Presentation

Bio

Return to Main Menu

P R E S E N T A T I O N

W15

Wednesday, Dec 6, 2000

Testing: A Heretic's View

Rose Marie Subasic

International Conference On
Software Testing, Analysis & Review
DEC 4-8, 2000 • COPENHAGEN, DENMARK

Testing: A Heretic's View

Rose Marie Subasic

Quality Manager
European Consulting Services
MDL Information Systems AG

Let’s start by cheating a little first ...

- E’ “Schnuppe” vom Lääbe in dr Schwyz

IN MY OPINION

- System testing as a profession is a beautiful, challenging, and worthwhile art form. ***It is perhaps the single most valuable function in the entire software life cycle, since it is the one thing that gives us all hope that life can (and generally does) get better.***

THE REST OF MY OPINION
"The Heresy"

- ***System testing as a specific software management goal should die as quick a death as possible***.
- This includes, but is not limited to:
  - Zero defects
  - Fully automated testing
  - 100% coverage

Example from real life

- The testers
  - Fully automated testing *and reporting*
  - Essentially 100% new feature coverage
  - Multiple platforms & versions (op sys/RDBMS)
  - Full regression cycles
- The results
  - 10 - 12 months late
  - Re-release required within weeks

Having learned from the previous release ...

- Everyone vowed that things would get better,
- however...
- What do you think????

What Went Wrong (Again)?

- What happens when release is late?
  - Code late to testers (again)
  - Testers find too many defects
  - Iterate as needed

What Specifically Went Wrong the First Time

- Backward functionality regression tested (old features still worked),
- however ...
- Backward compatibility not adequately tested (new release / old database)

What Essentially Goes Wrong Every Time

- What happens when a release is late?
  - Customers are unhappy
  - Sales and marketing prepare for battle
  - Development is exhausted, with low morale
- Worst case
  - Customers shop elsewhere (?)
  - Company loses loyal employees (?)
  - **Company loses its investment**

Why Did It Go Wrong?

- Asking the wrong questions
- Making the wrong changes
- ***Aiming toward the wrong software management goals***
- ***Ignoring human nature***

Another Look at the Heresy

- Proposition
  - ≠ System Testing should die
  - but rather, that System Testing *as a Major Project Goal* should die
- ***We must stop managing projects by technical feature-based risks***
- ***We must begin managing projects based on overall business goals***

Proposed Universal Project Goals

- Company makes enough money
- Customers remain loyal
- Project members gain satisfaction

- **Every project activity MUST map to one of the above**
  - **Otherwise, take it off the list**

Rules for Specifying Project Goals

- A few simple questions:
  - What went wrong last time?
  - What's going wrong now?
  - What's going right now, that would cause a disaster if it started to go wrong?
- In the context of, " If it goes wrong...":
  - How much will it cost to fix? (money, time, effort, customer satisfaction, intellectual property)

Intellectual property is PEOPLE, not code

How to Fix Things Depends on Project Scope

- Small projects
  - Requires acting quickly, and without fear
  - Okay to following instincts
  - Position it as “experimentation”
- Large projects
  - Must be based on metrics and history
  - Must be done right the first time

How to Get Started

- Small or Wholly Internal Projects
  - Implement as you go, making several changes at once, when possible
  - New process for this project becomes standard for the next
  - Focus on “willing victims”
- Essential attitude
  - **Okay to back up and start anew**

How to Get Started

- Large or Global Projects
  - Analyze past failures and choose one or two
  - Make a thorough analysis
  - Formulate a specific plan
- Crucial requirement
  - **Get a Sponsor**

Small Project Successes

- Tools driven by the Human Factor
  - Issue Tracking - Excel Spreadsheet (demo)
    - Simple enough for everyone to register, filter, and search
    - Useful for both Development and Management
  - Project Quality Plan - MS Word Document
    - Common definitions of all project terms
    - Documents the “Social Contract”
    - Explicit criteria for code transfer and acceptance

Small Project Successes

- Standards aimed at Business Issues
  - Default Hardware & Software Configuration
    - Desktop and Server
    - ***Okay to change, but must be documented in PQP, with signatures!***
- Strategies relying on Individual Histories
  - Creative short-term use of existing personnel
    - Quality Manager still remembers how to write test specs and code
    - Project Manager enjoys ‘poking around’

Large Project Successes

- Tools that reflect Personal Commitment
  - In response to the loss of 3 person-year's worth of test scripts
    - Source Control System created by a top programmer, who became part of the test team during the design and implementation
    - Temporary backups done by Testing while replacement backup system being implemented
  - More important benefit
    - Team spirit lasted well after project completed

Large Project Successes

- "Stupid Question" ⇒ Brilliant Solution
  - After numerous "fire alarm" corrections to a particular program area, tester investigated problem with programmer, and lobbied for feature redesign
    - Redesign took about 1/5 or less of the time already spent fixing the defect
    - After redesign, no customer-reported defects registered against this feature to date
  - Sponsor (V.P. Development) was essential

“Crossing the Chasm” - The Eternal Paradox

- Definition of a good release
  - Smart Programmers
  - Great Marketing
  - Successful Sales Force
- Definition of a bad release
  - Lousy testing

“Crossing the Chasm” - Taking up the Challenge

- CHALLENGE - We must find a way to move the SKILLS of System Testers into a position that is higher up in the Software Development “food chain”
- **Translation - Since it’s going to be our fault anyway if anything goes wrong, take the risks required to truly improve things. BE BOLD.**

My Personal Bold Moves

- **QA reviews of Business Document CONTENT** (Requirements, Project Plans, Sales Quotations, etc.)
- **Refuse to Maintain the "Status Quo"**
  - **No Code Released** without required quality documents in place, *with signatures*
  - **No Resources** without required planning documents in place, *with signatures*
  - **No Testing** if code quality too low

More Bold Moves

- **When a manager asks what I think, I tell the explicit truth, even if he/she is at the top of the organization.** (After all, they don't write my review at the end of the year...)
- **Stage a 'silent revolution' by doing.** As the American saying goes: *It is better to ask for forgiveness than for permission.*

Parting Advice

- Look for Universal Languages, especially on cross-department / global projects
  - My favorites: Wine/Beer/Ice Cream/Chocolate
- Always say “Thank you”,
  - even when the answer is “No”.
- Never give up hope.

Wednesday 6 December 2000

W15

Testing: A Heretic's View

Rose Marie Subasic

Fourteen years industry experience including microcomputer programmer, technical writer, customer support analyst, product administration training expert, Sr. tester specializing in executive programs and chemical searching libraries, and current role as QA Manager for Consulting Services.

In past three years, role has included, depending on the project:

- *creation and/or standardization of documentation and/or tools (e.g. quality plan, test descriptions, issues tracking, code turnover);*
- *resource scheduling of programming and test staff;*
- *quality reviews of business documentation (e.g. requirements, sales quotations, customer presentations, project plans)*
  and an assortment of other strategic tasks for a total of nearly fifty projects, with durations of one month to one year.