

The Role of the Tester

Hermann Lacheiner

Thomas Reiter

Christoph Rieseder

Werner Putschögl

The Role of the Tester

- ◆ The Tester is the headlights of the project
 - Testing is done to find information.
 - Critical decisions are made on the basis of that information
- ◆ Tasks done by the tester are based on the
 - Project
 - Team
 - Company

The tester has many clients

- ◆ The project manager
- ◆ The programmer
- ◆ The technical writer
- ◆ Technical support
- ◆ Marketing
- ◆ Top management
- ◆ The user

What to test

- ◆ One of the most important tasks is to find important bugs fast
- ◆ Ways to achieve this are to test...:
 - Things that are changed before things that are the same
 - Core functions before contributing functions
 - Capability before reliability
 - Common situations before esoteric ones
 - Common threats before esoteric ones
 - High-impact problems before low-impact ones
 - Most wanted areas before not requested

dos...

- ◆ Quick, short feedback loop with programmers
- ◆ Focus on failure
 - Do not redefine your mission to a more positive, productive view
- ◆ Question everything...

...and don'ts

- ◆ ...but not necessarily out loud
- ◆ Think you can find all the bugs
- ◆ Test completely
 - For one reason see previous argument
 - Completely can mean very different things, depending on whom you ask:
 - ◆ Completed the discovery of every bug
 - ◆ Completed the agreed upon tests
 - ◆ Completed of usefull und cost-effective tests
 - ◆ ...
- ◆ The tester doesn't assure quality
 - The programmers create the software and the tester focuses on finding errors they make
- ◆ Become an process improvement group
- ◆ Become the „gatekeeper“ of the project

Thinking like a tester

◆ Technical thinking

- Understand causes, effects
- Predict the behaviour of systems

◆ Creative thinking

- You will look only for problems that you can imagine exists

◆ Critical thinking

- The ability to detect errors from your thinking

◆ Practical thinking

- Applying test tools
- Making test techniques

Exploring

- ◆ Is a detective work:
 - Forward thinking: know → don't know
 - Backward thinking: know ← suspect
- ◆ By induction
- ◆ Everything is a conjecture
 - Knowing if a product works well → refute that it works well
- ◆ Consider circumstances
 - It works is ambiguous without further qualification
- ◆ Statusreport of product quality should include how you tested and the known limitations of your test process

Discovery of requirements

◆ Three information sources:

- Conference: Confer with someone whose opinion about quality matters
- Inference: Extrapolating from other things you know
- Reference: Discover implicit and explicit specifications

Testactivities

◆ Configure

- Prepare product for the test

◆ Operate

- Inputdata, Commands, Interaction

◆ Observe

- Collect information about behaviour

◆ Evaluate

- Apply rules, reasoning that detect bugs

Heuristics for testing

- ◆ Test at the boundaries
- ◆ Test every error message
- ◆ Test configurations that are different from the programmer's
- ◆ Run tests that are annoying to set up
- ◆ Avoid redundant tests

Biases

- ◆ Assimilation
- ◆ Confirmation
- ◆ Availability
- ◆ Primacy
- ◆ Recency
- ◆ Framing effect
- ◆ Prominence
- ◆ Representativeness
- ◆ → compensate in your thinking!

Interaktion mit Programmierern(1)

- ◆ Verständnis der Denkweise von Programmierern
 - Die meisten Programmierer spezialisieren sich
 - Programmieren ist eine komplizierte Aktivität
 - Programmierer schätzen nicht Routine Tätigkeiten
- ◆ Vertrauen der Programmierer gewinnen
- ◆ Unterstützung anbieten
 - Third-party Komponenten testen
 - Prototypen testen
 - Test Umgebungen für Programmierer erstellen
 - Anforderungsdokumente auf Testbarkeit prüfen



Interaktion mit Programmierern(2)

◆ Respekt durch Integrität und Kompetenz

- Fehler klar und knapp berichten
- Beurteilung des beobachteten Verhalten des Produktes
- Bei nicht reproduzierbaren Fehlern Reproduktionsversuche zeigen
- Schlechte Nachrichten direkt überbringen
- Wissen nicht vorgaukeln
- Fehlerberichte nicht übertreiben



Interaktion mit Programmierern(3)

- ◆ Konzentration auf die Arbeit, nicht auf die Person
- ◆ Programmierer reden gerne über ihre Arbeit → Fragen
- ◆ Programmierer wollen gut testbare Programme
 - Verwendung der gleichen Sprache
 - Fragen frühzeitig stellen
 - Realistisch sein

Your Career in Software Testing

Thomas Reiter

Overview

- ◆ Technical and/or Managerial career path
- ◆ Extend your career beyond testing
- ◆ Career Management / Job Hunting
- ◆ Improve your skills

Technical career path

- ◆ Automation programmer
- ◆ Automation architect
- ◆ Performance and scalability tester
- ◆ Systems analyst
- ◆ User interface / Human factors analyst and critic
- ◆ Test planner
- ◆ Subject matter expert
- ◆ Black-Box Tester

Managerial career path

- ◆ Test lead or supervisor
- ◆ Test manager
- ◆ Director of testing or director of quality
- ◆ Internal consultant
- ◆ External consultant

Shift into other management positions

- ◆ Program / Project manager
- ◆ Technical support manager
- ◆ Product manager (for domain experts)
- ◆ Documentation group manager
- ◆ Sales support manager

Process Management:

- ◆ *Software metrics specialist*
- ◆ *Software process improvement specialist*

Extend your career beyond testing

- ◆ Move in and out of the field:
 - More perspective / Better insight
 - Flexibility
 - Job security
 - Market appeal
 - More credibility
 - Higher salary

Career Management / Job Hunting

- ◆ Conferences
- ◆ Keep in touch
- ◆ Testers always see what's going wrong
- ◆ Look for a different job
- ◆ Build a portfolio
- ◆ Resume (functional / historical)
- ◆ Learn about companies
- ◆ Prepare well for job interviews

Improve your skills

- ◆ Learn Perl
- ◆ Learn Java or C++
- ◆ Test Demos of testing tools
- ◆ Improve writing skills
- ◆ Improve speaking skills