Documenting Analysis and Test
Learning objectives

• Be able to explain the purposes and importance of documentation
• Identify some key quality documents and their relations
• Be able to structure and provide content for key quality documents
• Appreciate needs and opportunities for automatically generating and managing documentation
Why Produce Quality Documentation?

• Monitor and assess the process
  • For internal use (process visibility)
  • For external authorities (certification, auditing)

• Improve the process
  • Maintain a body of knowledge reused across projects
  • Summarize and present data for process improvement

• Increase reusability of test suites and other artifacts within and across projects
Major categories of documents

• Planning documents
  • describe the organization of the quality process
  • include organization strategies and project plans

• Specification documents
  • describe test suites and test cases
    (as well as artifacts for other quality tasks)
  • test design specifications, test case specification, checklists, analysis procedure specifications

• Reporting documents
  • Details and summary of analysis and test results
Metadata

• Documents should include *metadata* to facilitate management
  • **Approval**: persons responsible for the document
  • **History**: of the document
  • **Table of Contents**
  • **Summary**: relevance and possible uses of the document
  • **Goals**: purpose of the document—Who should read it, and why?
  • **Required documents and references**: reference to documents and artifacts needed for understanding and exploiting this document
  • **Glossary**: technical terms used in the document
Naming conventions

• Naming conventions help people identify documents quickly
• A typical standard for document names include keywords indicating
  • general scope of the document (project and part)
  • kind of document (for example, test plan)
  • specific document identity
  • version
Sample naming standard

Project or product (e.g., \( W \) for “web presence”)

Sub-project (e.g., “Business logic”)

Item type

Identifier

Version

\[ W \quad B \quad XX \quad – \quad YY \quad .ZZ \]

example:

\[ W \quad B \quad 12 \quad – \quad 22 \quad .04 \]

Might specify version 4 of document 12-22 (quality monitoring procedures for third-party software components) of web presence project, business logic subsystem.
Analysis and test strategy

• Strategy document describes quality guidelines for sets of projects (usually for an entire company or organization)
• Varies among organizations
• Few key elements: common quality requirements across products
• May depend on business conditions - examples
  • safety-critical software producer may need to satisfy minimum dependability requirements defined by a certification authority
  • embedded software department may need to ensure portability across product lines
• Sets out requirements on other quality documents
Analysis and Test Plan

• Standardized structure  see next slide
• Overall quality plan comprises several individual plans
  • Each individual plan indicates the items to be verified through analysis or testing
  • Example: documents to be inspected, code to be analyzed or tested, ...
• May refer to the whole system or part of it
  • Example: subsystem or a set of units
• May not address all aspects of quality activities
  • Should indicate features to be verified and excluded
    • Example: for a GUI– might deal only with functional properties and not with usability (if a distinct team handles usability testing)
  • Indication of excluded features is important
    • omitted testing is a major cause of failure in large projects
Test Design Specification Documents

• Same purpose as other software design documentation:
  • Guiding further development
  • Preparing for maintenance

• Test design specification documents:
  • describe complete test suites
  • may be divided into
    • unit, integration, system, acceptance suites (organize by granularity)
    • functional, structural, performance suites (organized by objectives)
    • ...
  • include all the information needed for
    • initial selection of test cases
    • maintenance of the test suite over time
  • identify features to be verified (cross-reference to specification or design document)
  • include description of testing procedure and pass/fail criteria (references to scaffolding and oracles)
  • includes (logically) a list of test cases
Test case specification document

• Complete test design for individual test case
• Defines
  • test inputs
  • required environmental conditions
  • procedures for test execution
  • expected outputs
• Indicates
  • item to be tested (reference to design document)
• Describes dependence on execution of other test cases
• Is labeled with a unique identifier
Test and Analysis Reports

• Report test and analysis results

• Serve
  • Developers
    • identify open faults
    • schedule fixes and revisions
  • Test designers
    • assess and refine their approach see chapter 20

• Prioritized list of open faults: the core of the fault handling and repair procedure

• Failure reports must be
  • consolidated and categorized to manage repair effort systematically
  • prioritized to properly allocate effort and handle all faults
Summary reports and detailed logs

• Summary reports track progress and status
  • may be simple confirmation that build-and-test cycle ran successfully
  • may provide information to guide attention to trouble spots

• Include summary tables with
  • executed test suites
  • number of failures
  • breakdown of failures into
    • repeated from prior test execution,
    • new failures
    • test cases that previously failed but now execute correctly

• May be prescribed by a certifying authority
Isn’t this a lot of work?

• Yes, but
  • Everything produced by hand is actually used
    • Always know the purpose of a document. Never expend effort on documents that are not used.
  • Parts can be automated
    • Humans make and explain decisions. Let machines do the rest.

• Designing effective quality documentation
  • Work backward from use, to output, to inputs
    • and consider characteristics of organization and project
  • Capture decisions and rationale at cheapest, easiest point and avoid repetition
Summary

• Mature software processes include documentation standards for all activities, including test and analysis

• Documentation can be inspected to
  • verify progress against schedule and quality goals
  • identify problems

• Documentation supports process visibility, monitoring, and standardization