




**SOFT3302/3602
Software Quality
Assurance**

Introduction

Michaelmas 2007 SOFT3302 Lecture 1 1



DIA Baggage Handling System


Denver International Airport opened in 1995

- *Scheduled to open in 1993*

Delays caused by baggage handling software
High profile failure
In 2005, United announced that it would
abandon the automated baggage system

[Gal04] [Wik05]

Michaelmas 2007 SOFT3302 Lecture 1 2



Ariane 5

On June 4, 1996, the maiden flight of the
European Ariane 5 launcher crashed about
40 seconds after takeoff.

The loss was \$500,000,000 uninsured
An uncaught exception caused entire software
to crash

[Jéz97]

Michaelmas 2007 SOFT3302 Lecture 1 3

RMIT's AMS

Original budget \$12.6M + 3 x \$6M

- Blew out to \$47.3M*

Chancellor resigned

- Many councilors resigned*

Government forced to cover costs

Lack of (structured) testing identified

[AGV03] [Sch03]

Michaelmas 2007 SOFT3302 Lecture 1 4

Other High-profile Disasters

Ebay

- Repeated crashes*
- Outages*
- Lost revenue*

NASDAQ

- 17 min outage due to upgrade*
- Up to 1 500 000 000 trades a day*
- Lost revenue*

Michaelmas 2007 SOFT3302 Lecture 1 5

Near disasters

Still cost someone somewhere

Lost productivity

- Viruses*
- Intrusions*
- Cleanup*

Annoying

- BSOD on State Rail indicators*

Embarrassing

- Damaging to reputation and future sales*

Michaelmas 2007 SOFT3302 Lecture 1 6

Projects fail

In 1994 Standish Group *Chaos Report* found of 365 respondent and 8380 applications

- 16.2% *ontime, in budget, met requirements*
- 52.7% *completed but over budget, time, incomplete*
- 31.3% *cancelled during development*

Failure can be due to factors unrelated to testing

[Sta95]

Michaelmas 2007 SOFT3302 Lecture 1 7

Projects still fail

In 2006:

- 35% *ontime, in budget, met requirements*
- 46% *completed but over budget, time, incomplete*
- 19% *cancelled during development*

Things may be slightly better but still problems

Michaelmas 2007 SOFT3302 Lecture 1 8

Liability

Take a look at a software license
How much warranty does it give?
None!

- To the extent permitted by law*

Don't vendors have confidence in their products?

Michaelmas 2007 SOFT3302 Lecture 1 9

Liability

SAP AG fielded lawsuits from

- Whirlpool*
- Hershey*

GoreTex sued

- PeopleSoft*
- Deloitte & Touche*

[Co]

Michaelmas 2007 SOFT3302 Lecture 1 10

Why We Want Testing

Avoid disasters

Avoid legal nightmares

Increase productivity

- Testing brings confidence*
- Confidence reduces worry*
- Worry decreases productivity*

Michaelmas 2007 SOFT3302 Lecture 1 11

IT Helping or Hurting?

Labour productivity growth fallen since 1970's

At the same time IT has been expanding

Causal relationship?

Michaelmas 2007 SOFT3302 Lecture 1 12

Productivity

Tools and especially processes have not kept up with complexity

Higher level languages are certainly more productive than direct machine language

But systems are far more complex

- To design*
- To use*

Michaelmas 2007 SOFT3302 Lecture 1 13

Productivity

The IRS invested \$US50M in PCs for agents

Supposed to help

- Data entry and retrieval*
- Speed up calculations*

Number of cases processed dropped 40%

[Lan97]

Michaelmas 2007 SOFT3302 Lecture 1 14

Productivity

An insurance underwriter spend \$US30M to streamline a claims department

Number of claims processed by 65 employees increased over 30%

Total cost to process claims increased 30%

[Zub98]

Michaelmas 2007 SOFT3302 Lecture 1 15

Time shortages

Programmers are optimists

- *Estimation techniques are poor*
- *Want to please rather than defend estimates*

Effort is not the same as progress

Estimates are *guesses*

Progress often poorly monitored

- *Things slip a year one day at a time*

Michaelmas 2007 SOFT3302 Lecture 1 16

The Mythical Man Month

Cannot just throw more people at a problem

Need to make teams more effective

Debugging and testing are constraints

Time dependent on errors

Optimists assume fewer errors than the reality

Testing is one of the most mis-scheduled phases of the endeavour

Michaelmas 2007 SOFT3302 Lecture 1 17

Brooks' Scheduling Guidelines

1/3 design

1/6 coding

1/4 unit testing

1/4 system testing

50% of time spent on testing!

[Bro95]

Michaelmas 2007 SOFT3302 Lecture 1 18

Brooks' Scheduling Guidelines

Too little testing the norm

- *Far more common than too much*

Brooks found around 50% was spent testing

- *Even if less was allocated initially*

Testing handled as final phases of schedule

- *Too late*

Michaelmas 2007 SOFT3302 Lecture 1 19

Cost of errors

The cost of errors increases exponentially

An error caught in analysis costs less to fix than an error caught in the software maintenance cycle

Michaelmas 2007 SOFT3302 Lecture 1 20

The Mythical Man Month

Has anything changed?

Brooks revisited the issues 20 years later

Ten years later again things seem the same

Why?

Michaelmas 2007 SOFT3302 Lecture 1 21

Reasons

Hardware technology has exploded
Software technology has not kept pace

- *Software is invisible*
- *Software maintenance is different to hardware*

Must deal with obsolescence

- *Programmes hang around longer than intended*

People are still people

- *People are the problem*
- *How do we work in teams?*

Michaelmas 2007 SOFT3302 Lecture 1 22

What Is Testing?

Risk management

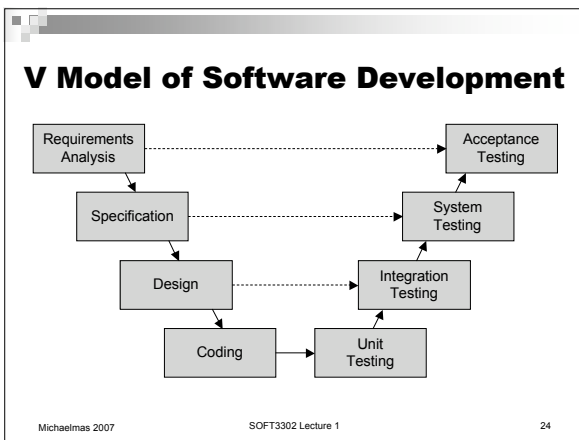
- *Confidence management*

Part (but not all) of quality assurance

- *The tests are not the goal*

Good testing should be an integral part of good programming

Michaelmas 2007 SOFT3302 Lecture 1 23



Types of Testing

Unit testing
Integration testing
System testing

And many more...

Michaelmas 2007 SOFT3302 Lecture 1 25

Exploratory Testing

Manual testing
Ad-hoc
Informal
Unscripted

Michaelmas 2007 SOFT3302 Lecture 1 26

User Acceptance Testing

Client reviews acceptability

- Performs typical tasks*
- Confirms requirements are met*

Needed for handover
May occur in cycles if using RAD

Michaelmas 2007 SOFT3302 Lecture 1 27

Performance Testing

Verify system works under 'normal' load
Various resource requirements/configurations
Ensures tasks complete in usable time
Ensures appropriate performance levels are achieved and maintainable

Michaelmas 2007 SOFT3302 Lecture 1 28

Load/Stress Testing

Examine performance of system under abnormal load
Determine how the system degrades

Michaelmas 2007 SOFT3302 Lecture 1 29

Usability Testing

Does the system fit the intended workflow?
Is the system effective?
Is the system convenient?

Michaelmas 2007 SOFT3302 Lecture 1 30

Security Testing

Authorisation versus authentication
Authority
Access
Identity
Can unauthorised actions be performed?

Michaelmas 2007 SOFT3302 Lecture 1 31

Backup and Recovery

Most don't do backups
Even more don't check backups
 Can information even be restored?

Michaelmas 2007 SOFT3302 Lecture 1 32

Alpha and Beta Testing

Alpha testing is in-house
Beta testing is external
Is a cost which must be factored in

Michaelmas 2007 SOFT3302 Lecture 1 33

Parallel Testing

Is a new system being developed?
Can they be run side-by-side?
Old system can serve as a reference

Michaelmas 2007 SOFT3302 Lecture 1 34

Regression Testing

Compare new and old behaviour
Small changes increase entropy
 New problems get added
Can be tricky at the application level
 Consider a GUI-based programmes

Michaelmas 2007 SOFT3302 Lecture 1 35

Reviews

Part of analysis, design and coding
Validation of requirements and design
Formal code review

Michaelmas 2007 SOFT3302 Lecture 1 36

Testing Is Measuring

Testing is a collection of metrics designed to measure a confidence level (risk)

Measures many factors

Results of testing are not numbers

- *Tests pass or fail*

Levels are embodied in the tests

- *Tests must be good*

Michaelmas 2007 SOFT3302 Lecture 1 37

Testing Is Not...

Prototyping

Verification of analysis or design by simulation

Scrutiny of code by review or walkthrough

Static code analysis, e.g. lint

Dynamic analysis, e.g. purify

Debugging

Michaelmas 2007 SOFT3302 Lecture 1 38

Testing Is Not...

A final tick in a box

A guarantee that software is error free

A standalone activity separate to development

Michaelmas 2007 SOFT3302 Lecture 1 39

A Tester Is Not...

- A failed developer
- The client or an end user
 - Unfair to client and lazy...*
- A quality assurance team
- The goal and controller of development
- Time and motion/process improvement person
- Bad news

Michaelmas 2007 SOFT3302 Lecture 1 40

A Good Tester Is...

- Empirical
 - Gather and assess evidence*
- Logical (and illogical!)
 - Learn to infer incorrectly and use different logic*
 - Understand common fallacies in informal reasoning*
- Rational
 - Must be able to justify beliefs*
 - Discern formal versus informal reasoning*
- Literate
 - Find meaning and ambiguity in natural language*
- Competent
 - Be able to make good decisions*

Michaelmas 2007 SOFT3302 Lecture 1 41

Testers...

- Identify unpopular threats, risks, error, problems
- Find big problems as quickly as possible
- Are there to *support* developers
- Question everything (but not always out loud)
- Focus on failure so others may succeed
- Should be just as creative, practical and technical as developers

[KBP02]

Michaelmas 2007 SOFT3302 Lecture 1 42

Testing Misconceptions

Those who can do, those who can't test
Anyone can do testing
Programmers should do all the testing
Testing is boring and requires no intelligence

Michaelmas 2007 SOFT3302 Lecture 1 43

The Limits of Testing

'Testing can only show the presence of bugs,
never their absence.'

E. W. Dijkstra

Michaelmas 2007 SOFT3302 Lecture 1 44

The Limits of Testing

Testing has limits
Testing cannot be exhaustive
Testing is risk management and so is a
cost/benefit trade-off

Michaelmas 2007 SOFT3302 Lecture 1 45

No Silver Bullet

Tools are just that: *tools* not solutions

- XML
- UML
- Java
- <insert favourite of choice>

They should not be used for the sake of it
Read [Bel06]

Michaelmas 2007 SOFT3302 Lecture 1 46

Course Outline

| # | Date | Subject |
|---------------------------|--------|--------------------|
| Introduction | | |
| 1 | Jul-26 | Introduction |
| Testing | | |
| 2 | Aug-2 | Structured testing |
| 3 | Aug-9 | Test design |
| 4 | Aug-16 | White box testing |
| 5 | Aug-23 | Testing frameworks |
| The Test Process | | |
| 6 | Aug-30 | Defect management |
| 7 | Sep-6 | Test management |
| 8 | Sep-13 | Test strategy |
| 9 | Sep-20 | Test process |
| Mid-semester break | | |
| Further Topics | | |
| 10 | Oct-4 | Formal methods |
| 11 | Oct-11 | Code reviews |
| 12 | Oct-18 | Useability testing |
| 13 | Oct-25 | Course revision |

Michaelmas 2007 SOFT3302 Lecture 1 47

Recommended Reading

See references

- Read Brooks*

There will be recommended reading throughout the semester
It is recommended you read the recommended reading

Michaelmas 2007 SOFT3302 Lecture 1 48

Assessment

30% assignments
 3 assignments

70% exam

'Satisfactory performance' barriers

- Usually around 40% in each component
- 12/30 in assignments; 40/100 in exam

Michaelmas 2007 SOFT3302 Lecture 1 49

Consultation

Consult your tutor first

- Daniel Tse

Associate Professor Judy Kay

- judy@it.usyd.edu.au
- SIT Building J12, Room 318
- Usual consultation hour

Dr James Farrow

- SIT Building J12, Room 444
- By appointment

Michaelmas 2007 SOFT3302 Lecture 1 50

Course web site

<http://www.it.usyd.edu.au/~cs3/soft3302>

- Tutorials
- Assignments
- Lecture notes
 - Yay!

Read the section on plagiarism

Michaelmas 2007 SOFT3302 Lecture 1 51

References

- [AGV03] Auditor General Victoria, *Report on Public Sector Agencies: Part 5*, February 2003
http://www.audit.vic.gov.au/reports_mp_psa/PSA_Feb03_Part5b.pdf
- [Bel06] Alex E. Bell, *ACM Queue* 4(5), June 2006,
<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=401>
- [Bro95] Fred Brooks, *The Mythical Man-Month*, 20th Anniversary Edition, Addison-Wesley, 1995.
- [Bou99] Neal E. Boudette, *Europe's SAP Scrambles to Stem Big Glitches*, Wall Street Journal, 4 November 1999.

Michaelmas 2007

SOFT3302 Lecture 1

52

References

- [Col] Stacy Collett, *SAP: Whirlpool's Rush to Go Live Led to Shipping Snafus*, Computerworld,
<http://www.computerworld.com/news/1999/story/0,1280,29365,00.html>, July 2005.
- [Gal04] Daniel Galin, *Software Quality Assurance: From theory to implementation*, Pearson Addison-Wesley, 2004.
- [IAB93] *The V-Model*, http://www.v-modell.iabg.de/kurzv/vm/k_vm_e.doc, February 1993.

Michaelmas 2007

SOFT3302 Lecture 1

53

References

- [Jéz97] Jézéquel and Meyer, *Computer* (IEEE) 1997 30:2, pp. 129–130 also available at
<http://archive.eiffel.com/doc/manuals/technology/contract/ariane/page.html>
- [KBP02] Cem Kaner, James Bach, and Bret Pettitcord, *Lessons Learned in Software Testing*, Wiley 2002.
- [Lan97] Thoman Landauer, *The Trouble with Computers*, MIT Press, 1997.
- [Sch03] Ben Schneiders, *Bungled Project Plunges RMIT into Financial Crisis*, The Financial Reviews 11 March 2003.

Michaelmas 2007

SOFT3302 Lecture 1

54

References

[Sta95] The Standish Group, *The Chaos Report (1994)*, <http://www.standishgroup.com/>

[Wik05] *Denver International Airport*, *Wikipedia: The Free Encyclopædia*, http://en.wikipedia.org/w/index.php?title=Denver_International_Airport&oldid=18918693, 21:52, 15 July 2005

[Zub98] S. Zuboff, *In the Age of the Smart Machine*, Basic Books, 1998.

Some material adapted from 2004 SOFT3103 course.

Michaelmas 2007 SOFT3302 Lecture 1 55
