同济大学软件学院

School of Software Engineering, Tongji University

Software Testing Foundation Level

Chapter 1:Foundamentals of Testing

刘琴 Qin Liu

Hello!

I am Qin Liu:

- **The course instructor**
- **A professor whose research interests are in heterogonous data analysis models(to applying DLN to big data in multi-disciplines).**
- A registered Lecturer of the Higher Education Academy UK / ISO29119 Co-editor / ISTQB National Representative Greater China / TMMi CN President

You can find me at qin.liu@tongji.edu.cn

Your TA: TBA

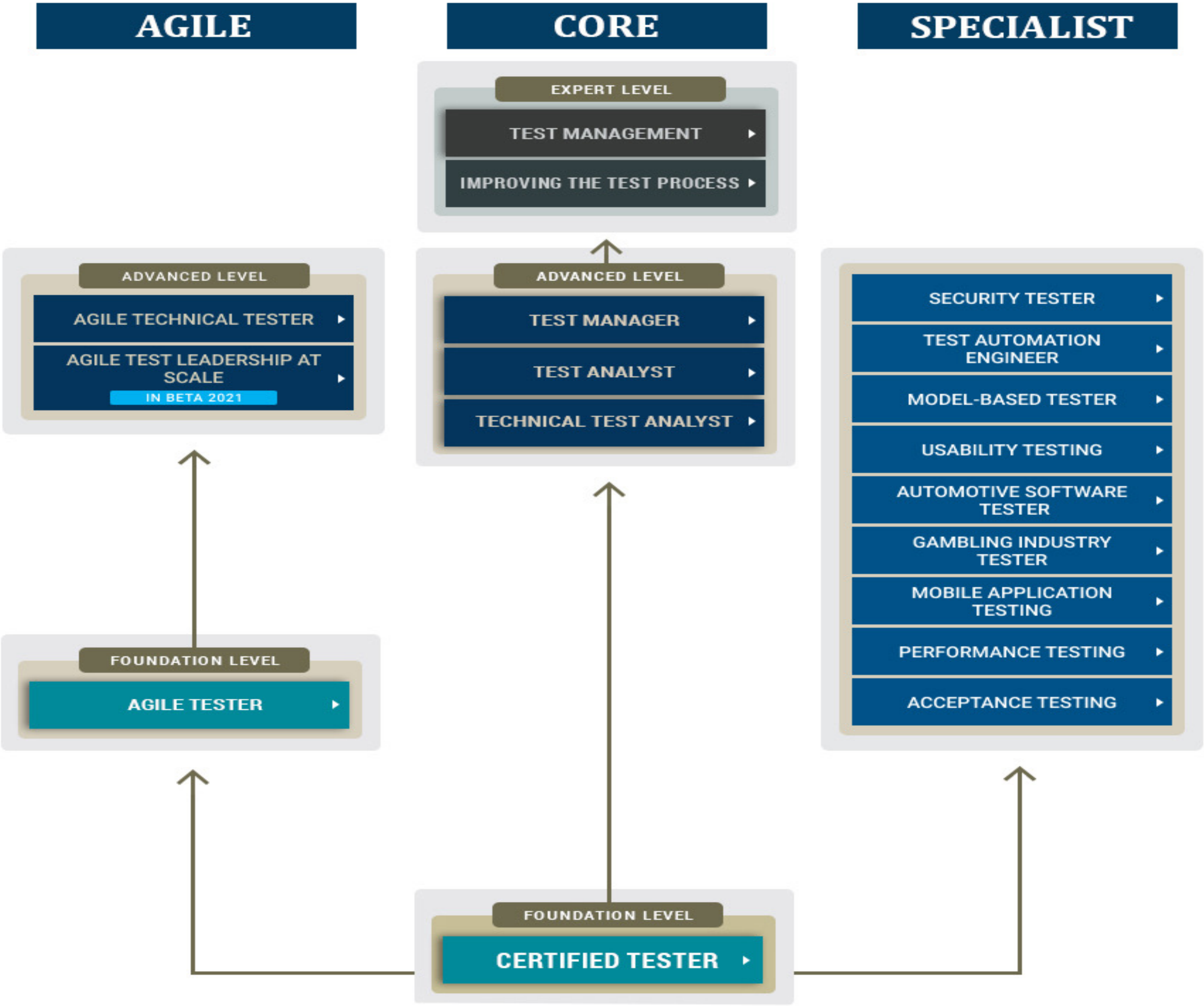
Course materials: http://canvas.tongji.edu.cn/courses/70690)

Software Testing Foundation Level

- **1 Fundamental of testing**
- **2 Testing throughout the life cycle**
- **3 Static techniques** ***(Assignment 1)***
- **4 Test Design Techniques**
- [Extension]: Risk Based Testing Analysis and Design
- [Extension]: Usability Testing ***(Assignment 2.2)***
- **5 Test management**
- **6 Tool support for testing**
- [Extension]: JUnit/Selenium

Assignment 2.1 & 3

ISTQB®
International Software
Testing Qualifications Board



AGILE

ADVANCED LEVEL

- AGILE TECHNICAL TESTER
- AGILE TEST LEADERSHIP AT SCALE (IN BETA 2021)

FOUNDATION LEVEL

- AGILE TESTER

CORE

EXPERT LEVEL

- TEST MANAGEMENT
- IMPROVING THE TEST PROCESS

ADVANCED LEVEL

- TEST MANAGER
- TEST ANALYST
- TECHNICAL TEST ANALYST

FOUNDATION LEVEL

- CERTIFIED TESTER

SPECIALIST

- SECURITY TESTER
- TEST AUTOMATION ENGINEER
- MODEL-BASED TESTER
- USABILITY TESTING
- AUTOMOTIVE SOFTWARE TESTER
- GAMBLING INDUSTRY TESTER
- MOBILE APPLICATION TESTING
- PERFORMANCE TESTING
- ACCEPTANCE TESTING

1 Fundamentals of Testing

1. Why is Testing Necessary?
2. What is Testing?
3. Seven Principles of Testing
4. Fundamental Test Processes
5. The Psychology of Testing
6. Code of Ethics

Software Testing Foundation Level

Chapter 1:Foundamentals of Testing

1.1 & 1.2: Why & What is Testing?

刘琴 Qin Liu

Why & What Testing

1. Why & What is Testing?
2. What is a bug?
3. How much testing is enough? (A tradeoff: effectiveness & efficiency)
4. What are test levels?
5. Testing is far beyond test execution…

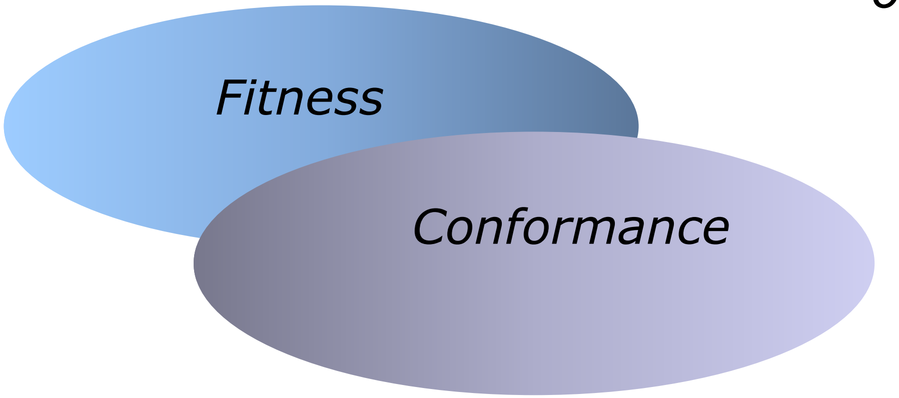
□ Chap1Sec1Ex1: The Triangle Test

Why Testing?

Concepts/Key Words

- Software Quality

- How to Assure the Quality?



Fitness

Conformance

Q & A

<u>What does QUALITY mean to you? </u>"Fitness for use" vs. "Conformance to requirements"

<u>How to define the QUALITY?</u>

A mobile vs. Its applications

e.g., an iPhone vs. WeChat

Other Issues (Risks)
Features: Right set
Schedule: Quickly enough
Budget: Acceptably cheap
Quality: Ready for customers/release/next step

Ref: ISO 9126 Quality Standard

Characteristic

Subchar-acteristics

- Functionality: Suitability, accuracy, interoperability, security, compliance
- Reliability: Maturity (robustness), fault-tolerance, recoverability, compliance
- Usability: Understandability, learnability, operability, attractiveness, compliance
- Efficiency: Time behavior, resource utilization, compliance
- Maintainability: Analyzability, changeability, stability, testability, compliance
- Portability: Adaptability, installability, co-existence, replaceability, compliance

Addressed by functional tests

Addressed by non-functional tests

What is Testing?

Concepts/Key Words

- Hunting Bugs?!
- Preventing Risks!
- Be in Control?!

- Failure
- Bug ≈ Defect
- Incident
- Fault ≈ Error ≈ Mistake

Think

- Bugs occur due to ...
  - programmer, analyst, and other individual contributor (including tester) fallibility
  - time pressure
  - complexity of the code, infrastructure, or problem to be solved
  - changing and meshing technologies
  - many system interactions.
- Failures occur due to bugs and…
  - environmental conditions
  - misuse (deliberate and accidental)

What is a bug?

Concepts/Key Words

- Code (Business Logic and Interface)
- In documents
- A design diagram

The First Computer Bug

Photo # NH 96566-KN First Computer "Bug", 1945

[illegible]

Relay #70 Panel F (moth) in relay.

First actual case of bug being found.

[illegible]

In 1947, Grace Murray Hopper was working on the Harvard University Mark II Aiken Relay Calculator (a primitive computer).

Defect(bug): A flaw in a component or system that can cause the component or system to fail to perform its required function

Threat of Bugs

Company

- *Damaged reputation for quality*
- *High or unpredictable maintenance costs*
- *Unexpected delays in release cycles*
- *Lack of confidence in system*
- *Lawsuits*

Environment

- *Pollution*
- *Waste*

People, Societies, and states

- *Lost jobs*
- *Lost lives*
- *Lost rights*
- *Lost missions*
- *Lost wars*

Case of Software Failure

NASA Mars Climate Orbiter
the $125-million spacecraft has not been heard from since
Incident Date: 9/23/1999

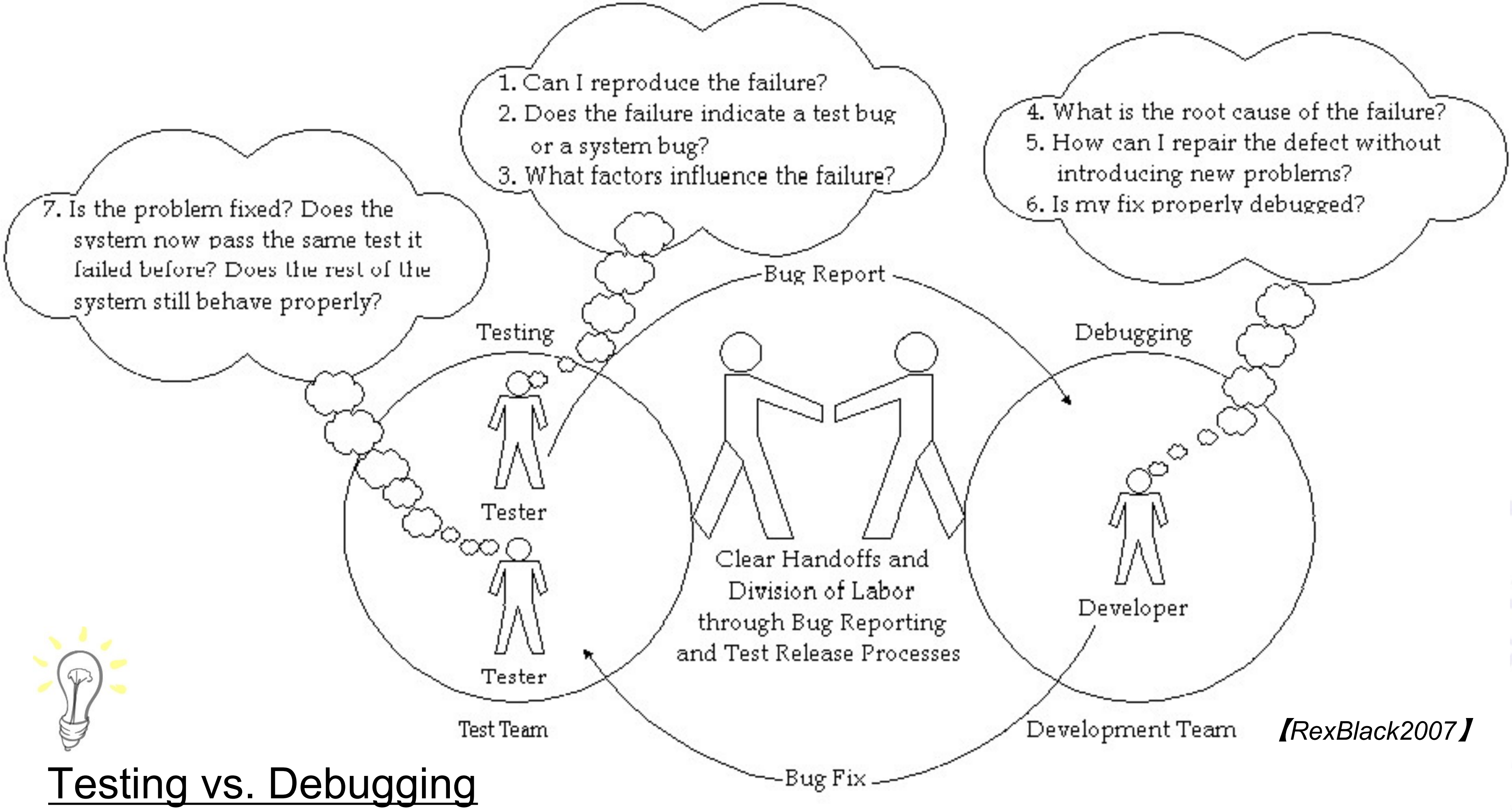
Air-Traffic Control System in LA Airport
Lost contact with 400 airplanes
Incident Date: 9/14/2004

http://www.cse.lehigh.edu/~gtan/bug/softwarebug.html

Where Bugs Come From?

- *Invalidity of communication*
- *Imperfect definition of requirement analysis*
- *Deviation of requirement*
- *Logistical mistake of design*
- *Not obey the rule with document and code*
- *Procedure error*
- *Code mistake*
- *Document mistake*

Find-Debug-Confirm


1. Can I reproduce the failure?
2. Does the failure indicate a test bug or a system bug?
3. What factors influence the failure?
4. What is the root cause of the failure?
5. How can I repair the defect without introducing new problems?
6. Is my fix properly debugged?
7. Is the problem fixed? Does the system now pass the same test it failed before? Does the rest of the system still behave properly?
Bug Report
Testing
Debugging
Tester
Tester
Developer
Clear Handoffs and Division of Labor through Bug Reporting and Test Release Processes
Test Team
Development Team
Bug Fix

【RexBlack2007】

Testing vs. Debugging

Test Levels?

Concepts/Key Words

- Unit/Component Test:

individual pieces before fully integration

- Integration/String Test:

relationships and interfaces between pairs and groups of components

- System Test:

overall and particular behaviors, functions, and responses

- Acceptance/Pilot Test:

demo, deployment/release

- Maintenance Test:

during development of the changes

- Operational Test:

in the operational environment (e.g., reliability or availability)

Beyond Test Execution

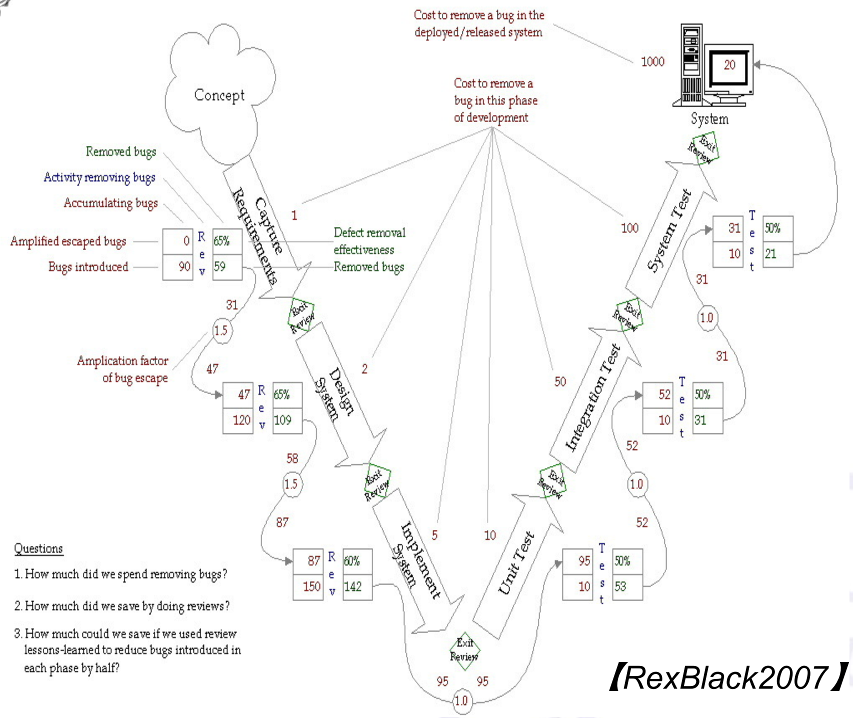
Concepts/Key Words

- Planning and control
- Choosing test conditions
- Design test cases
- Checking test results
- Evaluating exit criteria
- Test result reporting
- Closure/end-of-test tasks

- Validation and Verification

  These terms are most meaningful in the context of the entire development or maintenance process, not just the testing process.

Think



Questions

1. How much did we spend removing bugs?
2. How much did we save by doing reviews?
3. How much could we save if we used review lessons-learned to reduce bugs introduced in each phase by half?

【RexBlack2007】

How Much Testing is Enough?

Concepts/Key Words

- How much is enough?
  - Effectiveness vs. Efficiency
- How to Measure?
- How to Assure? (monitoring and control, validation and verification)

Think

- Effectiveness vs. Efficiency
  - Find common bugs (the bugs cannot be ignored)
  - By as few as possible test

  Mission: Select or design a limited number of test from a infinite set of possible test

- Validation and Verification

  These terms are most meaningful in the context of the entire development or maintenance process, not just the testing process.

Summary

- Quality vs. Testing
- Bug (where bugs come from? incident, failure, error, mistake, defect?)
- Testing vs. Debugging
- Test Level

Self Study

Reading

- Textbook Chapter 1 Section 1
- Glossary

  Debugging

  Requirement

  Review

  Test case

  Testing

  Test objective

Searching The Classic Triangle Test Design Examples in books /on internet

Triangle Test Scenario: Design Test for a program accepts three integers representing the lengths of a triangle’s sides. It outputs “scalene” (no equal sides), “isosceles” (two equal sides), or “equilateral” (three equal sides).

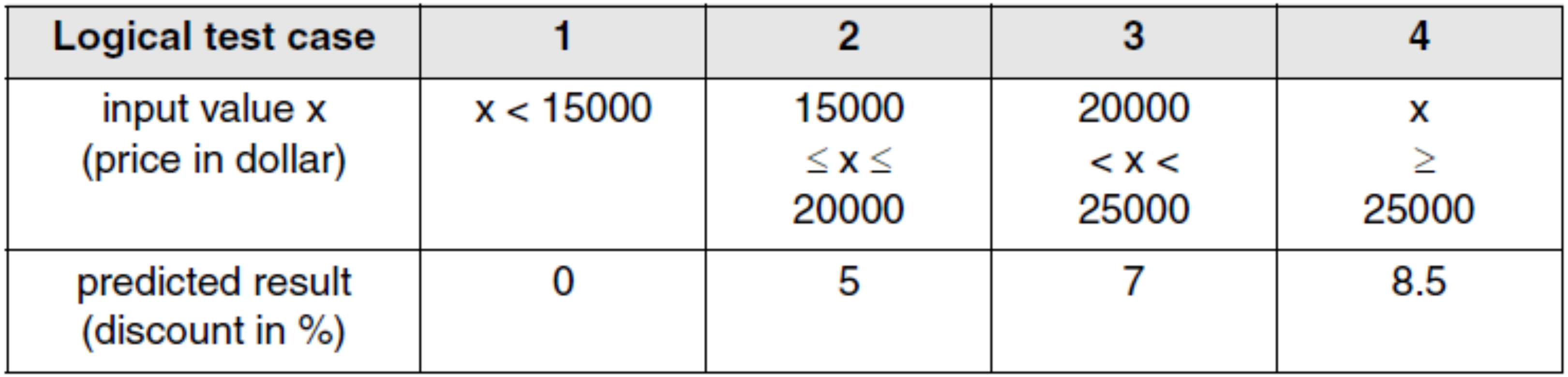
How many test cases would you design?

And WHY?

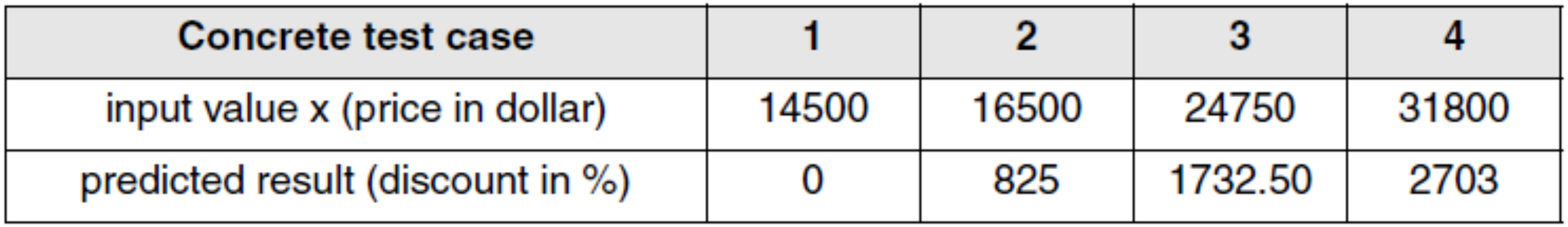
Exercise1.1.1 （A2.1.3）：The Triangle Tests

- A program accepts three integers representing the lengths of a triangle’s sides.
- It outputs “scalene” (no equal sides), “isosceles” (two equal sides), or “equilateral” (three equal sides).
- Write an effective (finds common bugs) and efficient (as few tests as possible) set of test cases. Usually, a test case consists of tester action, data, and expected result, but here the action is generally “input data”.
- Discuss.

Example: Logical vs. Concrete Test Cases



Logical test case	1	2	3	4
input value x (price in dollar)	$x < 15000$	$15000 \le x \le 20000$	$20000 < x < 25000$	$x \ge 25000$
predicted result (discount in %)	0	5	7	8.5



Concrete test case	1	2	3	4
input value x (price in dollar)	14500	16500	24750	31800
predicted result (discount in %)	0	825	1732.50	2703

Exercise1.1.2 （ A2.1.1） : A Marketing Survey

Conduct a survey by searching on the internet to find out the possible testing job positions in both Chinese and an oversea market (e.g., US）.

- What are the required key skills (from the job descriptions) ? And related salary range (compare with programmer)?
- What are the differences between the career ladders in testing area in two countries?
- Discuss.

1 Fundamentals of Testing

1. Why is Testing Necessary?
2. What is Testing?
3. Seven Principles of Testing
4. Fundamental Test Processes
5. The Psychology of Testing
6. Code of Ethics

1.3 Seven Testing Principles

1. Testing shows the presence of defects
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence-of-errors fallacy

P1: Testing Reveals Presence of Bugs

You have a very beautiful apple garden. But one day…

You see an eaten apple. And you know you have bugs in your garden.

Testing can reveal the presence of bugs, but cannot prove their absence

P2: Exhaustive Testing Is Impossible

☹ "Just make sure the software works before we ship it…"

- This charter is demonstrably impossible
  - The execution paths in non-trivial software are almost infinite
  - Large data flows separated across space (features) and time (static data)
  - Slight changes can cause regressions which are not linear to the size of the change
  - Myriad usage profiles and field configurations, some unknown and some unknowable
- Bottom line: Exhaustive testing (all combinations of inputs and preconditions) is not possible

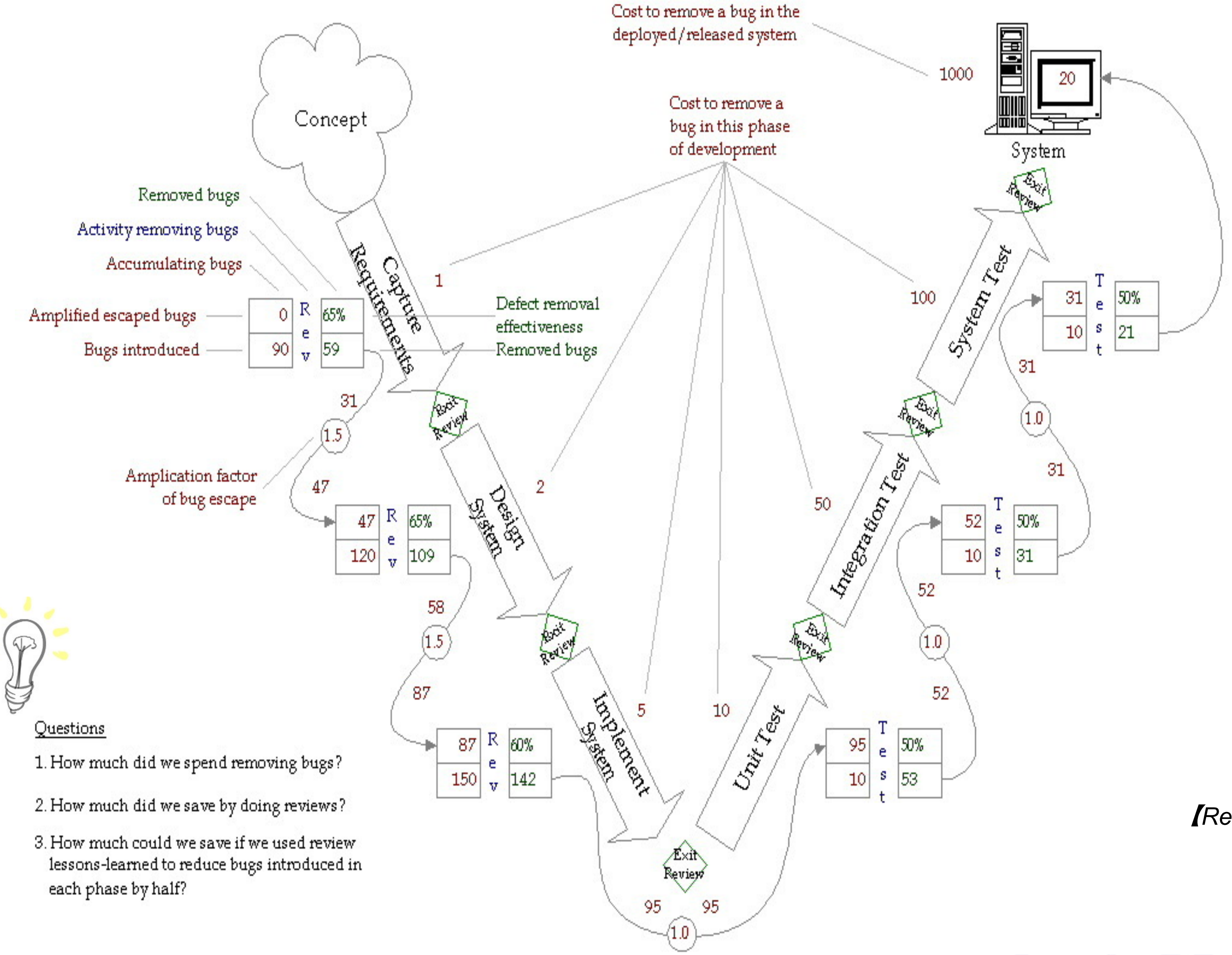
Defusing Exhaustive Testing Expectations

- Exhaustive testing as a way to prove the software works is a common (mis)expectation
- Bad expectations create problems for test professionals and test teams
  - Unachievable high demands on test group
  - Perception of incompetence when these demands aren’t met
- Testers must be ready to communicate (in words the project stakeholders will understand) how testing can contribute

P3: Early Testing
Benefits of Early QA and Testing

- The cost of a bug tends to increase as the project continues
- Most of the costs associated with pre-release bugs tend to be associated with the effort required to remove them, so the higher cost means longer schedules
- The more bugs enter a quality assurance or test activity, the more bugs will escape from that activity

P3: Early Testing. (A2.1.2)



Questions

1. How much did we spend removing bugs?
2. How much did we save by doing reviews?
3. How much could we save if we used review lessons-learned to reduce bugs introduced in each phase by half?

【RexBlack2007】

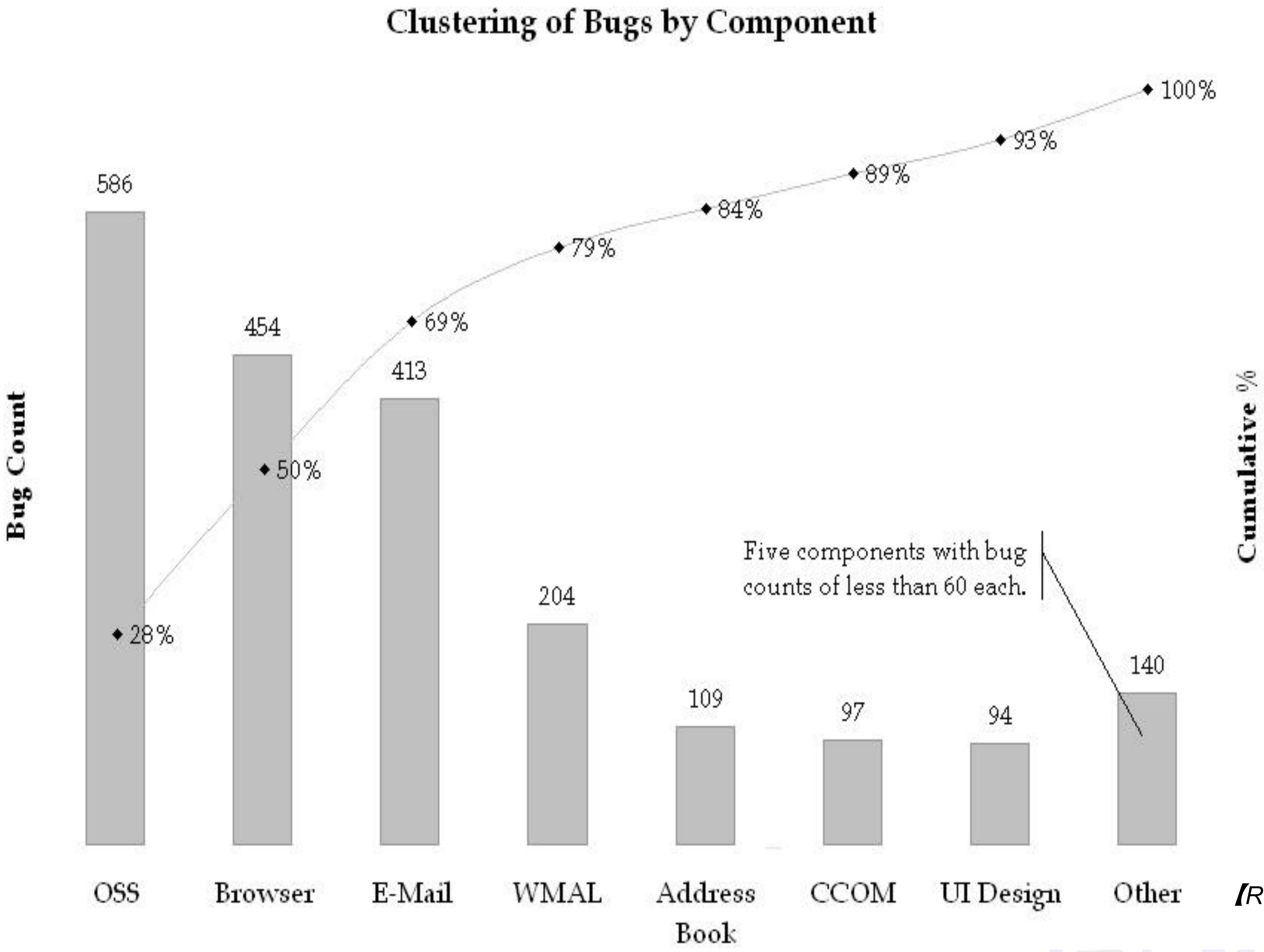
P3: Early Testing

- How much did we spend removing bugs?

- How much did we save by doing reviews

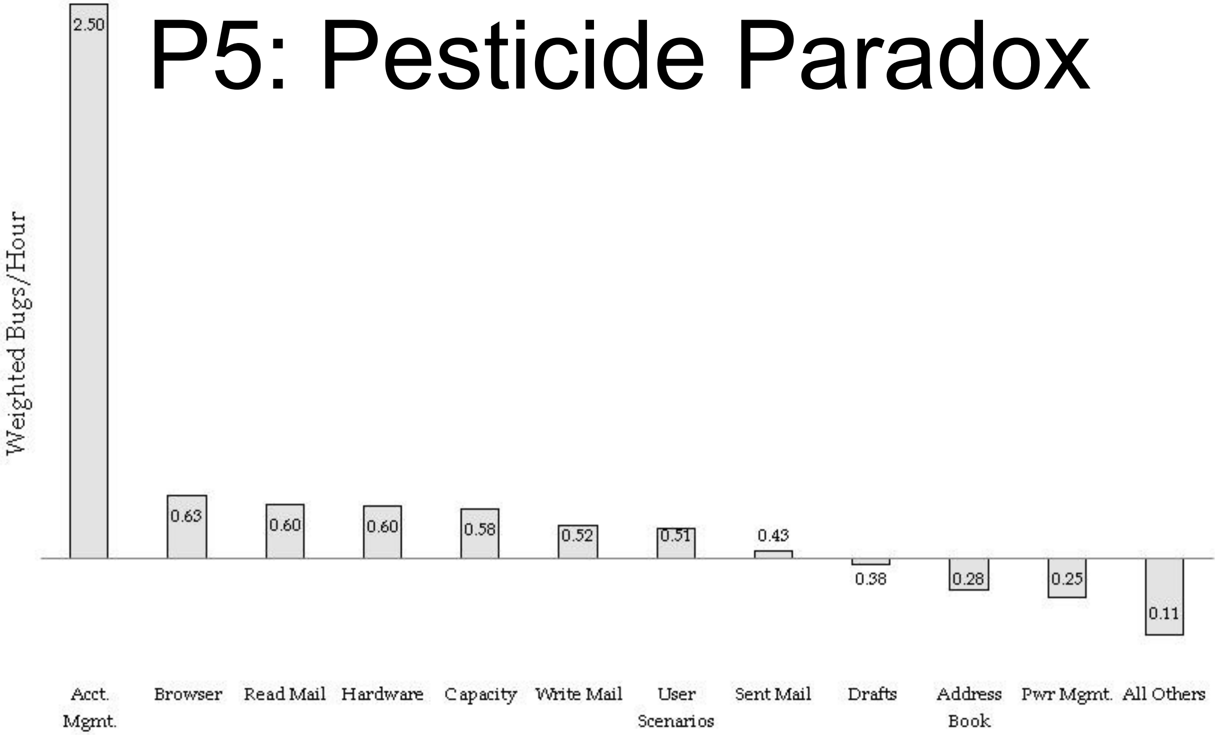
- How much could we save if we used review lessons-learned to reduce the bugs introduced in each phase by half?

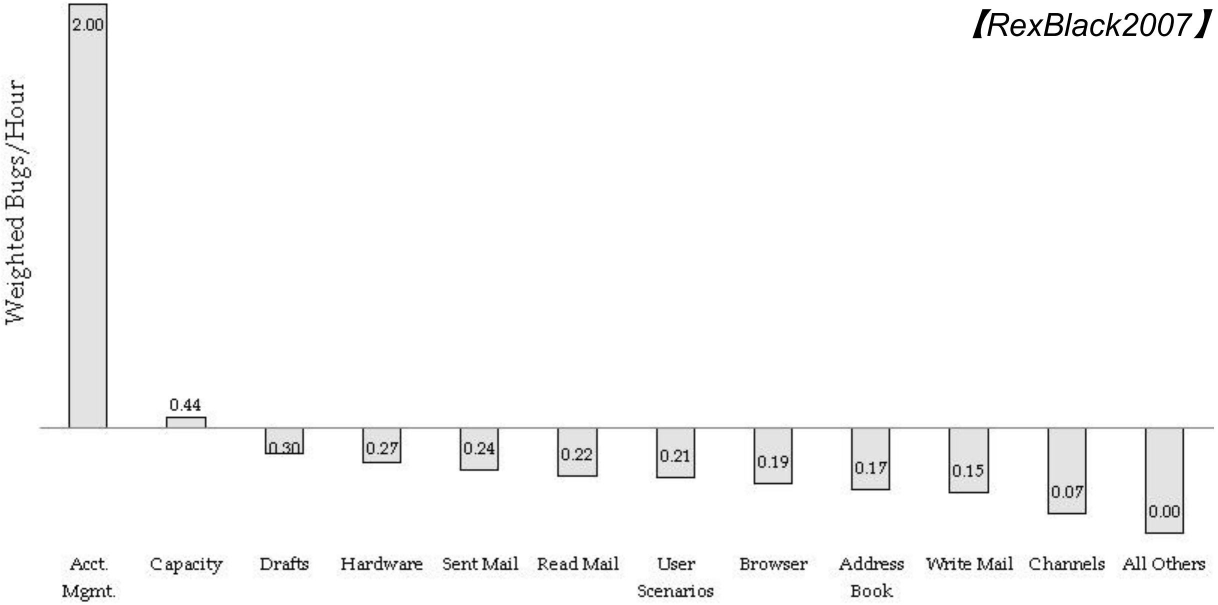
P4: Defect Clustering



【RexBlack2007】

P5: Pesticide Paradox





【RexBlack2007】

These two pictures compare efficiencies of the top 11 bug-finding test suites (out of 27). The top graph shows the first time the set of test suites was run. The bottom graph shows the results for the fifth time that same set of test suites was run. The average test suite efficiency in the first pass, 0.4, is used as the axis crossing point in both graphs.

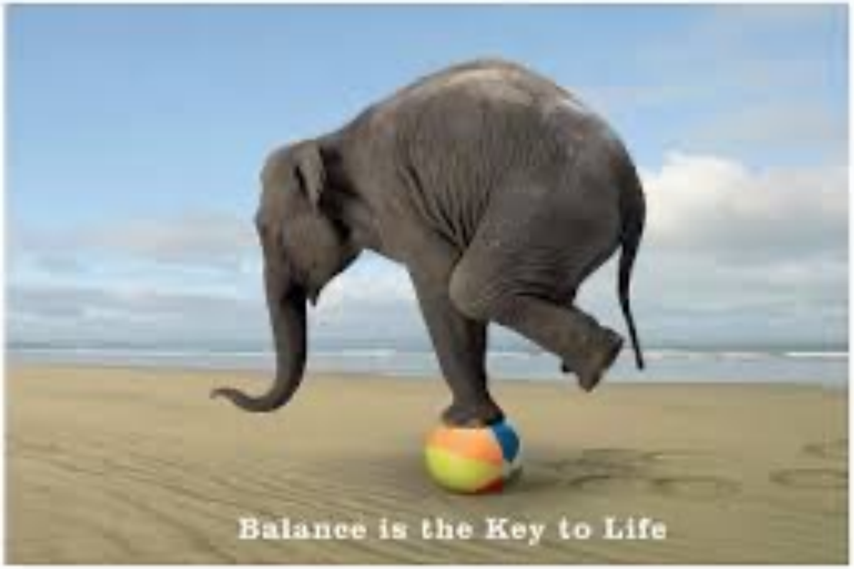
With one exception, the same test suites are in the top 11 test suites, which shows bug clustering again. However, all test suites are less effective after five executions.

P6: Testing is context depend
-*Testing Should Adapt to Needs*

- Different projects, organizations, and products have different testing needs
- A safety-critical software is tested differently from an e-commerce software
- *Always analyze your testing object before conduct actions*

P7: Absence-of-Errors Fallacy

- Finding and fixing many bugs does not guarantee user, customer, and/or stakeholder satisfaction
- Many low-defect products have failed in the market place
- Successful projects balance competing forces in terms of features, schedule, budget, and quality



1.4 Testing Principles Summary

1. Testing shows the presence of defects
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence-of-errors fallacy

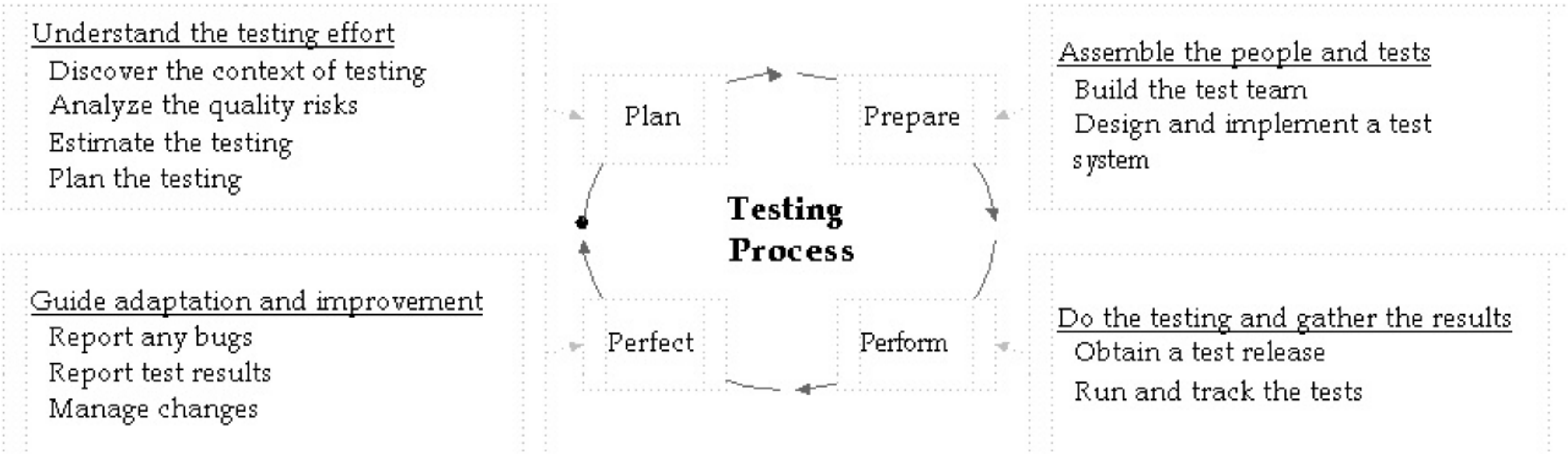
1 Fundamentals of Testing

1. Why is Testing Necessary?
2. What is Testing?
3. Seven Principles of Testing
4. Fundamental Test Processes
5. The Psychology of Testing
6. Code of Ethics

1.4 Fundamental Test Process

1. Critical Testing Processes
2. ISTQB Fundamental Test Process
3. Testing is far beyond test execution…

Test Processes: Critical Testing Processes



Understand the testing effort

- Discover the context of testing
- Analyze the quality risks
- Estimate the testing
- Plan the testing

Assemble the people and tests

- Build the test team
- Design and implement a test system

Do the testing and gather the results

- Obtain a test release
- Run and track the tests

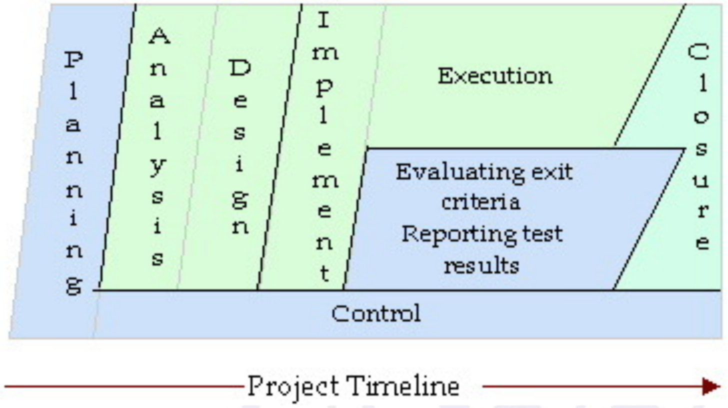
Guide adaptation and improvement

- Report any bugs
- Report test results
- Manage changes

The test engineer's role focuses on some of these activities, not all, depending on how roles are defined. However, the effective and efficient test engineer must understand how the test process works and how it fits into the overall project from a big picture perspective.

ISTQB Fundamental Test Process

- Test process in terms of the following steps
  - Planning and control
  - Analysis and design
  - Implementation and execution
  - Evaluating test exit criteria and reporting
  - Test closure activities
- These steps may overlap, be concurrent, and/or iterate



Planning and Control

Planning

- Determine test scope, risks, objectives, strategies
- Determine required test resources
- Implement the test strategies
- Schedule test analysis and design
- Schedule implementation, execution and evaluation of tests
- Determine the test exit criteria

Control

- Measure and analyze results
- Monitor and document progress, coverage and test exit criteria
- Initiate corrective actions
- Make decisions

Many planning and control activities involve obtaining project team and project management agreement, support, and consensus.

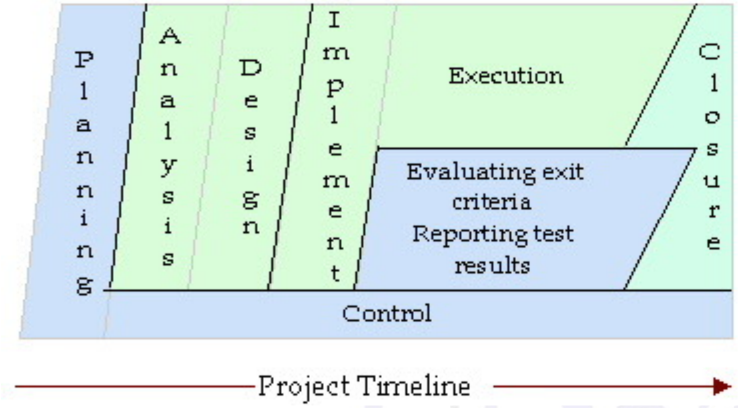
Analysis and Design

Analysis

- Review the test basis (e.g., requirements or design specifications, network/ system architecture, quality risks)
- Identify and prioritize test conditions, test requirements, or test objectives and required test data based on analysis of test items (e.g., their behavior, specification, and structure)
- Evaluate testability of the requirements and system

Design

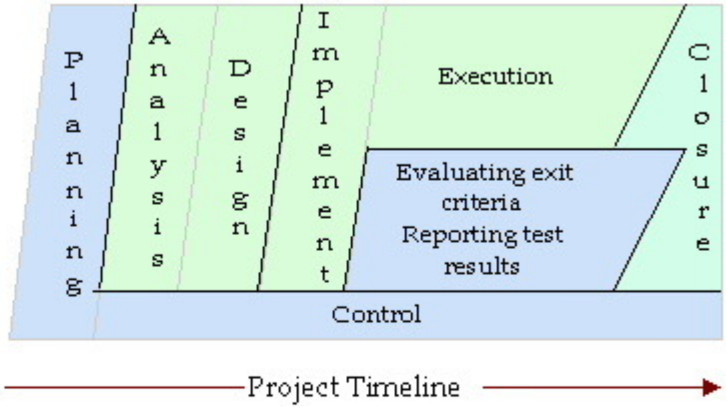
- Design and prioritize combinations of test data, actions, and expected results
- Identify the test data needed for test conditions and cases
- Design the test environment
- Identify infrastructure, tools



Implementation and Execution

Implementation

- Develop, implement, and prioritize test cases, create data, write procedures
- Create test harnesses, scripts
- Organize test suites and sequences of test procedures



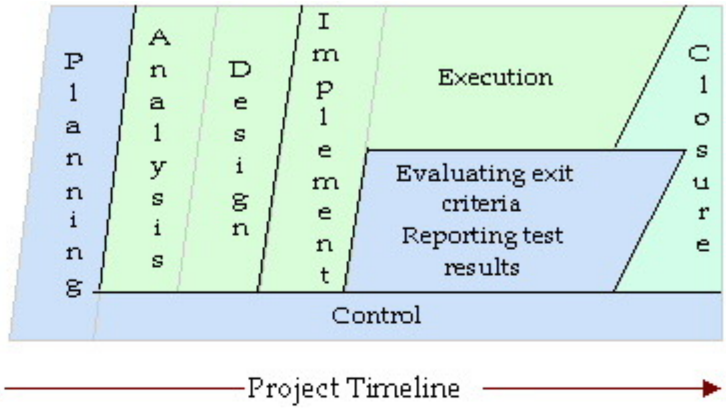
Execution

- Execute test cases (manual or automated)
- Log test results, and the versions of the software under test, test tools and the testware
- Compare actual and expected results
- Report and analyze incidents
- Repeat corrected and/or updated tests
- Run confirmation and/or regression tests

Exit Criteria, Reporting, and Closure

Exit and reporting

- Check test logs against the exit criteria in the test plan
- Assess if more tests are needed or if the exit criteria specified should be changed
- Write a test summary report for stakeholders



Closure

- Confirm test deliverables, final resolution or deferral of bug reports, and the acceptance of the system
- Finalize and archive testware, test environment and test infrastructure
- Deliver testware to the maintenance organization
- Perform a retrospective to capture improvements for future releases, projects, and test processes

1.4 Test Process Summary

Critical Testing Processes

- Plan
- Prepare
- Perform
- Perfect

ISTQB Testing Processes

1. Planning and control
2. Analysis and design
3. Implementation and execution
4. Evaluating test exit criteria and reporting
5. Test closure activities

Exercise1.4.1: Test Steps and Tasks Performed

- Think back on a recent project.
- Note which of the steps and tasks of the ISTQB test process were carried out.
- Note which of the steps and tasks of the ISTQB test process were not carried out.
- Note whether some steps overlapped or were completely parallel.
- Discuss.

Exercise1.3.1: Test Principles Observed (and Not)

- Think back on a recent project or an assignment.
- Note which of the principles that you can recall were observed. Illustrate them by examples.
- How about principles that you can recall were not observed (i.e., violated)?
- Discuss.

1 Fundamentals of Testing

1. Why is Testing Necessary?
2. What is Testing?
3. Seven Principles of Testing
4. Fundamental Test Processes
5. The Psychology of Testing
6. Code of Ethics

1.5 Phycology of Testing

1. Psychological factors for testing success
2. Defining Tester Skills

Psychological Factors 1:
Balancing Certainty & Progress

- Thorough research
  - indisputable bug reports,
  - unquestionably correct test cases, etc.

- Test projects adapt to schedule pressures or fail

Testing is more about *engineering*

- Not for exploring scientific certainty
- Engineering is making useful objects for customers

Psychological Factors 2:
Avoiding Test Result Misinterpretation

- Overestimated
  - report correct behavior as a bug
  - assign excessively high severity or priority
  - otherwise overstate bug significance

- Underestimated
  - fail to detect or report incorrect behavior
  - assign excessively low severity or priority
  - otherwise understate bug significance

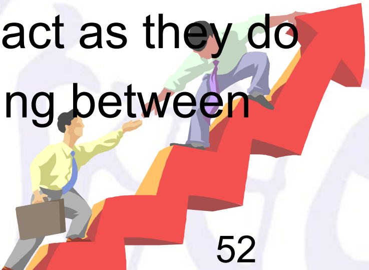
- *Some tips to avoid these errors*

- *1. Have testers **take breaks** so they don't miss important events*
- *2. **Automate** where practical*
- *3. **Define expected results** as clearly as possible*
- *4. Assign **the right testers** to each test execution task*
- *5. **Use peer reviews** for test execution and bug reports*

Psychological Factors 3:
Being Professional Pessimism

- Testing is often considered as a deconstructive activities
- How to deliver unwanted news of defect?
- Defect or failures are needed to be respectful communicated in a constructive way !

- Challenge:
  - to be positive, pleasant, and the bearer of bad news, all at once
- Tips:
  - Remind the common goal of better quality system
  - Focus on the product fact rather than persons in a neutral
    - Objective and factual incident report; Review findings together
  - Try to understand other people's feel and why they react as they do
  - Confirm understanding between each other



Psychological Factors 4: **Balancing Curiosity**

- Balance need for thoroughness in any one area with need to cover many areas in a short time

- Challenge:
- Focused on where the bugs are and isolated them

- Tips:
- Always analyze the impact of the bugs
- Keep balancing/updating the priority of the bugs
- Focus on the non-trivial bugs
- Stay focused on the goals of the test project

Exercise1.5.1: Psychology in Action

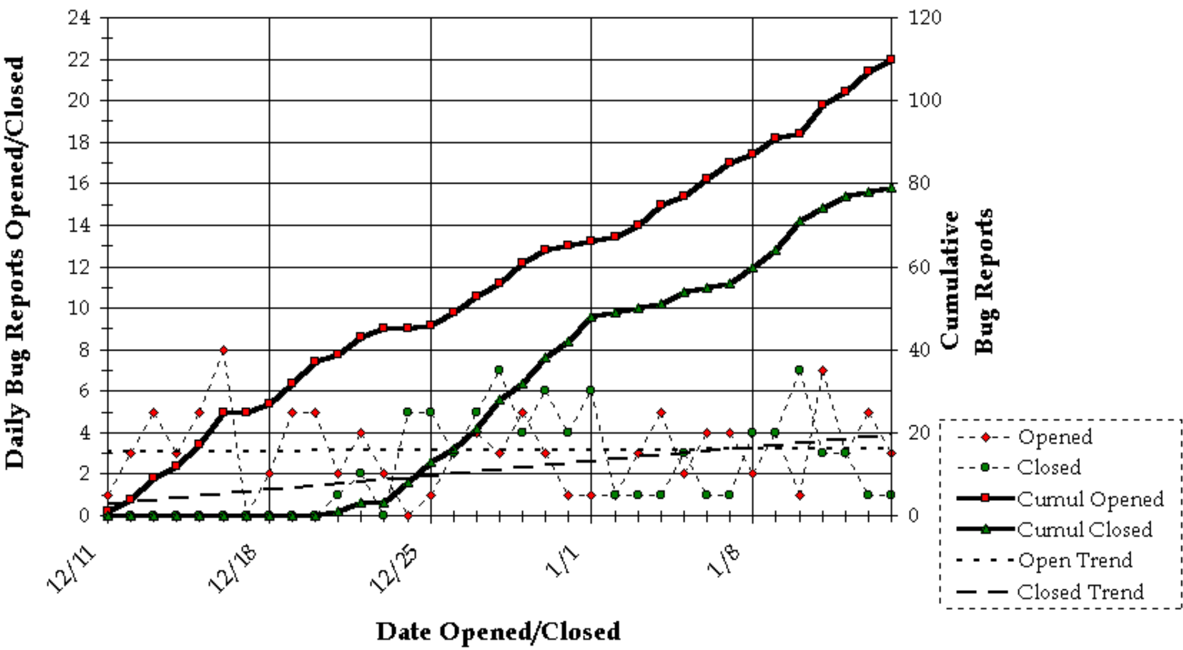
- Present one of the recent bug report or bug summary report to your team in the class.
- E.g., “Too Many Bugs,

Not Enough Time”

Explain the possible impact, action and implications in text.

Do you need more data to support your presentation?

SpeedyWriter System Test
Bug Reports Opened and Closed



1.5 Psychological Factors Summary

1. **Balancing Certainty & Progress**
2. **Avoiding Test Result Misinterpretation**
3. **Being Professional Pessimism**
4. **Balancing Curiosity**

1.6 Code of Ethics

- Public Interest *– always act consistently with the public interest*
- Client and Employer *– in their best interests*
- Product *– meet the highest professional standard possible*
- Judgment *– independent in professional judgment*
- Management *– promote an ethical approach*
- Profession *– with the public interest*
- Colleagues *– be fair and supportive*
- Self *- Life long learning*

Ethics Examples

- In January 2019, Google paid a €50 million fine to French authorities for failing to account for the collection and use of personal data on targeted ad placements. And even earlier, a Portuguese hospital paid €400,000 for its poor management of medical records. The hospital had created 1,000 doctor-level administrative accounts for convenience.

- And that's not all; the GDPR online enforcement tracking tool captures all the violations online, including an under-review, €204 million fine against British Airways for leaking the payment information of 500,000 passengers.

- January 2020 California passes the Consumer Privacy Act (**CCPA**), similar to the EU bill, the CCPA will regulate all data business practices that have business with California residents in the United States.

- November 1, 2021 China officially implements the Personal Information Protection Act. (2021年11日1日我国正式实施《个人信息保护法》)

Summary of Chapter 1

1. Why is Testing Necessary?
2. What is Testing?
3. Principles of Testing - 7 principles(P25)
4. Fundamental Test Processes – 5 processes (P45)
5. The Psychology of Testing - 4 factors (P50-53)
6. Code of Ethics

上海市嘉定区曹安公路4800号，同济大学嘉定校区软件学院