant Stories

24 Sep 99 | Sci/Tech
Scientist fights Mars setback

23 Sep 99 | Sci/Tech
Mars probe feared destroyed

23 Sep 99 | Sci/Tech
What the loss of Mars Climate Orbiter means

17 Jul 99 | Sci/Tech
Astronauts call for Mars mission

Internet Links

Mars Climate Orbiter

The BBC is not responsible for the content of external internet sites.

In this section

World's smallest transistor

Scientists join forces to study Arctic ozone

Mathematicians crack big puzzle

From Business
The growing threat of

Integration Testing Goals

- To do proper integration testing
  - Test the relationships between components (classes/subsystems)
- What bugs can happen during integration?
  - Units of measurement differ
  - Mismatched parameters
  - Method arguments out of range

Frequent Errors: Units of Measurement Differ

```
...
// Acceleration by Earth's Gravity in m/s^2
private static final double G = 9.81;
...

public double speed(double time) {
    return G * time
}
```

- What happens if:
  - We enter time in hours?
  - Interpret speed as km/h?

- Good documentation helps!

Frequent Errors: Mismatched Parameters

```
public GameBoard(int w, int h) {
    this.board = new int[h][w];
}
```

- See the bug?
- We could also have the reverse bug where we call

  new GameBoard (h, w);
- Code is right and integration is wrong

Frequent Errors: Method Arguments Out of Range

```
GameBoard gameb = new GameBoard(100,200);
....
gameb.placePiece(100,200);
....
```

- We all right?
- No we aren’t! Out of range error.
    - Board indexes are [0...99],[0...199]
- Out of bounds exception

Stubs and Drivers

- In order to perform integration testing we will make use of stubs and drivers:
  - Dummy methods/classes.
  - They do not implement the entire programming logic of the software module but just simulate data communication with the calling module.

- **Stub**: Is called by the Module under Test.
- **Driver**: Calls the Module to be tested.

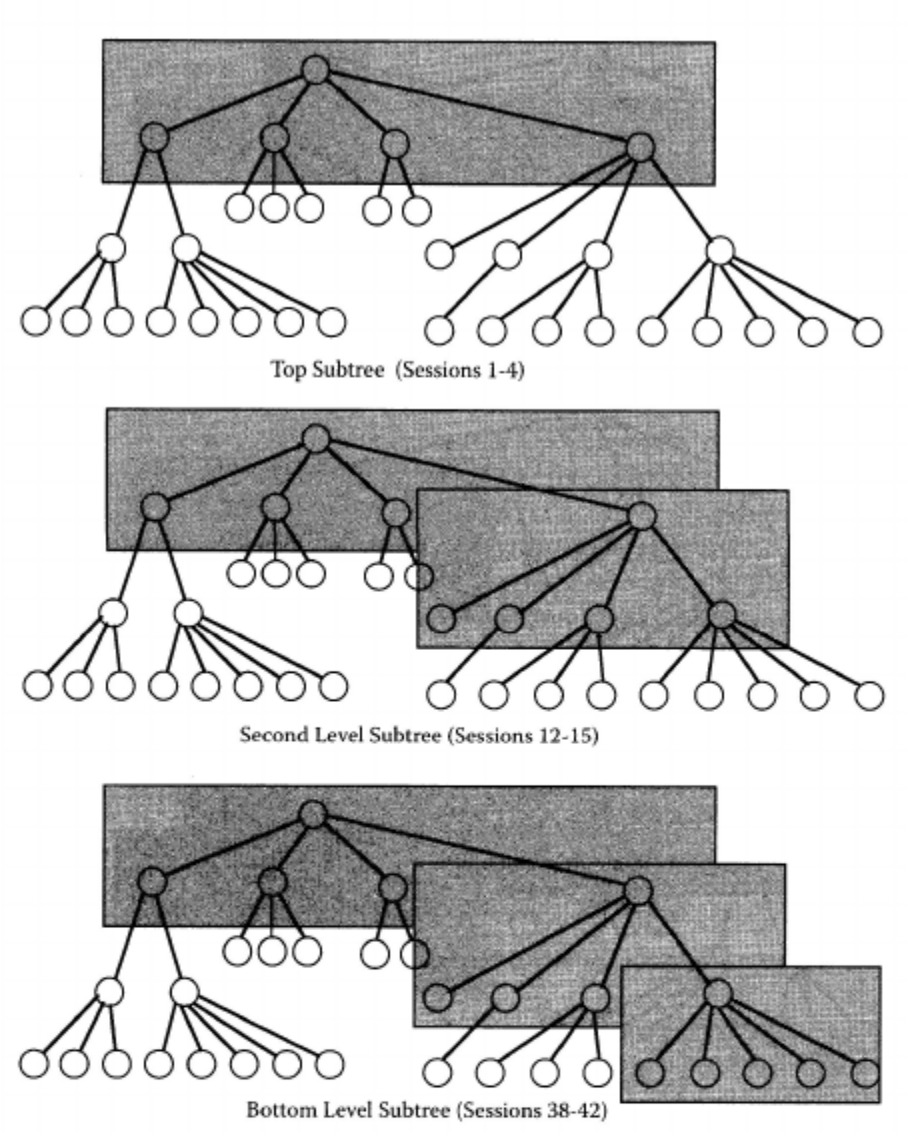
Top Down Integration Testing

- Test integration from calling method perspective.
  - Start at the entry point of the system.

  - To test a module integrates with those it uses:
    - For each used/called class/method:
      - Replace class/method with a stub.
      - Stub acts or returns the proper answer for the test case.
    - Now see if integration works out. We do not rely on implementation of used methods/classes.

  - Work down towards unit testing.
  - Good for testing additional features added to the system.

Top Down Integration Testing



Top Subtree (Sessions 1-4)

Second Level Subtree (Sessions 12-15)

Bottom Level Subtree (Sessions 38-42)

Jorgensen Book and M. Roggenbach Notes

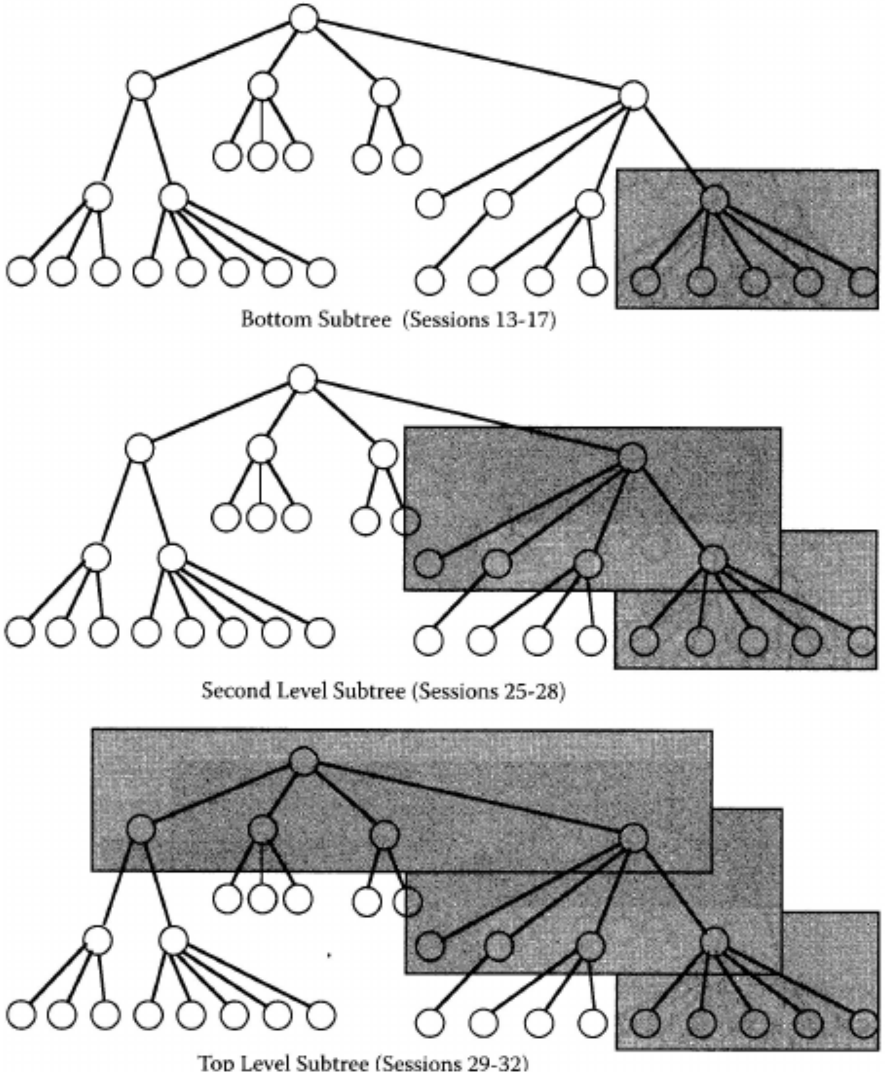
Bottom Up Integration Testing

- Test integration from the called method perspective.
    - Start at the class that does not require further calls.

    - To test that a module integrates with those that use it:
        - Unit test it the called module.
        - For a class/method that calls the module:
            - Replace with a driver.
            - Driver runs logic on the module.
        - Now see if integration works out. We do not rely on implementation of “higher” methods/classes.

- Work back toward entry point of the system.

Bottom Up Integration Testing



Bottom Subtree (Sessions 13-17)

Second Level Subtree (Sessions 25-28)

Top Level Subtree (Sessions 29-32)

Jorgensen Book and M. Roggenbach Notes

Summary

- Integration testing examines the collaborations:
  - Hold all communications fixed, but one (stubs).
  - Test the communication on that one.
  - Proceed to the next one.

- Methods help to localize bugs in communication between classes/subsystems.

End of Software Engineering?

Have We Reached the End?

- We have reached the end of the lectures for this module.
- What now?
- For the Software Engineers:
    - Next Term CS-235
- For all of you:
    - Finishing off A2.

The Big Picture

- The information in this unit is only the starting point.
- You will use a lot of the module content in your dissertations.
  - Well you will if you do a good dissertation.

- What about employment?
  - All companies work differently.
  - Hopefully, you now have the background knowledge to adapt and pickup how things are done in the real world.
  - Companies will expect you to know the basics and be able to adapt to their way of working.

End