CS608
Software Testing

Dr. Stephen Brown

Room Eolas 116

stephen.brown@mu.ie

Taking Notes

- **Take notes** during each lecture – don't just sit and listen
  - Active participation helps to retain knowledge
  - And improves understanding
- Evidence that handwritten notes are more effective than `typed notes`
- Slides and textbook are additional learning aides
  - Not a substitute for **taking your own notes**
  - After each day's lecture & lab, read relevant chapter(s) in the book
- Labs are essential part of the course
  - Only learn how to test software through active engagement

MODULE OVERVIEW

CS608 Module Overview

- This module provides students with an understanding of the **essential principles** of testing, and experience in **applying** these in an automated test environment

CS608 Module Overview

- This module provides students with an understanding of the essential principles of testing, and experience in applying these in an automated test environment
- Topics include:
  - Essential **principles** of testing
  - **Automated test** tools
  - Testing in the **software process**,
  - **Black-box** and **white-box** test techniques
  - Testing **object-oriented** software
  - Testing software systems (**Web Application)**

CS608 Module Overview

- This module provides students with an understanding of the essential principles of testing, and experience in applying these in an automated test environment
- Topics include essential principles of testing, automated test tools, testing in the software process, black-box test techniques, white-box test techniques, testing object-oriented software, system/application testing
- **Balance between principles and applying them in practice**

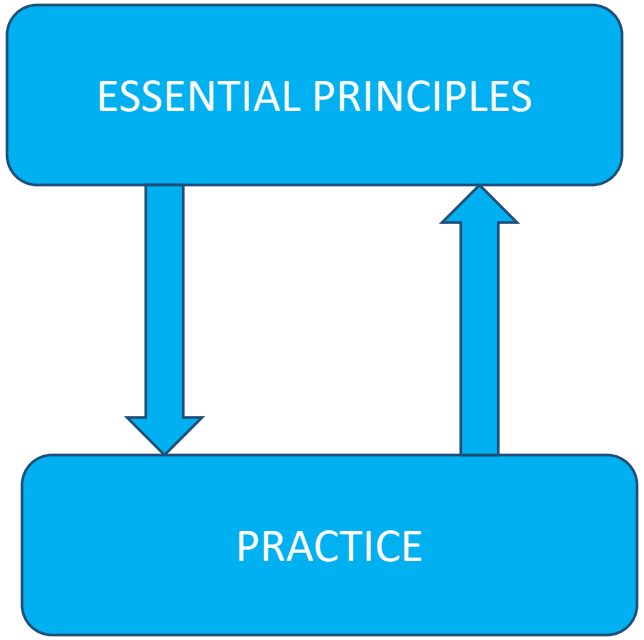
Goals

- Develop knowledge and understanding
- Gain some experience

Goals

- A **knowledge** and **understanding** of:
  - The essential principles of testing
  - The essential principles of test automation
  - Advanced testing issues, including random testing

Goals



- A knowledge and understanding of:
  - The essential principles of testing
  - The essential principles of test automation
  - Advanced testing issues
- **Experience** in:
  - Applying the principles to unit, object-oriented, and application testing
  - In an automated test environment

Learning Outcomes

- Be able to:
  - describe the principles of software testing
  - describe and compare testing techniques
  - select test techniques and design test cases
  - design and implement automated tests
  - evaluate test coverage
  - describe and evaluate testing in the software development process

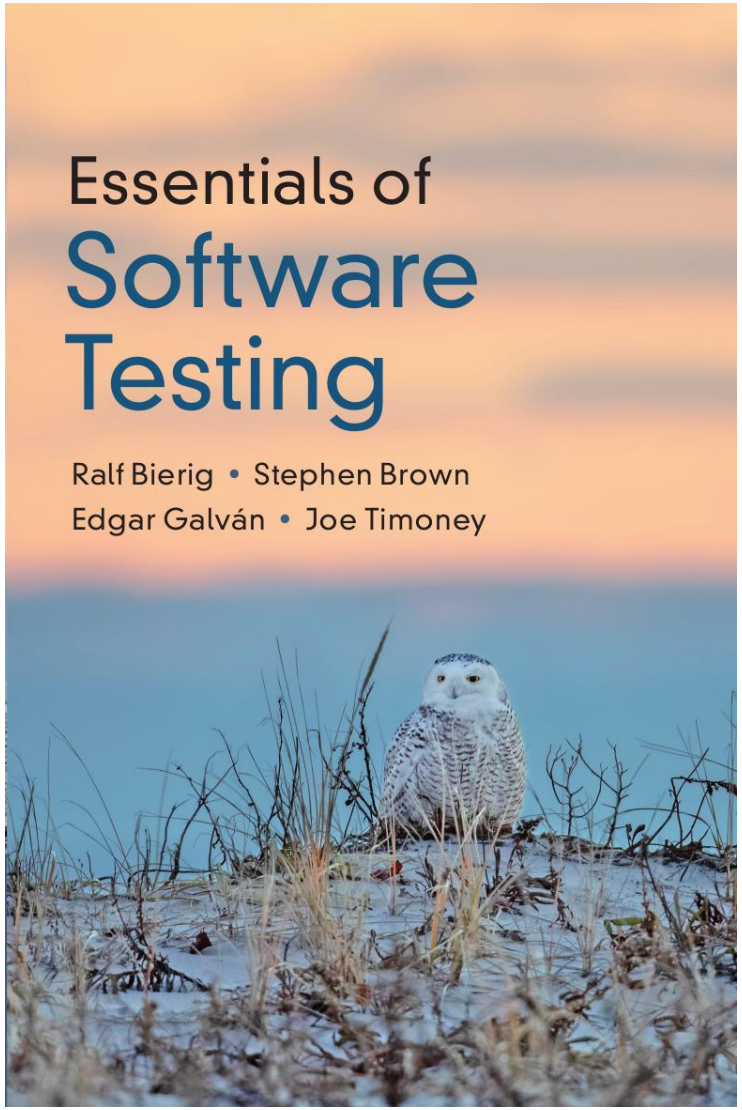
Approach

- Fast paced:
  - Cover unit testing and application testing relatively quickly
  - To give more time for:
    - Object-oriented testing
    - Test automation
    - Random testing
    - Other advanced Issues

Tools

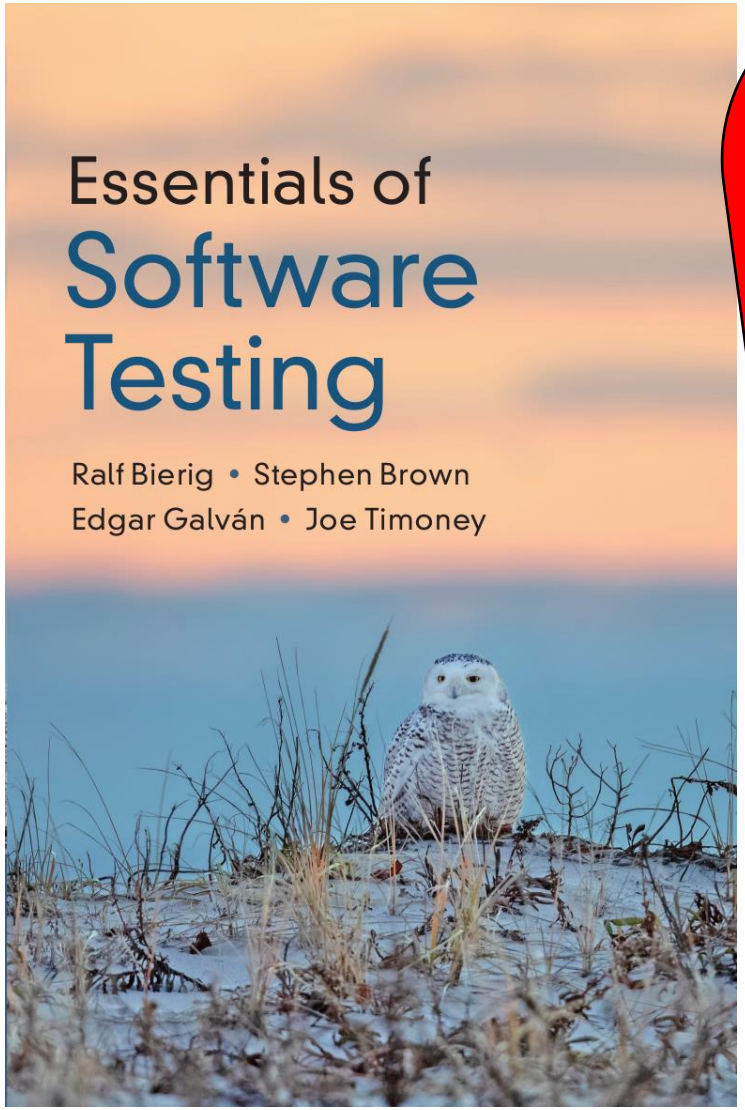
- In this module, you will learn to make use of a number of software test automation tools
- These tools are representative examples – picked for ease of use
- But CS608 is not a tools training course – you will not learn all the details of these tools
- Once you know how to use representative tools, moving to new tools should be straightforward
- Reference Chapter 11 “Test Automation”

TEXTBOOK



- *Essentials of Software Testing* has been written specially as a textbook for this module (published by Cambridge University Press)
- Each topic starts with worked examples, then covers principles in more detail
- Uses test automation throughout

TEXTBOOK



BOOK & SLIDES

BUT I STRONGLY RECOMMEND YOU TAKE NOTES DURING LECTURES

Timetable

Semester 2 *Subsequent weeks*

Times	Monday	Tuesday	Wednesday	Thursday	Friday
9:00-10:00		CS608 Callan S/W Lab	CS605 Callan S/W Lab	CS615/CS615C Callan S/W lab	CS603 Callan S/W lab
10:00-11:00	CS610 Callan S/W lab				
11:00-12:00					
12:00-13:00					
13:00-14:00					
14:00-15:00	CS610 Callan S/W lab			CS615/CS615C Details pending	
15:00-16:00		CS608 Eolas 019 Group A 1500-16:30			CS603 Callan S/W lab
16:00-17:00		CS608 Eolas 019 Grp A 15:00-16:30 Grp B 16:30-18:00	CS605 Callan S/W Lab		
17:00-18:00		CS608 Eolas 019 Group B 16:30-18:00			
18:00-19:00		CS608 Callan S/W Lab			

Typical Morning Lecture (Callan SW Lab/1.105) 9:00 – 12:00 with two 10-minute breaks

1. Complete previous topic:
   - Consider the topic in more detail
   - Identify key points
   - Notes for experienced testers
   - Questions from previous week's lab
2. Start a new topic:
   - Introduction
   - Worked example
   - Test implementation and results
   - Evaluation

Typical Afternoon Lab
14:00-18:00

- Develop tests based on the day's topic
  - I strongly recommend working the problems on paper first
  - Then use your paper work to code the automated tests
  - And answer the assessment quiz
- Lab Timetable (3h labs)
  - 13:00-15:00 in Callan SW lab (Callan 1.105)
  - 15:00-16:30 in Eolas E019 (Group A)
  - 16:30-18:00 in Eolas E019 (Group B)
- Extra lab time
  - 18:00-19:00 in Callan SW lab (Callan 1.105)
- Change your group on Moodle (subject to space limits)

Typical Afternoon Lab

- Exercises based on the day's topic
- I expect it to take you up to 3 hours of work
- Implement test code on the lab PC's or on your own PC
  - Use Lab 1 to check you have the necessary software/setup
  - The book examples and labs have all been checked on lab PCs
  - The book examples and labs have been checked on macOS & Linux
  - But I can’t support you on your own laptop if you have software problems…

Typical Afternoon Lab

- Exercises based on the day's topic
- Do on the lab PC's or on your own PC
- Three parts:
  1. **Develop the tests**
     1. Generate the answer to a testing problem
     2. You can use Excel, but I **strongly recommend** using pen and paper
  2. **Implement** the automated test code
  3. **Complete the lab assessment**
     1. Use your answer to complete a Moodle Quiz
     2. Grade assigned when then quiz closes (6 days)

Typical Afternoon Lab

- Exercises based on the day's topic
- Do on the lab PC's or on your own PC
- Three parts: develop tests, implement tests, and assessment
- You have 6 days to complete the assessment for each lab
  - I am available during Tuesday afternoons for questions
  - You must complete the lab assessment/quiz **on your own**
  - However, as software testing is a collaborative activity, you may **work together** developing the answers for the labs
  - The assessment quiz closes on Monday at 12:00 the following week to give you time to review your marks before next lecture

Typical Afternoon Lab

- Exercises based on the day's topic
- Do on the lab PC's or on your own PC
- Three parts: develop tests, implement tests, and assessment
- You have 6 days to complete the assessment for each lab
- **Let me know immediately if you are unable to complete a lab through illness etc.**
  - You will have to submit your written work also in this case

Continuous Assessment Activities

- Everything is on Moodle
- Assessment – typically for each topic, the lab consists of:
  - Develop your tests (on paper)
  - Write an automated test program (in Java)
  - Run your tests and collect the output
- Assessment
  - Complete a quiz to grade your lab work

Assessment

- Formal Written Examination:
  - 80%
  - 1 x 3 hour written exam at the end of the semester
- Continuous Assessment:
  - 20%
  - 10 labs, one every week

Material

- CS608 on Moodle (moodle.maynoothuniversity.ie)
  - Slides
  - Lab exercises
    - Instructions
    - Code
  - Support Material
    - Executable book examples (ZIP file)

Other Useful Books

- *Software Testing* [Roper] – highly recommended
- *Testing OO Systems* [Binder] – very detailed!
- *The Art of Software Testing* [Myers] – classic text, but out of date
- *Introduction to Software Testing* [Ammann & Offutt] – rigorous, mathematical approach

Asking Questions

- During lectures, if you don’t understand or need clarification, ask!
- Ask questions via Moodle “Class Discussion Forum”
  - I will copy any questions asked by email into the forum so everyone can see my answer
- Ask lab-related questions in the following week's lecture

Software Tools
Examples in the Book and Labs

- Environment
  - Windows, linux, and macos
  - Command line/terminal (quick tutorial available on Moodle)
  - **Gradle** build automation tool
- Software requirements (only 2)
  - Java LTS version: Java JDK 21
  - Google Chrome browser
- Note: gradle will download any other software needed automatically
- Test Framework: **TestNG**
- Test coverage measurement: **JaCoCo**
- Web Application Testing: **Selenium** with Chrome browser
- All details available on Moodle

Focus on Automated Testing

- This means running tests automatically, rather than by hand
- Introduced as needed during the lectures
- Covered in more detail as a topic

Independent Study

- After the lectures/labs:
  - Complete outstanding lab work and assignments
  - Tidy up/review your notes
  - Read the relevant chapters in the book
  - Re-do worked examples in the book
  - Do additional exercises
  - Do exam questions (library EXPERT database)
  - Read up material in the secondary books (library)
  - Prepare questions for the following week

Artificial Intelligence for Software Testing

- AI Tools such as Copilot have become a valuable tool for coding
- For example:
  - ChatGPT
  - GitHub Copilot
  - Microsoft Copilot
- These tools can be used to assist in testing also
- Once you are familiar with the software testing techniques, we will examine how well ChatGPT produces software tests

Any Questions?