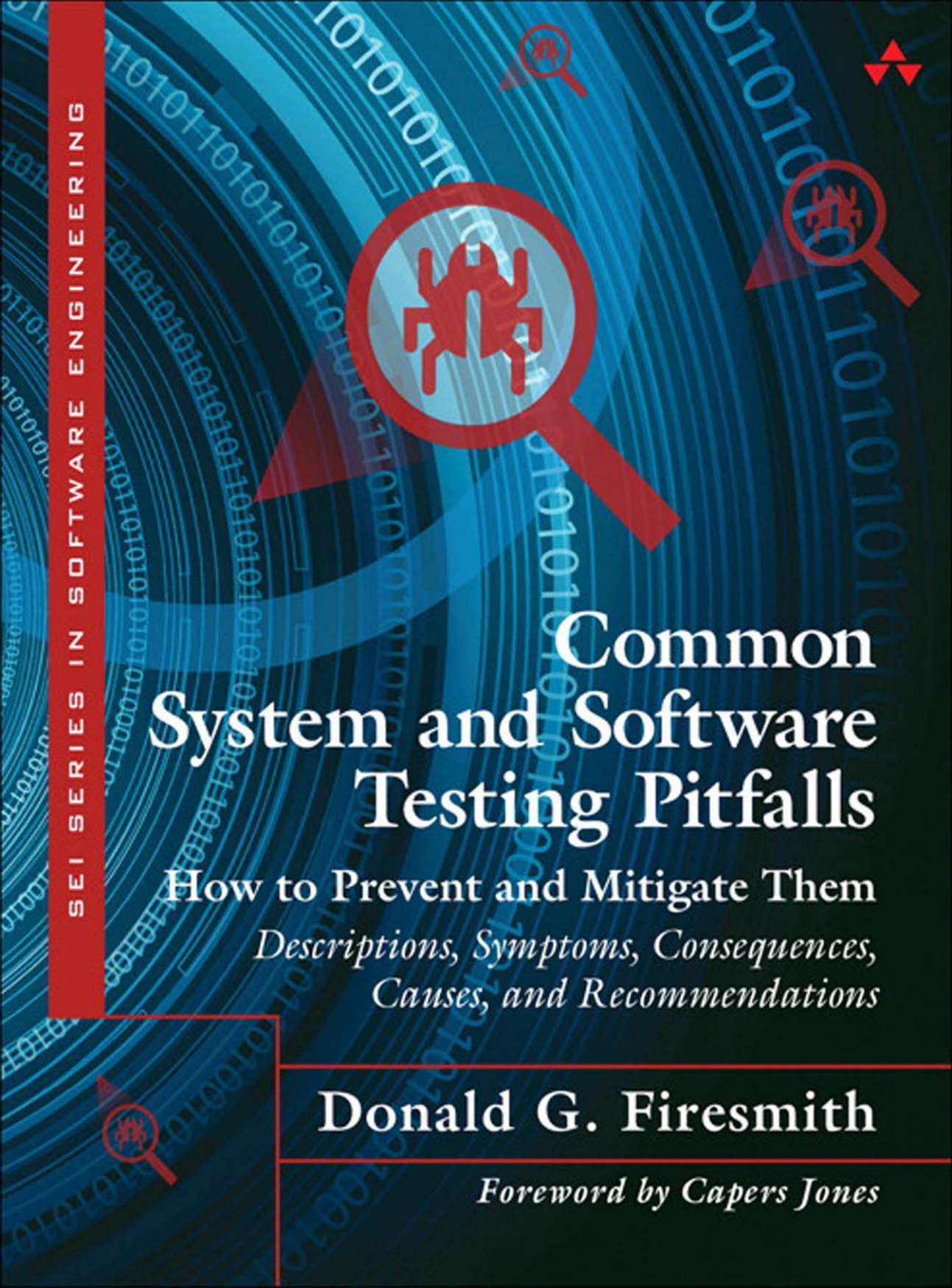
SEI SERIES IN SOFTWARE ENGINEERING
Common System and Software Testing Pitfalls
How to Prevent and Mitigate Them
Descriptions, Symptoms, Consequences, Causes, and Recommendations
Donald G. Firesmith
Foreword by Capers Jones

Praise for *Common System and Software Testing Pitfalls*

"Firesmith's collection of actionable practices for real-world, non-trivial testing and the processes in which they're applied is comprehensive and uniquely valuable. Nothing published about software testing in recent years provides anything like it."

—Robert V. Binder, robertvbinder.com

"Don's compilation of real-world testing problems, symptoms, and solutions is the most comprehensive available. You can use it early in your project to prevent these problems. Or you can use it at the end, as a ready list of costly lessons you could have avoided, had you used it early on. I'm afraid this book's publication will undermine a lot of excuses for repeating these mistakes."

—Vince Alcalde, National Australia Bank

"Excellent, Excellent, Excellent! This book should be mandatory reading for anyone involved in product development. Donald's book addresses the pitfalls that need to be understood and allowed for in all product development verification and validation planning. While the focus of the book is on software projects, most of the pitfalls are equally applicable to any size project that involves both hardware and software."

—Louis S. Wheatcraft, Requirements Experts Inc.

"The potential impact of this book cannot be overstressed. Software systems that are not adequately tested do not adequately evolve. I highly recommend this book as a must-read for people directly involved in the development and management of software-intensive systems."

—Dr. Kenneth E. Nidiffer, Director of Strategic Plans for Government Programs, Software Engineering Institute, Carnegie Mellon University

"*Common System and Software Testing Pitfalls* identifies realistic testing pitfalls. More importantly, it also identifies solutions for avoiding them on your next project. Every manager should read this book and follow the recommendations."

—Barry Stanly, Enterprise Technology Alliance

Developers Responsible for All Testing (GEN-STF-4)

Description There is no separate full-time tester role.

Potential Applicability This pitfall is always potentially applicable. In fact, it is becoming increasingly common due to the popularity of agile development that largely (and incorrectly) implicitly depends on all developers on the team having adequate testing expertise.

Characteristic Symptoms

- There is no separate tester role.
- The members of each development team share responsibility for testing what the team has developed:
  - Not just individual programmers performing unit testing of their own code, but also integration testing, system testing, specialty engineering testing, and so on.
- Individual development team members are responsible for testing:
  - The software that they personally developed
  - The software developed by other members of their team (that is, peer testing)

Potential Negative Consequences

- Testing is incomplete and inefficient due to lack of expertise and experience, thereby permitting excessive numbers of residual defects to remain in the delivered system.
- The developers performing testing are neither objective nor unbiased because they have a major conflict of interest: As designers and implementers, they must ensure that the system works properly; as testers, they must determine why the system does not work correctly.
- When the developers performing testing make false assumptions concerning how the system under test is supposed to behave, their false assumptions:
  - Will not be spotted by the testers
  - Will influence the test cases such that the test cases do not spot the false assumptions
- Because they don't report independently, developers performing testing are intimidated into withholding information about negative test results in order to meet schedule deadlines.

Potential Causes

- According to Capers Jones, individual programmers typically find less than one-half of the defects when they test their own software [Jones 2012].
- Testing stakeholders (especially acquisition and development management) were unaware of the levels of expertise and experience needed to be an effective professional tester.
- The developers performing testing did not have adequate training, expertise, and experience to perform testing, especially higher-level testing such as integration, system, and acceptance testing.
- An agile development process and philosophy were used, whereby everyone performs the same tasks. Therefore, some testing is being performed by non-testers who do not have adequate training, expertise, or experience in testing.
- Management did not see the value of or understand the need for professional testers (that is, people who specialize in and are responsible for testing).
- Management was not able to hire a sufficient number of testers with the appropriate training, expertise, and experience.

Recommendations

- **Prepare:**
  - Ensure that the test planning documentation includes the use of full-time, professional testers.
- **Enable:**
  - Clarify to testing stakeholders (especially project management) the value of and need for professional, full-time testers.
  - Staff the project with full-time, professional testers.
  - Give the testers the resources they need to properly perform testing.
- **Perform:**
  - Ensure that the testers perform high-level (for example, integration and system) testing.
- **Verify:**
  - Determine whether the test planning documentation specifies using full-time, professional testers.
  - Determine whether full-time, professional testers are performing the more sophisticated, higher-level testing while developers do the lowest-level testing.

Related Pitfalls Inappropriate External Pressures (GEN-MGMT-2), Inadequate Testing Expertise (GEN-STF-3), Testers Responsible for All Testing (GEN-STF-5)

Testers Responsible for All Testing (GEN-STF-5)

Description Testers are responsible for all of the testing during system development.

Potential Applicability This pitfall is potentially applicable anytime that tester is a role or job title of one or more persons or teams during development.

Characteristic Symptoms

- Developers are not performing unit testing (either of their own software or that of their peers).
- Developers have essentially no testing training, expertise, or experience.

Potential Negative Consequences

- Testers must develop a deep understanding of the internals of everyone's units and low-level components.
- There needs to be a greater number of testers than is typical.
- Testers rarely have sufficient resources to perform integration and higher-level testing.
- Testing becomes a significant bottleneck that lowers developers' productivity.

Potential Causes

- There was a misunderstanding that testers were the only ones responsible for quality.
- There was a misunderstanding about the need for independent testing of units and small components.
- The developers did not know enough about how to perform unit and low-level integration testing.
- There was not sufficient time in the schedule or funding to provide the developers with any training in testing.
- Because the testers had to perform the lowest-level testing, there were too few testers to perform all of the other testing within schedule and funding constraints.

Recommendations

- Prepare:
  - Ensure that the test planning documentation includes the testing responsibilities of developers who are not professional testers.

- **Enable:**
  - Provide training in low-level testing to the developers.
  - Hire developers who have sufficient training, expertise, and experience performing low-level testing.
  - Provide developers with sufficient resources (time, funds, and tools) to perform unit and low-level integration testing.
- **Perform:**
  - Ensure that the developers perform unit testing and, potentially, integration testing.
- **Verify:**
  - Determine whether the test planning documentation includes the developers' responsibilities with regard to testing.
  - Determine whether developers are performing their assigned testing.

Related Pitfalls Developers Responsible for All Testing (GEN-STF-4)

3.3.5 Test Process Pitfalls

The following testing pitfalls are related to the processes and techniques being used to perform testing:

- Testing and Engineering Process Not Integrated (GEN-PRO-1)
- One-Size-Fits-All Testing (GEN-PRO-2)
- Inadequate Test Prioritization (GEN-PRO-3)
- Functionality Testing Overemphasized (GEN-PRO-4)
- Black-Box System Testing Overemphasized (GEN-PRO-5)
- Black-Box System Testing Underemphasized (GEN-PRO-6)
- Too Immature for Testing (GEN-PRO-7)
- Inadequate Evaluations of Test Assets (GEN-PRO-8)
- Inadequate Maintenance of Test Assets (GEN-PRO-9)
- Testing as a Phase (GEN-PRO-10)
- Testers Not Involved Early (GEN-PRO-11)
- Incomplete Testing (GEN-PRO-12)
- No Operational Testing (GEN-PRO-13)
- Inadequate Test Data (GEN-PRO-14)
- Test-Type Confusion (GEN-PRO-15)

Testing and Engineering Processes Not Integrated (GEN-PRO-1)

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.
- Testers are not included in the requirements teams, architecture teams, or any cross-functional engineering teams.

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.
- Testing is less effective and takes longer than necessary.

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.
- Testing was outsourced.

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

Related Pitfalls Inadequate Communication Concerning Testing (GEN-COM-5)

One-Size-Fits-All Testing (GEN-PRO-2)

Description All testing is performed the same way, to the same level of rigor, regardless of its criticality.

Potential Applicability This pitfall is always potentially applicable.