Common System and Software Testing Pitfalls

ER-2014 Conference
Atlanta, Georgia

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Donald G. Firesmith, Principle Engineer
27 October 2014

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Topics

Extremely brief introduction to *Common System and Software Testing Pitfalls*

Current status of Testing Pitfalls Repository since book was published by Addison-Wesley in January 2014.

Future Work

Terminology

Testing Pitfall

- A *human mistake* that unnecessarily and unexpectedly causes testing to be:
  - Less effective at uncovering defects
  - Less efficient in terms of time and effort expended
  - More frustrating to perform
- A bad decision, an incorrect mindset, a wrong action, or failure to act
- A *failure* to adequately:
  - Meet a testing challenge
  - Address a testing problem
- A way to screw up testing

Common Testing Pitfall

- Observed numerous times on different projects
- Having sufficient frequency (and consequences) to be a significant risk

Testing Pitfalls Taxonomy and Ontology

Anti-Pattern Language of how-not-to do testing

Past and Present

Common System and Software Testing Pitfalls (Addison-Wesley, 2014) (Note: 35% conference discount):

- ***92*** pitfalls classified into ***14*** categories
- Technically reviewed by 47 testing international SMEs

Current taxonomy/repository with new pitfalls and pitfall categories:

- ***139*** pitfalls classified into ***20*** categories
- http://sites.google.com/a/firesmith.net/donald-firesmith/home/common-testing-pitfalls [Work in progress - new content is draft and may be incomplete]

Goals and Potential Uses

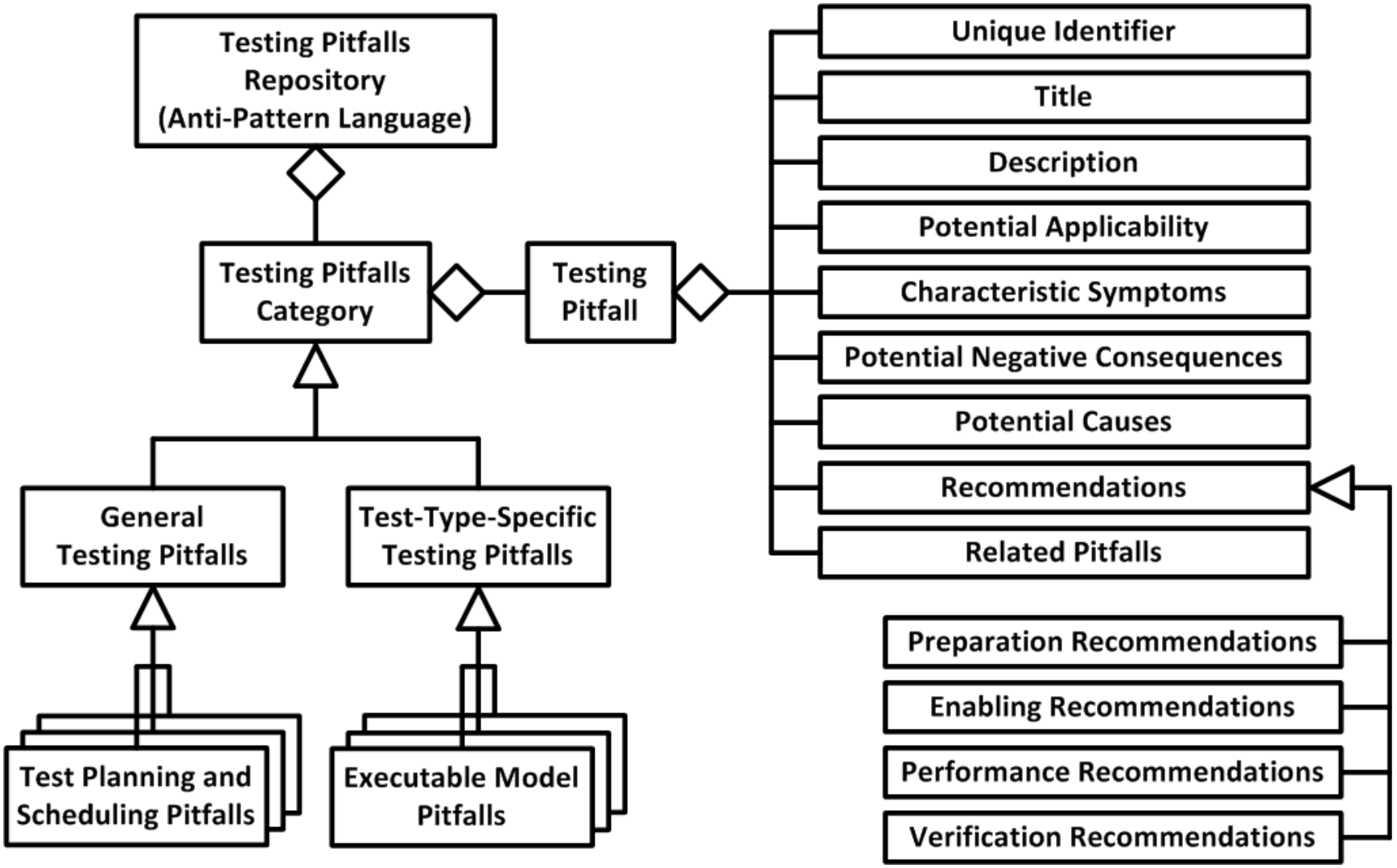
Goals:

- To become the de facto industry-standard taxonomy of testing pitfalls
- To reduce the incidence of testing pitfalls and thereby improve testing effectiveness and efficiency
- To improve the quality of the objects under test (OUTs)

Potential Uses:

- Training materials for testers and testing stakeholders
- Standard terminology regarding commonly occurring testing pitfalls
- Checklists for use when:
  - Producing test strategies/plans and related documentations
  - Evaluating contractor proposals
  - Evaluating test strategies/plans and related documentation (quality control)
  - Evaluating as-performed test process (quality assurance)
  - Identifying test-related risks and their mitigation approaches
- Categorization of pitfalls for test metrics collection, analysis, and reporting

Testing Pitfall Taxonomy and Ontology



Example – Testing and Engineering Processes Not Integrated (GEN-PRO-7)

Description The testing process is not adequately integrated into the overall system engineering process.

Potential Applicability This pitfall is potentially applicable anytime that engineering and testing processes both exist.

Characteristic Symptoms

- There is little or no discussion of testing in the system engineering documentation: System Engineering Management Plan (SEMP), Software Development Plan (SDP), Work Breakdown Structure (WBS), Project Master Schedule (PMS), or System Development Cycle (SDC).
- All or most of the testing is done as a completely independent activity performed by staff members who are not part of the project engineering team.
- Testing is treated as a separate specialty engineering activity with only limited interfaces with the primary engineering activities.
- Test scheduling is independent of the scheduling of other development activities.
- Testers are not included in the requirements teams, architecture teams, or any cross-functional engineering teams.

Example – Testing and Engineering Processes Not Integrated (GEN-PRO-7)

Potential Negative Consequences

- There is inadequate communication between testers and other system or software engineers (for example, requirements engineers, architects, designers, and implementers).
- Few nontesters understand the scope, complexity, and importance of testing.
- Testers do not understand the work being performed by other engineers.
- Testers can produce test cases and automated testing scripts before the requirements, architecture, and design has stabilized, thereby forcing the testers to modify their test cases and test scripts as the system or software changes and incorrect hidden assumptions are uncovered.
- Testing is less effective and takes longer than necessary.

Example – Testing and Engineering Processes Not Integrated (GEN-PRO-7)

Potential Causes

- Testers were not involved in determining and documenting the overall engineering process.
- The people determining and documenting the overall engineering process did not have significant testing expertise, training, or experience.
- The testing schedule has not been integrated into the overall project schedule.
- Testing was outsourced.

Example – Testing and Engineering Processes Not Integrated (GEN-PRO-7)

Recommendations

- **Prepare:**
  - Include testers in the initial staffing of the project.
- **Enable:**
  - Provide a top-level briefing or training in testing to the chief system engineer, system architect, and process engineer.
- **Perform:**
  - Subject-matter experts and project testers collaborate closely with the project chief engineer or technical lead and process engineer when they develop the engineering process descriptions and associated process documents.
  - Provide high-level overviews of testing in the SEMP(s) and SDP(s).
  - Document how testing is integrated into the system development or life cycle, regardless of whether it is traditional waterfall, evolutionary (iterative, incremental, and parallel), or anything in between.
  - For example, document handover points in the development cycle when testing input and output work products are delivered from one project organization or group to another.
  - Incorporate testing into the Project Master Schedule.
  - Incorporate testing into the project's Work Breakdown Structure (WBS).
- **Verify:**
  - Determine whether testers were involved in planning the project's system or software development process.
  - Determine whether testing is incorporated into the project's System engineering process, System development cycle, System Engineering Master Plan and System Development Plan, Work Breakdown Structure, Master Schedule

Example – Testing and Engineering Processes Not Integrated (GEN-PRO-7)

Related Pitfalls

- Testing at the End (GEN-TPS-6) If the testing and engineering processes are not properly integrated, then testing is more likely to be delayed to the end of development.
- Independent Test Schedule (GEN-TPS-7) If the testing and engineering processes are not integrated, then the test schedule will be independent of and therefore probably incompatible with the overall project master schedule.
- Testers Responsible for All Testing (GEN-STF-4) If the testing and engineering processes are not properly integrated, then the developers are more likely to believe that the testers are responsible for all testing.
- Adversarial Relationship (GEN-STF-9) If the testing and engineering processes are not integrated, then the developers and the testers are more likely to develop and adversarial rather than cooperative relationship.
- Testing as a Phase (GEN-PRO-18) If the testing and engineering processes are not properly integrated, then testing is more likely to be viewed as a phase that is separate from the rest of development.
- Testers Not Involved Early (GEN-PRO-19) If the testing and engineering processes are not properly integrated, then testers are less likely to be involved early in the development process (such as during initial planning, requirements engineering, and architecture engineering.
- Testing in Quality (GEN-PRO-23) If the testing and engineering processes are not properly integrated, then the developers will be more likely to believe that quality is the testers responsibility and that quality can be testing into the system or software.
- Developers Ignore Testability (GEN-PRO-24) If the testing and engineering processes are not properly integrated, then the developers are more likely to ignore testability.
- Inadequate Communication Concerning Testing (GEN-COM-5) If the testing and engineering processes are not properly integrated, then it is more likely that there will be inadequate communication between the testers and the rest of the engineering staff.

Categories of Testing Pitfalls – General

1 Test Planning and Scheduling

2 Stakeholder Involvement and Commitment

3 Management

4 Staffing

5 Testing Process

6 Pitfall-Related [new pitfall category since book]

7 Test Tools and Environments

8 Automated Testing [new pitfall category since book]

9 Test Communication

10 Testing-as-a-Service (TaaS) [new pitfall category since book]

11 Requirements

Categories of Testing Pitfalls – Test-Type-Specific Pitfalls

1 Executable Model Testing [new pitfall category]

2 Unit Testing

3 Integration Testing

4 Specialty Engineering Testing

5 System Testing

6 User Testing [new pitfall category]

7 Acceptance Testing [new pitfall category]

8 System of Systems (SoS) Testing

9 Regression Testing

Remaining Limitation and Questions

Current Taxonomy is Experience Based:

- Based on experience testing and assessing testing programs (author, SEI ITAs, technical reviewers)
- Not the result of documentation study or formal academic research

Remaining Questions:

- Which pitfalls occur most often? With what frequency?
- Which pitfalls cause the most harm?
- Which pitfalls have the highest risk (expected harm = harm frequency x harm)?
- What factors (e.g., system/software size and complexity, application domain, process) influence frequency, harm, and risk?

Future Work

Second Edition of Book or Supplement to First Book

Extensive Technical Review:

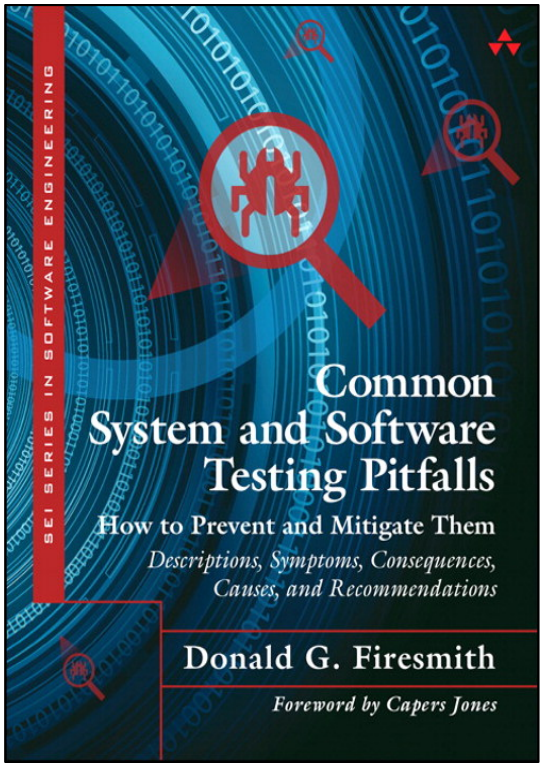
- New testing pitfall categories
- New and modified testing pitfalls

Proper Industry Survey:

- How likely are the different testing pitfalls? What are the 10 most common?
- What pitfalls have the worst consequences? What are the 10 worst pitfalls?
- What pitfalls have the highest risk? What are the 10 highest risk pitfalls?
- Do the answers to these questions vary by:
  - System (size, complexity, criticality, application domain, software only vs. HW/SW/people/documentation/facilities/procedures…, system vs. SoS vs. PL)?
  - Project (type, formality, lifecycle scope, schedule, funding, commercial vs. government/military,…)
  - Organization (number, size, type, governance, management/engineering culture,…)

Wiki

Save 35%* at informit.com



Discount code:

FIRESMITH550

- informit.com - search on Firesmith
- Available as book & eBook
- FREE shipping in the U.S.

*Offer expires Dec 31, 2014

http://sites.google.com/a/firesmith.net/donald-firesmith/home/common-testing-pitfalls

Addison Wesley

PEARSON

Contact Information Slide Format

Donald G. Firesmith
Principal Engineer
Software Solutions Division
Telephone: +1 412-268-6874
Email: dgf@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257