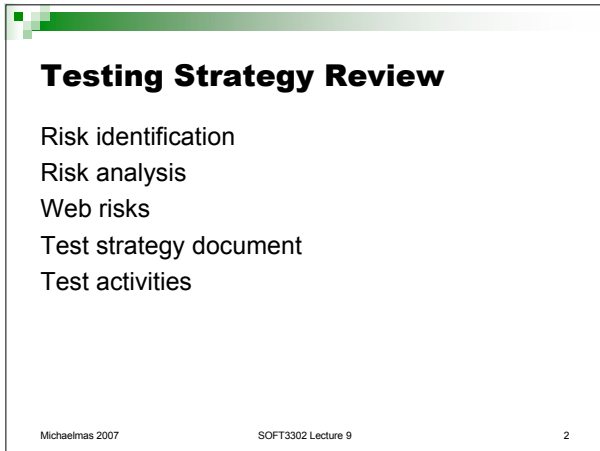




**SOFT3302/3602
Software Quality Assurance**

Test Process

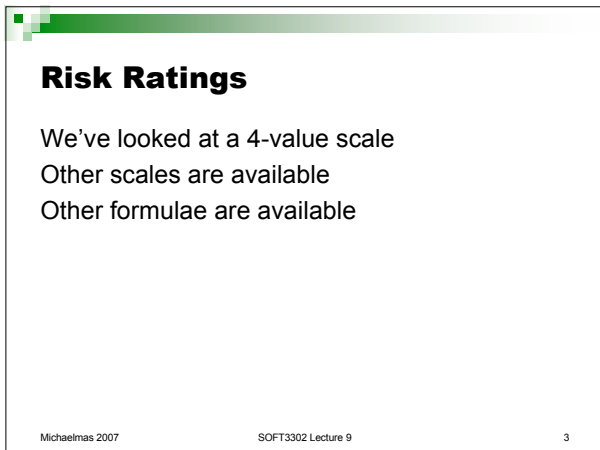
Michaelmas 2007 SOFT3302 Lecture 9 1



Testing Strategy Review

- Risk identification
- Risk analysis
- Web risks
- Test strategy document
- Test activities

Michaelmas 2007 SOFT3302 Lecture 9 2



Risk Ratings

- We've looked at a 4-value scale
- Other scales are available
- Other formulae are available

Michaelmas 2007 SOFT3302 Lecture 9 3

Damage Severity Rating

Very High (VH)
 System crashes or becomes inoperative

High (H)
 Data corruption

Low (L)
 Failure of functionality

Very Low (VL)
 Cosmetic
 Does not result in failure to deliver functionality

Michaelmas 2007 SOFT3302 Lecture 9 4

Likelihood Rating

Very High (VH)
 Occurs >50% of time during >50% of typical usage scenarios

High (H)
 Occurs >50% of time for >10% of scenarios

Low (L)
 Occurs >10% of time for >10% of scenarios

Very Low (VL)
 Occurs <10% of time for <10% of scenarios

Michaelmas 2007 SOFT3302 Lecture 9 5

Damage: Users/Customers

Endangers safety of human lives
 Patient monitoring system; aeronautical/aerospace; weapons systems

Affects accomplishment of essential organisational function and no replacement capability available
 E-business; nationwide warehouse inventory

Affects the functioning of firmware causing malfunction of entire system
 Household appliances; cars; computerised equipment

Michaelmas 2007 SOFT3302 Lecture 9 6

Damage: Users/Customers

Affects accomplishment of an essential organisational function but replacement is available

- Front desk sales system with manual backup*

Affects proper functioning of software packages for business applications

- Slow response time for POS application; because of a fault information normally on one screen must be sourced from three*

Affects the proper functioning of software for a private customer

- Computer games; educational software; word processors*

Michaelmas 2007 SOFT3302 Lecture 9 7

Damage: Users/Customers

Affects functioning of a firmware application but without affecting entire system

- Blackout of appliance control board but without harming functionality; failure of secondary system, e.g. outside temperature display*

Inconveniences the user but does not prevent accomplishment of the system's capabilities

- Distorted but not misleading displays; inability to produce the output listed although alternative routes to obtaining the required information or performing the same operation are available*

Michaelmas 2007 SOFT3302 Lecture 9 8

Damage: Developer

Financial losses

- Damages paid for physical injuries*
- Damages paid for malfunctioning software*
- Purchase cost reimbursed to customers*
- High maintenance expenses for repair of failed systems*

Non-quantitative damages

- Expected to affect future sales*
- Substantially reduced current sales*

Michaelmas 2007 SOFT3302 Lecture 9 9

Issues Affecting Likelihood

Module/application issues

- Magnitude
- Complexity and difficulty
- Percentage of original software (vs reuse)

Programmer issues

- Professional qualifications
- Experience with the module's specific subject matter
- Availability of professional support (backup of knowledge and experience)
- Acquaintance with the programmer and the ability to evaluate his capabilities

Michaelmas 2007 SOFT3302 Lecture 9 10

Combining ratings

$R = S + L$
 $R = k \times S + m \times L$
 $R = S \times L$

Michaelmas 2007 SOFT3302 Lecture 9 11

Example: Super Teacher

Software package designed to support teachers in managing the grades of elementary school pupils

- Package includes 8 applications
- Applications 7 & 8 based on high percentages of reused code
- Application 2 was developed by a team of new employees

How to plan/allocate testing resources?

Michaelmas 2007 SOFT3302 Lecture 9 12

Example: Super Teacher

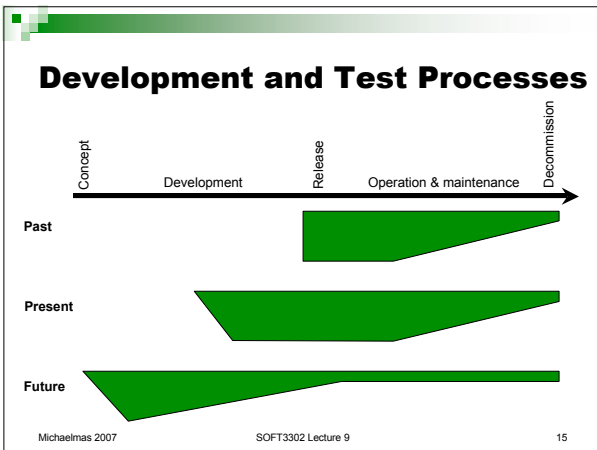
Application	Sev.	L'hood	Combined rating method		
			S + L	7S + 2L	S x L
Input of test results	3	2	5 (4*)	25 (5)	6 (4)
Interface for input and output of pupil's data to and from other teachers	4	4	8 (1)	36 (1)	16 (1)
Preparation of lists of low achievers	2	2	4 (6*)	18 (7)	4 (5*)
Printing letters to parents of low achievers	1	2	3 (8)	11 (8)	2 (8)
Preparation of reports for school principal	3	3	6 (3)	27 (4)	9 (3)
Display of a pupil's achievements profile	4	3	7 (2)	34 (2)	12 (2)
Printing of pupil's term report card	3	1	4 (6*)	23 (6)	3 (7)
Printing of pupil's year-end report card	4	1	5 (4*)	30 (3)	4 (5*)

Michaelmas 2007 SOFT3302 Lecture 9 13

Test Process Outline

Software Development and Test Process
 IEEE 829 Software Test Documentation
 Test Process Improvement

Michaelmas 2007 SOFT3302 Lecture 9 14

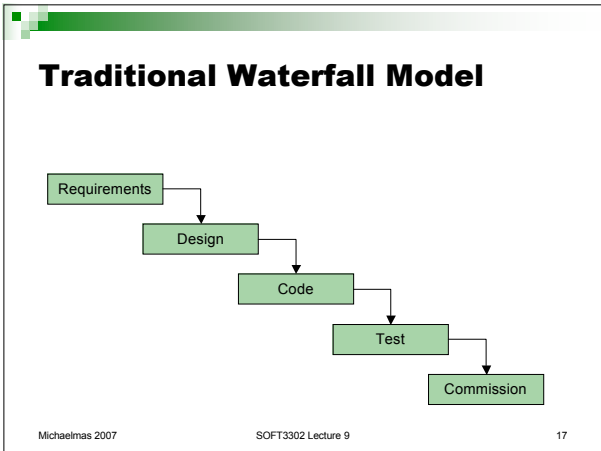


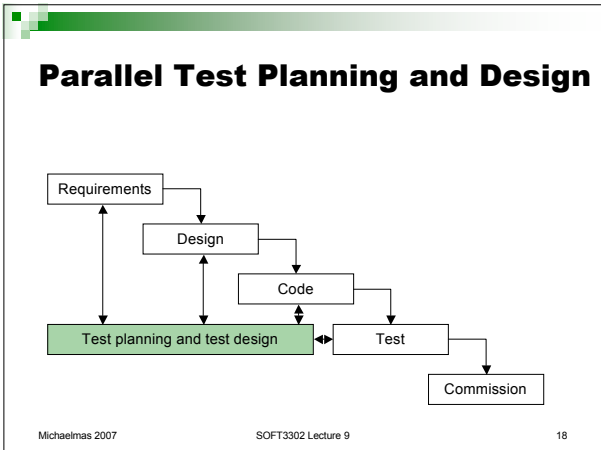
Fantasy Application Development

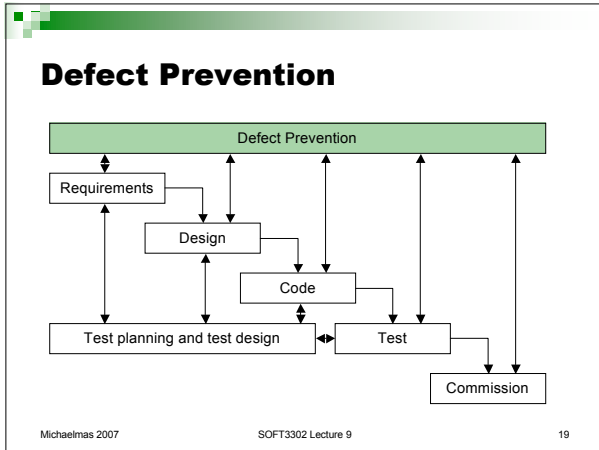
1. Announce product and release date
2. Design logo and make t-shirts
3. Determine what the product is
4. Estimate development time
5. Write the code and web pages
6. Write the specification (optional)
7. Beta release (ready or not)
8. Give incomplete version to test
9. Announce upgrade programme and patches

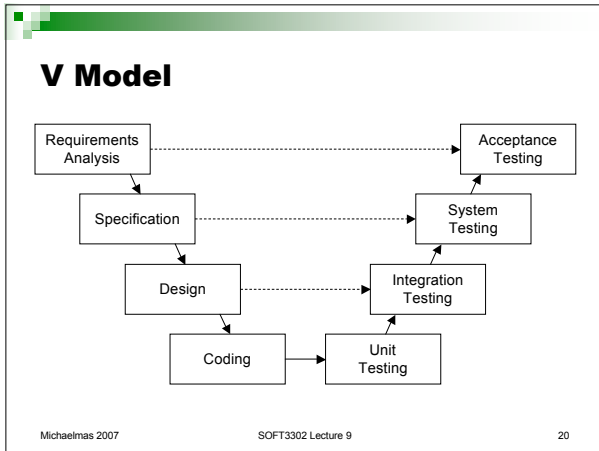
[Pop99]

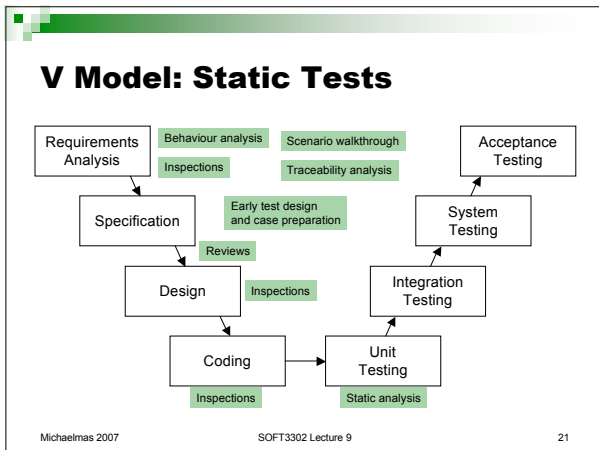
Michaelmas 2007 SOFT3302 Lecture 9 16

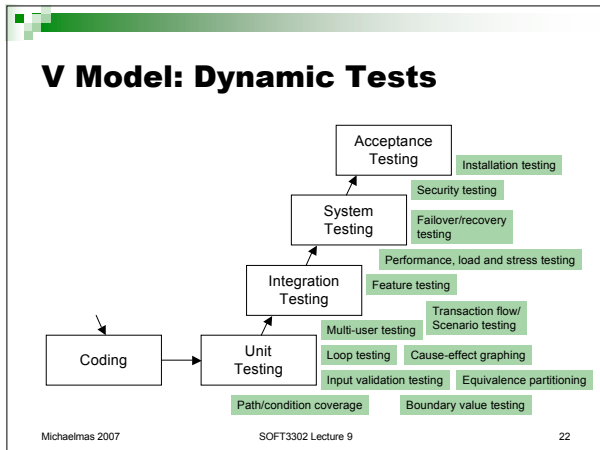












Extreme Programming

Significant departure from traditional software development methods

Proposes improvements in software projects

- Communication*
- Simplicity*
- Feedback*
- Courage*

Michaelmas 2007 SOFT3302 Lecture 9 23

- ### Core Practices of XP
1. On-site customer
 2. Small releases
 3. Planning
 4. Metaphor
 5. YAGNI
 6. Pair programming
 7. Collective ownership
 8. Continuous integration
 9. Coding standards
 10. Testing
 11. Refactoring
 12. Standard week
- Michaelmas 2007 SOFT3302 Lecture 9 24

XP: Unit Testing

Unit testing recommended for all development

- Unit must pass 100% of unit tests prior to release for integration
- Unit tests are developed prior to the actual coding of the unit
- Debugging is accomplished by adding tests to the unit tests

Creators of open source test framework *sunit*

- Extended by others to Java, C++, Perl, Python, HTTP
- www.xprogramming.com 'software' section

Michaelmas 2007 SOFT3302 Lecture 9 25

XP: Acceptance Testing

Requirements tested in a release should be acceptance tested in that release

- Quick feedback

Testers should prepare for testing with the customers and programmers while the development is underway

- Planning
- Customer feedback

Michaelmas 2007 SOFT3302 Lecture 9 26

Cleanroom

Developed by IBM in the 1980's

Designed to support the measurement and certification of MTBF

- Incidental goal was increase in MTBF

Defect prevention vs defect detection

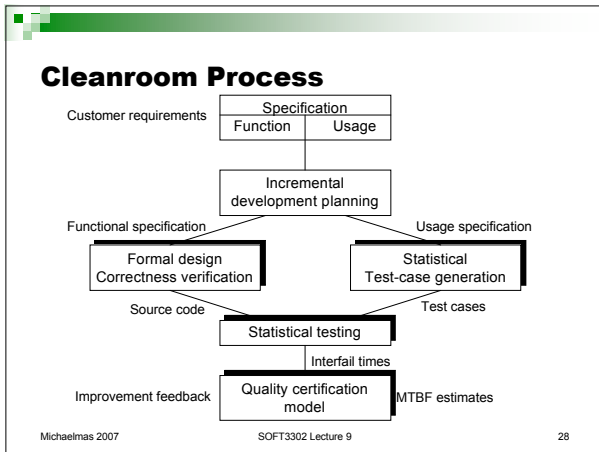
- Avoid entry of defects during development, hence 'cleanroom'

Discipline in development process

- Integrating formal methods for specification and design
- Non-execution based programme development
- Statistically-based independent testing

Incremental development life cycle

Michaelmas 2007 SOFT3302 Lecture 9 27



- ### Integrated Formal Methods
- Functional specification
- Entire product specified in an appropriate mathematical formalism
 - Specification is subject to inspections to reduce errors
- Design and verification
- Stepwise abstractions
 - Correctness conditions
 - Correctness reviewed by team
- Michaelmas 2007 SOFT3302 Lecture 9 29

- ### Rôle of Review and Testing
- Heavily dependent on inspection
- Inspection is the primary means for the developer to verify correctness
 - Much more cost effective than testing software
- Non-execution-based development
- Developers not allowed to test or debug their programmes
 - Testing process completely separated
 - Correctness achieved through
 - Stepwise abstraction
 - Code inspections
 - Group walk-throughs
 - Formal verification
- Michaelmas 2007 SOFT3302 Lecture 9 30

Rôle of Review and Testing

Statistically-based independent testing

- Testers simulate the operational environment with random testing
- Process includes
 - Definition of frequency distribution of inputs to system
 - Frequency distribution of different system states
 - Expanding range of developed system capabilities
- Test observations determine objective measure of reliability
 - Projection for MTBF
- In addition, a limited number of test cases ensure correct operation for catastrophic failure
- Reliability assessment rather than error detection

Michaelmas 2007 SOFT3302 Lecture 9 31

IEEE 829

Preparation of tests

- Test Plan: Plan how the testing will proceed
- Test Design Specification: Decide what needs to be tested
- Test Case Specification: Create the tests to be run
- Test Procedure: Describe how the tests are run
- Test Item Transmittal Report: Specify the items released for testing

Running the tests

- Test Log: Record the details of tests in time order
- Test Incident Report: Record details of events that need to be investigated

Completion of testing

- Test Summary Report: Summaries and evaluate tests

Michaelmas 2007 SOFT3302 Lecture 9 32

Test Documentation

Good software projects are dependent on good communication between stakeholders

```

graph TD
    TD[Test Documentation] <--> PD[Project Director]
    TD <--> C[Customer]
    TD <--> U[Users]
    TD <--> T[Testers]
    TD <--> TM[Test Manager]
    TD <--> D[Developers]
    PD <--> C
    C <--> U
    U <--> T
    T <--> TM
    TM <--> D
    D <--> PD
  
```

Michaelmas 2007 SOFT3302 Lecture 9 33

Test Documentation

Trade-off between big wall of paper and traceable decision support

Records

- What needs to be done*
- What has been accomplished*

Protection against litigation

Not necessarily a written document, e.g. documentation database

Most effective when traceable and auditable

Effort and commitment is required to keep everything up-to-date

Michaelmas 2007 SOFT3302 Lecture 9 34

Goals of the Test Process

Develop a support system that can assist testers through the testing project

Coordinate the testing project through development and management of test documentation

Provide varying levels of support

- Little documentation requirements through to formal documentation requirements*

Michaelmas 2007 SOFT3302 Lecture 9 35

Objectives of Test Documentation

Define overall testing strategy for a project

Plan how each level of testing is to be conducted

Identify what tests are to be carried out

Define how each test is to be performed

Record execution of each test

Track incidents that arise during testing

Summaries outcomes of testing activities and make recommendations regarding quality and acceptability

Michaelmas 2007 SOFT3302 Lecture 9 36

Test Documentation Standard

IEEE provides comprehensive guidance on the structure of test documentation

Also MIL-STD-498

- Test plan*
- Test specification*
- Test report*

Michaelmas 2007 SOFT3302 Lecture 9 37

Test Plan

Test-plan identifier

- Unique identifier assigned to plan*

Introduction

- Summarise what's in the plan, what are the main issues that a reader should look at in more detail if they were to pick up the plan, and provide reference to other documents*

Test items

- Identifies the components to be tested, including specific versions of variants*

Features to be tested

- Those aspects of the system that will undergo test*

Features not to be tested

- Aspects of system not being tested and reasons why they can be ignored*

Michaelmas 2007 SOFT3302 Lecture 9 38

Test Plan

Approach

- General approach to testing each feature to be tested*

Item pass/fail criteria

- Criteria to determine whether each test item passed or failed testing*

Suspension criteria and resumption requirements

- Identifies the conditions under which testing can be suspended and what testing activities are to be repeated if testing resumes*

Test deliverables

- Describes the documentation that represents all the testing activities applied to the test items covered under this plan*

Testing tasks

- Identifies all the tasks required to complete testing, including any dependencies between tasks, or special skills required for tasks*

Michaelmas 2007 SOFT3302 Lecture 9 39

Test Plan

Environmental needs

- Describes the test environment, including any special hardware facilities, software facilities and support tools

Responsibilities

- Groups responsible for managing, designing, preparing, executing, witnessing, checking and resolving

Staffing and training needs

- Specifies who is available to carry out the testing tasks, their skill level requirements and any special training needs

Schedule

- Proposed test milestones and events and proposals for task coordination and effort estimations

Risks and contingencies

- Identifies any high risk assumptions of the plan and contingencies for each risk proposed

Approvals

- Sign-off requirements for the plan

Michaelmas 2007 SOFT3302 Lecture 9 40

Test Plans

IEEE format may appear daunting

- Format addresses main test planning issues
- Worth the effort of completing each section

Amount of detail in each section may reflect

- Level of use of the plan
- Experience of the reader

Development of 'boiler-plate' text

- Used within many plans

Michaelmas 2007 SOFT3302 Lecture 9 41

Test Design Specification

Detailing test conditions and the expected results as well as test pass criteria

- Test-design-specification identifier
- Features to be tested
- Approach refinements
- Test identification
- Feature pass/fail criteria

Michaelmas 2007 SOFT3302 Lecture 9 42

Test Case Specification

Specifying the test data for use in running the test conditions identified in the Test Design Specification

- Test-case-specification identifier*
- Test items*
- Input specifications*
- Output specifications*
- Environmental needs*
- Special procedural requirements*
- Intercase dependencies*

Michaelmas 2007 SOFT3302 Lecture 9 43

Test Procedure Specification

Detailing how to run each test, including any set-up preconditions and the steps that need to be followed

- Test-procedure-specification identifier*
- Purpose*
- Special requirements*
- Procedure steps*
 - Log; setup; start; proceed; measure; shutdown; restart; stop; wrap-up; contingencies*

Michaelmas 2007 SOFT3302 Lecture 9 44

Test Item Transmittal Report

Reporting on when tested software components have progressed from one stage of testing to the next

- Transmittal-report identifier*
- Transmitted items*
- Location*
- Status*
- Approvals*

Michaelmas 2007 SOFT3302 Lecture 9 45

Test Log

Recording which tests cases were run, who ran them, in what order, and whether each test passed or failed

- *Test-log identifier*
- *Description*
- *Activity and event entries*
 - Execution description; procedure results; environmental information; anomalous events; incident report identifiers

Michaelmas 2007 SOFT3302 Lecture 9 46

Test Incident Report

Detailing, for any test that failed, the actual versus expected result, and other information intended to throw light on why a test has failed

- *Test-incident-report identifier*
- *Summary*
- *Incident description*
- *Impact*

Michaelmas 2007 SOFT3302 Lecture 9 47

Test Summary Report

A management report providing any important information uncovered by the tests accomplished, and including assessments of the quality of the testing effort, the quality of the software system under test, and statistics derived from Incident Reports.

The report also records what testing was done and how long it took, in order to improve any future test planning. This final document is used to indicate whether the software system under test is fit for purpose according to whether or not it has met acceptance criteria defined by project stakeholders

Michaelmas 2007 SOFT3302 Lecture 9 48

Test Summary Report

- *Test-summary-report identifier*
- *Summary*
- *Variations*
- *Comprehensiveness assessment*
- *Summary of results*
- *Evaluation*
- *Summary of activities*
- *Approvals*

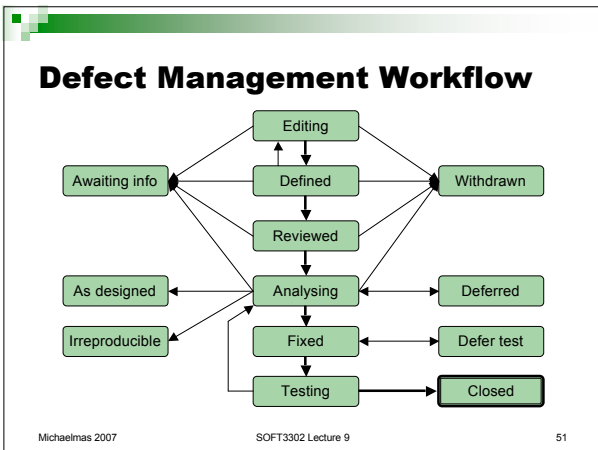
Michaelmas 2007 SOFT3302 Lecture 9 49

Test Process

Use documentation as a basis for identifying the process

Use status as a basis for workflow

Michaelmas 2007 SOFT3302 Lecture 9 50



Support

Tracking and progress reporting at a fine-grained level

- Risks in the risk register
- Features in a test plan
- Test objectives and tests cases in the test design
- Test procedures
- Test logs
- Defects

Reports can summarise progress based on fine-grained work items

- Overview of incremental progress
- Know what is yet to be done

Michaelmas 2007 SOFT3302 Lecture 9 52

Detailed Progress Reports

Precise indication of outstanding work
Good for day-to-day coordination
The kind of thing to avoid giving senior management

- Too much irrelevant detail

Michaelmas 2007 SOFT3302 Lecture 9 53

Summary Progress Reports

Trends and comparisons
Compare results from one report to a previous report
Deliverable statuses

Michaelmas 2007 SOFT3302 Lecture 9 54

Other Deliverables

Status not the only attribute to track and report
Project management attributes

- Assignee
- % complete
- Estimated effort until completion
- Accumulated effort

Other testing specific

- Outcome of test
- Risk level
- Applicable function or component

Combine attributes and link deliverables

- # times executed, status, effort (estimated and actual)
- Links, e.g. transmittal procedures, logs

Michaelmas 2007 SOFT3302 Lecture 9 55

Support Framework

Tools are needed for efficient support

- Static documents unworkable

Range of support

- Spreadsheets
- Database
- Custom tools
- More complex and needs increase
 - Control over status changes
 - Comparative reports require history
- Build vs buy
 - Cost to build; expertise required; meets needs
 - Inflexibility of OTS software products

Michaelmas 2007 SOFT3302 Lecture 9 56

Process Support

Process support need to be in place before automated tests

- Automation turns chaos into automated chaos

Michaelmas 2007 SOFT3302 Lecture 9 57

Process Improvement

Unchanged process will gradually improve

- *Familiarity: smoother subsequently*
- *Experience: pitfalls avoided subsequently*

Structural improvement consists of

- *Determine target and area of consideration*
- *Determine current situation*
- *Determine required situation*
- *Implement changes*

Michaelmas 2007 SOFT3302 Lecture 9 58

Software Process Improvement

A number of models have emerged

- *Capability Maturity Model (CMM)*
 - <http://en.wikipedia.org/wiki/CMM>
- *ISO 15504: Software Process Improvement and Capability dEtermination (SPICE)*
 - http://en.wikipedia.org/wiki/ISO_15504
- *TickIT*
 - <http://www.tickit.org/>

Michaelmas 2007 SOFT3302 Lecture 9 59

Capability Maturity Model (CMM)

Continuously improving processes <1%

Optimising (5)

Managed (4) Predictable processes 2%

Repeatable (3) Standard, consistent processes 13%

Defined (2) Disciplined processes 24%

Initial (1) Ad hoc chaotic processes 60%

No requirements for testing

Tough requirements for testing

Michaelmas 2007 SOFT3302 Lecture 9 60

What about *Testing Maturity?*

Testing processes needs to be improved as well as software development processes

Michaelmas 2007 SOFT3302 Lecture 9 61

Test Process Improvement Models

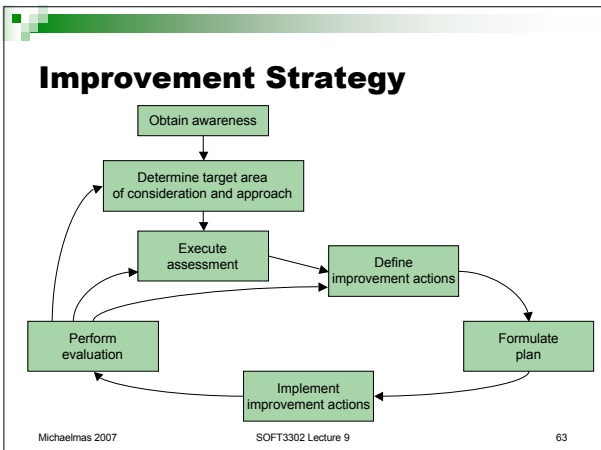
Software process improvement models deal cursorily with testing

- *Treated as a single issue or as a single step in the process*

Approaches emerging for process improvement of software testing

- *Testability Maturity Model (TMM): Gelperin*
- *Test Improvement Model (TIM): Ericson, Subotic, Ursing*
- *Testing Maturity Model (TMM): Illinois Inst. of Tech.*
- *Test Organisation Maturity (TOM): Gerrard*
- *Test Process Improvement (TPI): IQIP*

Michaelmas 2007 SOFT3302 Lecture 9 62



Continuous Improvement

Integration of time-tracking

- Time records on test procedures*
- Compare weekly tester effort breakdowns*

Enhanced management reports

- Adding management guidance information to reports*
- Automatic report distribution to staff to minimise coordination effort*

Michaelmas 2007 SOFT3302 Lecture 9 64

Evolution

Reports to support regression testing

- What procedures have failed most?*
- What procedures haven't been run for a long time?*

Reports to analyse quality of products tested

- Rate of defect discovery (defects/hr)*
- Feedback on whether products are improving*

Michaelmas 2007 SOFT3302 Lecture 9 65

Evolution

Reports to analyse productivity

- Rate of defect discovery (defects/hr)*
- Test procedures executed/hr*
- Ratio of test execution time to other activities, e.g. design, installation, &c.*

Introduce improvements judiciously

- Perhaps what for start of next project phase before introducing changes*

Michaelmas 2007 SOFT3302 Lecture 9 66

Outcomes

Objectives become clearer
Coordination and delegation improves
Evidence to back up reports to management
Data management becomes an issue

- *Data overload! Too much data*
- *Who can access the data?*
- *Who can change the data?*

Michaelmas 2007 SOFT3302 Lecture 9 67

References

[Bin99] Robert V. Binder, *Testing Object-Oriented Systems: Models, Patterns and Tools*, Addison-Wesley, 1999.

[Gal04] Daniel Galin, *Software Quality Assurance: From theory to implementation*, Pearson Addison-Wesley, 2004.

[Pop99] G. Pope, *You want it when? Estimating and negotiating test schedules*, STAR-West, 1999. <http://www.stickyminds.com/getfile.asp?ot=XML&id=2092&fn=XDD2092filelistfilename1.pdf>

http://en.wikipedia.org/wiki/IEEE_829

Some material adapted from 2004 SOFT3103 course.

Michaelmas 2007 SOFT3302 Lecture 9 68
