

Software Reliability and Testing

Summer semester 2013/2014
Fundamentals of Testing

Barbara Russo

27 Feb 2014

Table of Contents

Outline

Testing and
reliability

Testing

① Testing and reliability

② Testing

Testing and reliability



According to IEEE definition:

Software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item

In practice:

Testing is the process of executing a program with the intent of finding an error

According to IEEE definition:

Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time

Exhaustive testing is not feasible:
if a failure is detected then the software is a failure software,
but this does not imply that if no failure has been detected
the software is correct

Measuring and evaluating a software process may lead to the only notion of “good” process we can define:
a “good” process is correlated with the probability that no failures occur in a given interval of time → Software reliability

- Cost of software failures surpasses the cost of testing
 - Expenses for debugging and recall of product
 - Loss of customers? faith in company
 - Effective test building
 - Uncovering as many defects as possible with minimum amount of time and effort
- Is all of this feasible?

Basic Questions

- When do testing start? When is it complete?

Basic Questions

- When do testing start? When is it complete?
- What particular techniques should be applied during software development to get acceptable quality at acceptable cost?

Basic Questions

- When do testing start? When is it complete?
- What particular techniques should be applied during software development to get acceptable quality at acceptable cost?
- How can we assess the readiness of a product for release?

- When do testing start? When is it complete?
- What particular techniques should be applied during software development to get acceptable quality at acceptable cost?
- How can we assess the readiness of a product for release?
- How can we control the quality of a product to release?

- When do testing start? When is it complete?
- What particular techniques should be applied during software development to get acceptable quality at acceptable cost?
- How can we assess the readiness of a product for release?
- How can we control the quality of a product to release?
- How can the development process itself be improved over the course of current and future projects to improve products and make verification more cost-effective?

When do testing start? When is it complete?

- Testing activities are spread all over the development process
- There are several techniques at different stages of the process
- Different techniques involve different stakeholders

When do testing start? When is it complete?

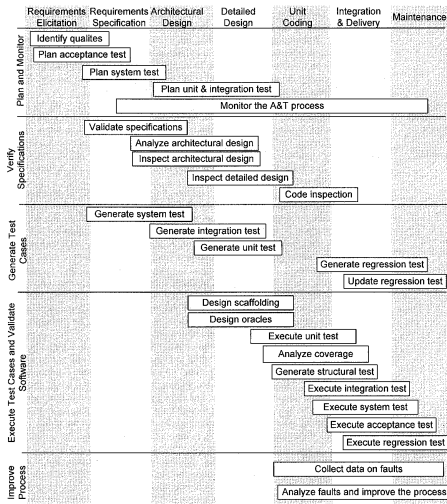


Figure : Main analysis and testing activities through the software life cycle. Source: Pezzè and Young (reference book)

What techniques should be applied during software development?

- Feasibility study to understand and select appropriate techniques
- Verification to lead development according to requirements
- Validation to check product against users' expectations

What techniques should be applied during software development?

- Techniques depend on quality, cost, scheduling, resources
- Not “one technique”
 - Effectiveness for different classes of fault to capture.
 - Applicability at different stages of the project (early stage artefacts might not be tested with automated tools)
 - Differences in purpose. Techniques can have different goals (to understand the coverage or to detect faults)
 - Trade-off cost and assurance, e..g, focus on key few properties

What techniques should be applied during software development?

- Feasibility study includes
 - Tentative architectural design to modularise work and identify properties that can be verified in different subsystems
 - Draft plan breaking projects into incremental deliveries preliminary decisions about test and analysis techniques

How can we assess the readiness of a product for release?

Dependability properties:

- Availability
- Mean Time Between Failures
- Reliability

How can we assess the readiness of a product for release?

Examples of dependability measures:

- Availability: daily down time
- Mean Time Between Failures (MTBF): days between two consecutive failures
- Reliability: number of functionalities that correctly operate

How can we assess the readiness of a product for release?

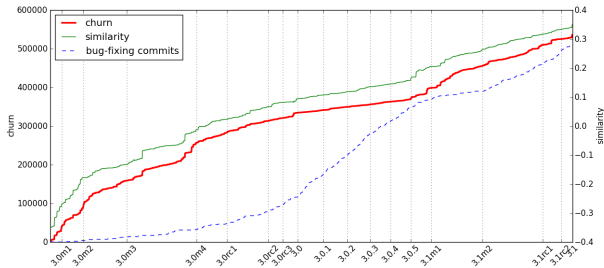
How:

- alpha testing: performed by users in a controlled environment. Capture operational profiles decide by the organisation.
- beta testing: performed by users in a their own environment. Capture different operational profiles

How can we control the quality of successive releases?

- Major revisions also called “point releases.” The full quality process is repeated including beta testing and regression testing.
- Small revisions also called “patch releases.” Revision related to incorporate solution to fault fixing. Fast testing often automated is foreseen. Subset of regression testing.
- in Open Source Software we might have “milestones,” “release candidates,” and “major releases”

How can we control the quality of successive releases?



How can the development process itself be improved over the course of current and future projects?

- Data collection across projects for modelling and prediction, e.g., faults and their severity or priority
- Personal Software Process (Humphrey)