12 Testing Processes and Why They Matter

Bugs are expensive
- US economy loses $60B yearly due to bugs, $20B of which could be saved through better testing

Bugs drive away customers
- Handwriting recognition bugs in Newton cost Apple the personal digital assistant (PDA) market

Bugs sometimes kill people
- Therac-25 overdoses, Patriot clock overflow

Inadequate testing endangers the project
- Bad testing processes can increase the risk of project delay or cancellation by 25 to 300%

Why Care About Testing Processes?

Bugs are expensive
Bugs drive away customers
Bugs sometimes kill people
Inadequate testing endangers the project
What is a Critical Testing Process?

- Process: A sequence of actions, observations, and decisions
- Testing: Assessing the quality of a system
- A test process becomes critical when it is...
  - Repeated frequently → affects efficiency
  - Highly cooperative → affects team cohesion
  - Visible to peers and superiors → affects credibility
  - Linked to project success → affects effectiveness
- In other words, critical test processes directly and significantly affect the test team’s ability to find bugs and reduce risks through valuable information and services.

What Are the 12 Critical Testing Processes?

1. Testing
2. Establishing context
3. Quality risk analysis
4. Test estimation
5. Test planning
6. Test team development
7. Test system development
8. Test release management
9. Test execution
10. Bug reporting
11. Results reporting
12. Change management
The Testing Process

What is it?
- Planning: figuring out what testing to do
- Preparing: building the tests and test team
- Performing: installing the system under test and testing it
- Perfecting: reporting findings and guiding the process

Why does it matter?
- Reduces costs by finding important bugs
- Provides useful information about less important bugs
- Reduces risk by identifying what works and what doesn’t
- Gives management essential information

A Quick Case Study in Testing ROI
- So, can we agree conceptually that testing matters?
- “Don’t give me that touchy-feeling [stuff],” say the executives, “show me the money!”
- Can we quantify the return on the testing investment?
- Yes!

Testing and Constituent Processes

Understand the testing effort
- Discover the context of testing
- Analyze the quality risks
- Estimate the testing
- Plan for testing

Guide adaptation and improvements
- Report any bugs
- Report test results
- Manage changes

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

To optimize the overall testing process—to provide the best possible, most valuable, most timely testing information and services—we must optimize each constituent testing process.

Establishing Context

What is it?
- Studying documents
- Talking to participants
- Fitting in organization
  - Testers’ relationship to the project team
  - Project team testing
  - Stakeholders
- Fitting in operation
  - System lifecycle
  - Testing deliverables
  - Hand-offs

Why does it matter?
- Aligns testing within the project
- Aligns testing with the organization
- Clarifies expectations on all sides
- Establishes the groundwork for tailoring all other testing processes
Quality Risk Analysis

What is it?
- Discussing potential problems with stakeholders
- Prioritizing the levels of risk based on likelihood and impact of potential problem
- Applying FMEA, ISO 9126, or other risk analysis processes
- Documenting findings

Why does it matter?
- Identifies the key risks to system quality
- Aligns testing with the key risks to system quality
- Builds quality and test stakeholder consensus around what is to be tested (and how much) and what is not to be tested (and why)

Test Estimation

What is it?
- Identifying the testing tasks, resources, and dependencies through a work-breakdown-structure
- Drawing on test and project team wisdom
- Putting together a budget
- Selling the estimate to management

Why does it matter?
- Balances costs and time required for testing against project needs and risks
- Accurately, actionably forecasts the tasks and duration of testing
- Demonstrates the return on the testing investment
Test Planning

**What is it?**
- Thinking through the strategic and tactical details of testing
- Creating a document as a communication tool
- Reviewing and finalizing the document with the test and project team

**Why does it matter?**
- Builds consensus and commitment among test team and broader project team participants
- Creates a detailed map for all test participants
- Captures information for retrospectives and future projects

Test Team Development

**What is it?**
- Identifying the critical skills needed on the test team
- Hiring people who have those skills
- Growing the skills of the existing and new team members in line with the critical skills
- Continuously revising the list of critical skills

**Why does it matter?**
- The testing is only as good as the team that does it
- Matches test team skills to the critical test tasks
- Assures competence in the critical skills areas
- Continuously aligns team capabilities with organizational value of testing
Test System Development

What is it?
- Identifying specific tests to address critical quality risks
- Designing and implementing test cases, data, scripts, oracles, and environments
- Documenting tests and test coverage

Why does it matter?
- Ensures coverage of the critical risks to system quality
- Creates tests that reproduce the customers’ and users’ experiences of quality
- Balances resource and time requirements against criticality of risk

Test Release Management

What is it?
- Delivery of a known set of components for testing
- A hand-off from the project team to the test team
- Balancing need for quick feedback against need for testing progress

Why does it matter?
- If we ain’t got it, we can’t test it
- If it doesn’t work in the test environment, we can’t test it
- If each test release is not better than the one before, we’re not on a path for success
**Test Execution**

**What is it?**
- Prioritizing tests with risk
- Assigning specific tests to specific people
- Putting the system under test into interesting states and seeing what happens
- Gathering results
- Fine-tuning the test work that remains

**Why does it matter?**
- Generates information about bugs, what works, and what doesn’t (i.e., this is where the value of testing is created)
- Consumes significant resources
- Occurs at the end of the project and gates project completion

---

**Bug Reporting**

**What is it?**
- Describing a specific way in which the system doesn’t work properly
- Giving key information
  - To developers
  - To managers
  - To other testers
  - To tech support
- A hand-off from the test team to the project team

**Why does it matter?**
- Creates an opportunity to improve system (and save money)
- Delivers (part of) the value of testing to the project team
- Builds tester credibility with programmers
Results Reporting

What is it?
- Understanding project goals and the questions we need to answer about those goals
- Developing metrics, graphs, and charts that answer those questions
- Presenting information and findings of testing
- A hand-off from the test team to the project team

Why does it matter?
- Provides management with the information needed to guide the project
- Delivers (another part of) the value of testing to the project team
- Separates message from messenger
- Builds tester credibility with managers

Change Management

What is it?
- Balancing risks and costs of changes/bug fixes with opportunities and benefits
- Balancing schedule, budget, features, and quality
- Scheduling changes and fixes for particular test releases and/or system releases

Why does it matter?
- Allows testing and the project team to respond to what they’ve learned so far
- Selects the right changes in the right order (i.e., focuses our efforts on the highest ROI activities)
Okay, Rex, Now What?

- **Assess**
  - How are we doing now?
- **Recognize challenges**
  - Why is it so hard to do this testing stuff right?
- **Improve**
  - Where do we want our testing processes to go?

Top 5 Assessment Questions to Ask

- Is our testing effective?
  - Do our testers find the scary bugs, or do our customers?
- Is our testing efficient?
  - What is the return on our testing investment?
- Does testing pervade our projects?
  - Do testing and quality risk management start on day one?
- Does testing provide the information needed, when needed, to make smart decisions?
  - Have we considered test results in managing our projects and risks?
- Are our testing processes and people improving?
  - What data and feedback drives continuous improvement?
Top 5 Challenges to Good Test Processes

- Unrealistic expectations
  - Testing can provide credible, timely information, but it does not sprinkle magic quality pixie dust on systems
- Immature development or maintenance processes
  - Chaos leaking into the test process makes it impossible to deliver valuable information and services
- Adversarial relationships
  - Testing provides information, services to entire organization
- "Addicted to test"
  - Testers compliment—not replace—programmer unit testing
- Inadequate time, resources, or involvement
  - Too little, too late finds few bugs, only slightly lowers risk

5 Easy Steps to Improve Your Test Processes

1. Identify gaps and problems in your current test processes, and prioritize them
2. Plan the process improvement and sell participants and stakeholders on your plan
3. Implement the change and measure the improvement over time
4. Institutionalize the change so it becomes “the way things are done”
5. Go back to step one and do it all over again!

Done right, testing is a smart investment that saves money and reduces risk. Save more money, reduce risk further: Optimize the twelve critical testing processes.