

Software Testing

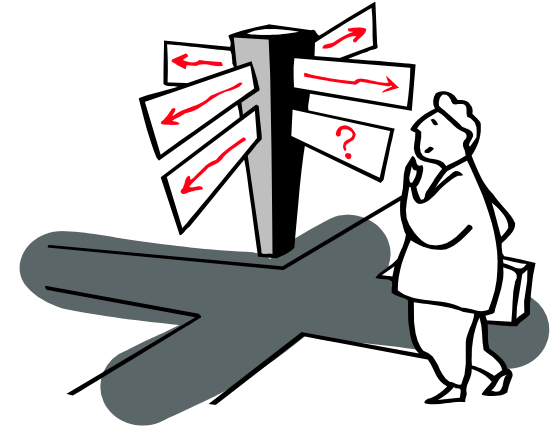
0. Practical Matters



Universiteit Antwerpen

0. Practical Matters

- Contents
- Goals
- Course Material
- Plan
- Exams and Grade



Contents

The student will acquire experience with thorough testing and verification of a software system to guarantee with a certain degree of confidence that a given software system meets its specification.

The course has a practical ring to it with

- a minimal theoretical content
 - taught as testing patterns,
- several lab sessions
 - trying out several test techniques and strategies on an existing representative software system
- a few guest speakers from industry
 - confirming that the testing techniques covered in the lectures indeed are used in practice

Attendance is required.

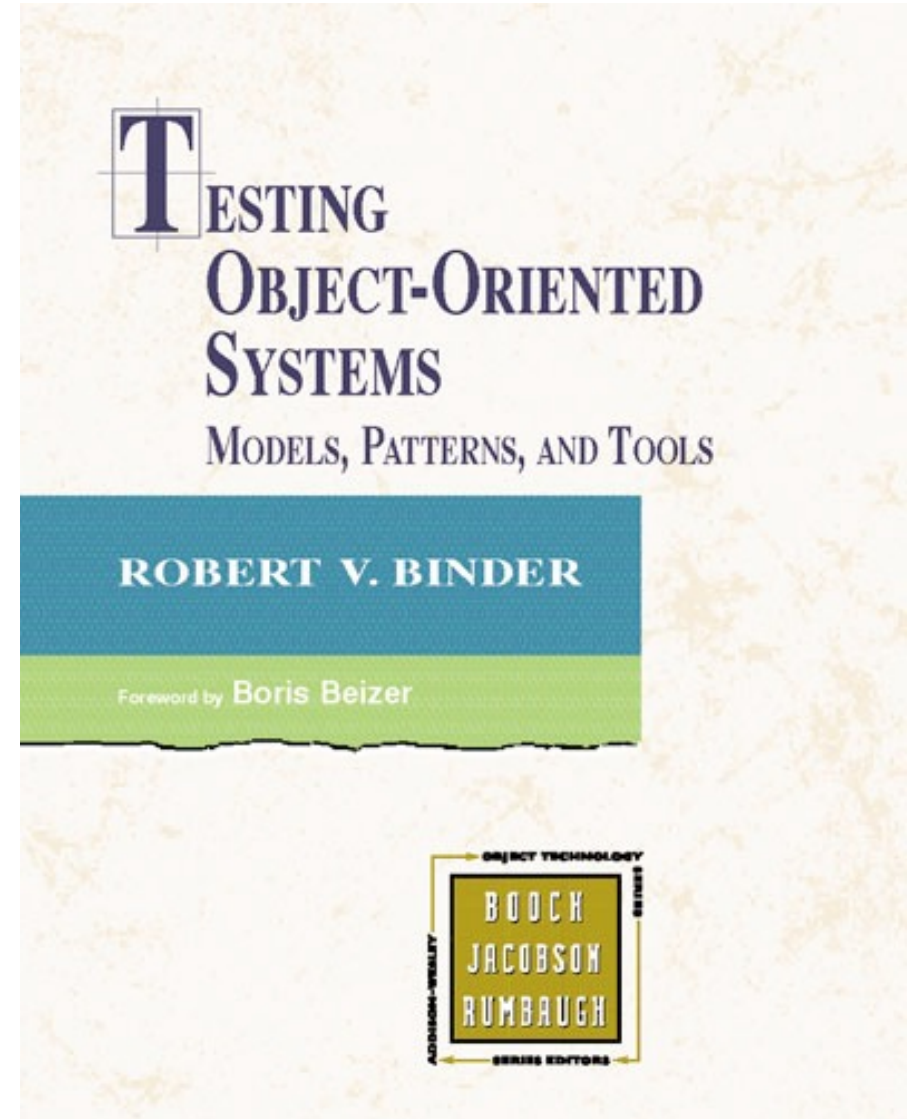
Goal(s)

You will be able to ...

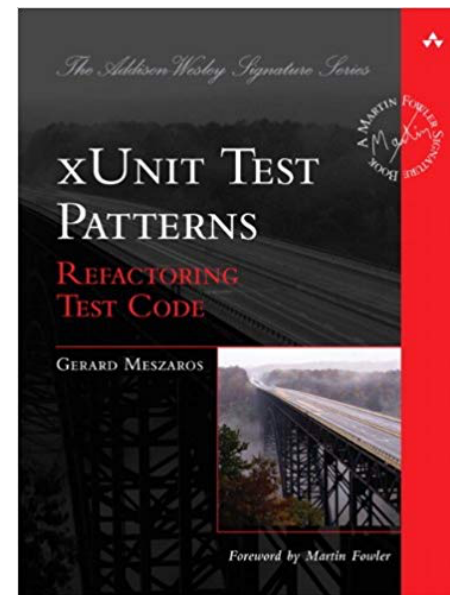
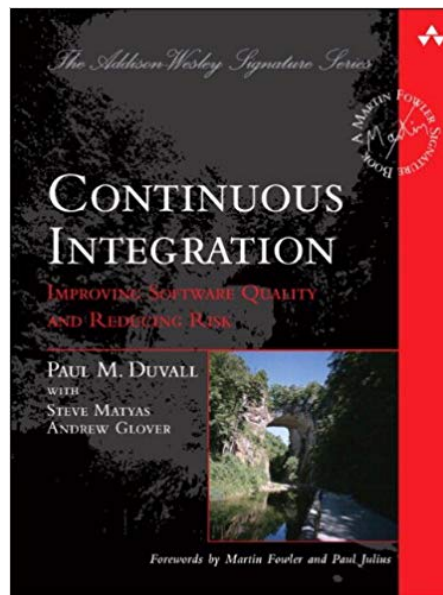
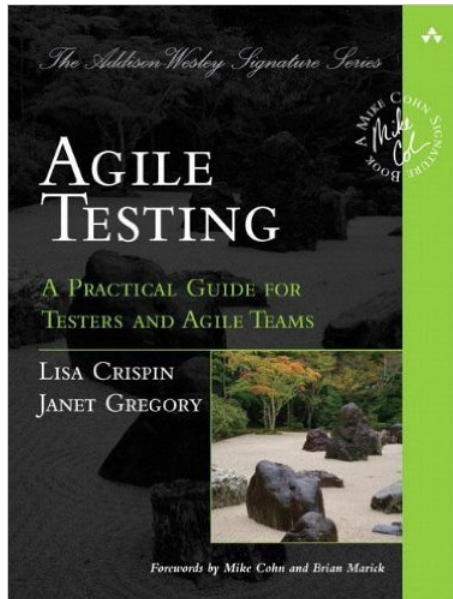
- apply white-box and black-box test techniques to build a test-suite;
- measure (and improve) the coverage of a test-suite;
- choose the most appropriate test strategy;
- review and verify a (test)model of a given system.

Course Material

- Robert V. Binder. Testing Object-Oriented Systems. Addison-Wesley, 2000
 - Multiple copies available in the library
 - Material available on author's web-site
<http://www.rbsc.com/>

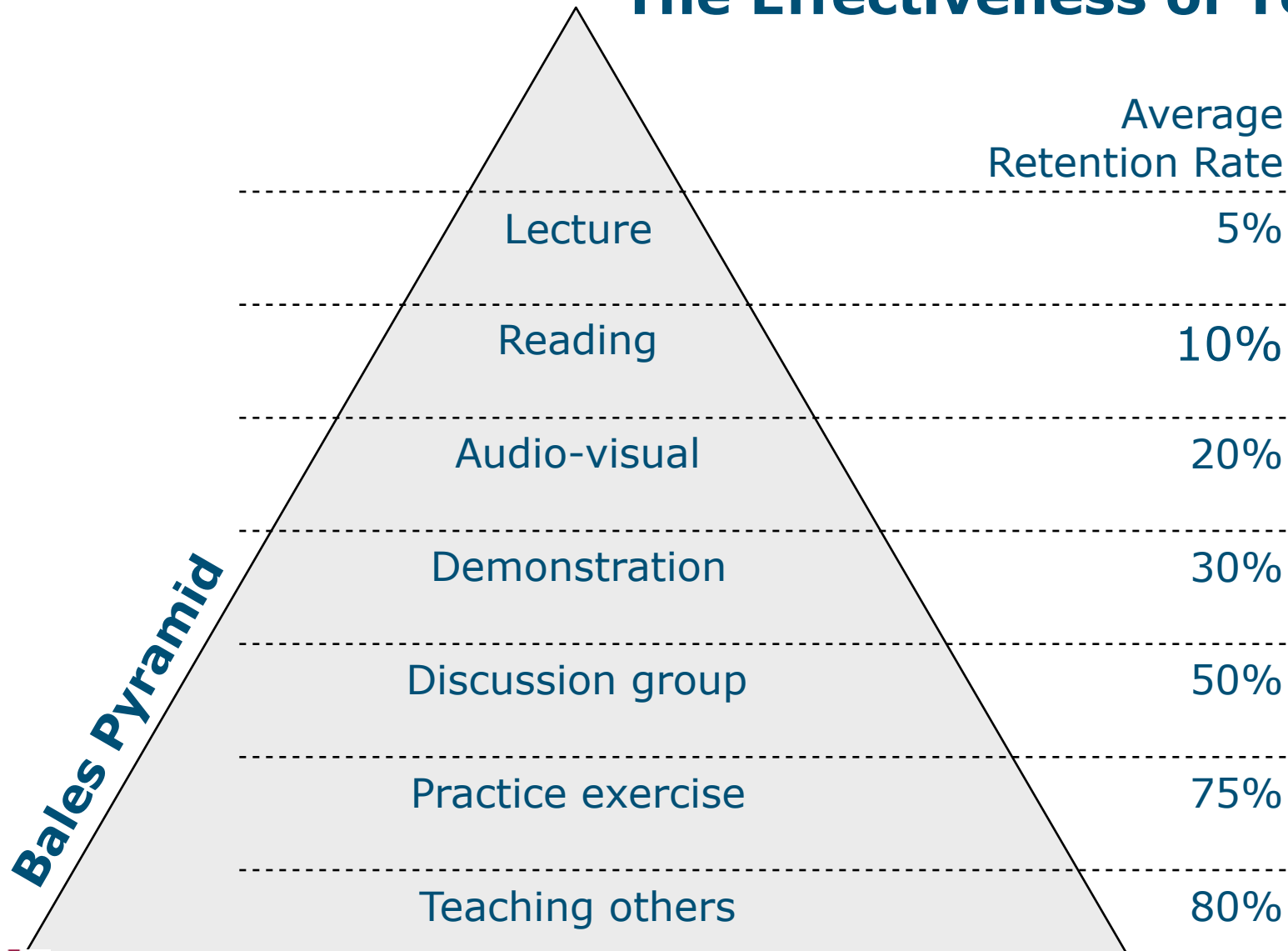


Course Material (Test Automation)



To be confirmed

The Effectiveness of Teaching



Attendance is *required*.

Teaching Approach

- One group of students (2-3) prepares one lecture
 - reads corresponding chapters in detail
 - prepares slides to present in the class room
slides will be distributed by means of the web-site (PDF)
 - include references to industry guest lectures
 - presents the material to her/his peers during lecture
- One group of students (2-3) serve as opponents
 - reads corresponding chapters in detail
 - prepares questions to be asked after lecture
- Peer assessment: all students rate the lecture
 - (a) knowledge about subject
 - (b) understandability of the lecture
 - (c) coherence of material presented

See separate checklist
for details !!

(Ideal) Schedule

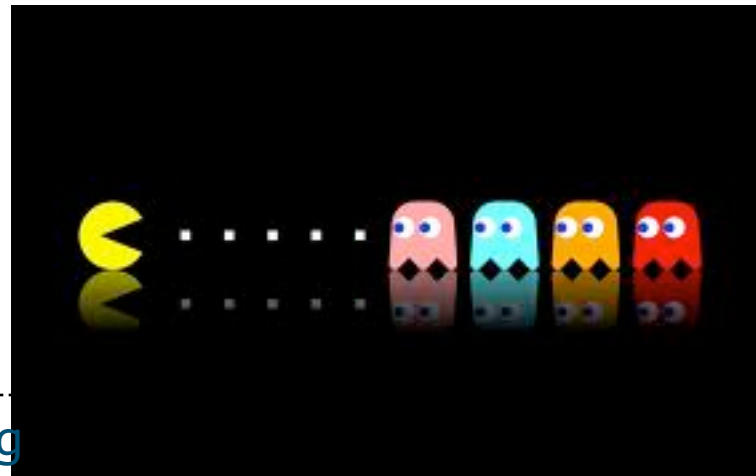
Detailed planning @ course web-site

1	Introduction (Ch. 3 en 4)	prof. Demeyer
2	Test Models (Ch. 5, 6 en 7)	prof. Demeyer
3	External industrial speaker	guest speaker
4	External industrial speaker	guest speaker
5	External industrial speaker	guest speaker
6	External industrial speaker	guest speaker
7	External industrial speaker	guest speaker
8	External industrial speaker	guest speaker
	-- holidays	
9	Ch.10 (Classes) + Ch. 11 (Components)	student(s)
10	Ch.12 (Subsystems) + Ch. 13 (Integration)	student(s)
11	Ch.14 (Application) + Ch.15 (Regression)	student(s)
12	?? Test Automation / Agile Testing ??	student(s)
13	-- buffer slot	-- buffer slot

jpacman

© Prof. Arie Van Deursen from TUDelft.

<https://github.com/avandeursen/jpacman-framework>



1) Java testing tools

2) Category Partitioning and Boundary Values

3) Enforcing proper associations with asserts

4) Decision Structures

7) GUI Testing

6) Mutation testing

5) State Machines

Exam and Grade

DURING THE SEMESTER (Continuous Evaluation)

- Student lecture
 - criteria:
 - (a) knowledge about subject;
 - (b) understandability of the lecture;
 - (c) coherence of material presented
 - peer assessment
- Lab sessions
 - hand in solutions via blackboard
 - at least 50/100 for each assignment

ORAL EXAM (+- 20 minutes)

- Guest lecture (random assignment)
- Testing Patterns (random assignment; not one that you presented)