

# Tackling Barriers in Multi-Customer Contract Acceptance Testing

Maaret Pyhäjärvi, Ilmarinen

Email: <maaret@iki.fi> | Twitter: maaretp

# *Or* “Why Can’t You Just Let Us Test!”

Maaret Pyhäjärvi, Ilmarinen

Email: <maaret@iki.fi> | Twitter: maaretp

# About the Speaker:

## Experiences in testing and being a tester on varied basis



- Started with testing -95: localization testing for greek version of an English software system
- Subcontractor-side on various projects: moving from localizations to functional testing
- Tried being in a developer role after thinking testers are not respected: the problem is not per role
- Teaching testing at HUT, giving public presentations and courses on testing
- Researcher in testing, consulting various organizations, teaching, being a test manager and a tester, with and without test automation
- At F-Secure with product business 2005 – 2008
- Moved to insurance pension sector, first with subcontractor side, now on customer side (Ilmarinen) as test manager / test specialist.
- Doing testing-related training on the side of a normal day-to-day job to get to meet brilliant testers in Finland.

# About the Speaker:

Experiences in testing and being a tester on varied basis



- Started with testing -95: localization testing for greek version of an English software system
- Subcontractor-side on various projects: moving from localizations to functional testing

**TESTER AND A WALKING TESTING DICTIONARY**  
**AGILE PROJECTS @ F-SECURE**  
**PLAN-DRIVEN PROJECTS @ ILMARINEN**

being a test manager and a tester, with and without test automation

- At F-Secure with product business 2005 – 2008
- Moved to insurance pension sector, first with subcontractor side, now on customer side (Ilmarinen) as test manager / test specialist.
- Doing testing-related training on the side of a normal day-to-day job to get to meet brilliant testers in Finland.

# The Project Scope



Procured for us  
in mid-80's  
(Cobol)

Procured for us  
in 90's  
(Cobol, J2EE)

Procurement for  
3 customers  
ongoing  
(Java/ Soap)



Procured since 80's for 14 customers  
(Cobol, Java/Soap)

- One vs. 3-14 owner organizations, multiple contractors
- Contract-acceptance testing in the end of the project
- Data-intensive system

# The Acceptance Testing

- Contractually "must act as a single customer" towards the contractor
- Needs in use same on principle level, practice does not follow the principle strictly
  - Testing-wise everyone tests how they use the system for acceptance
- Shared reporting, based on scripted testing methodology
  - Long test planning phase with reviews, counting numbers of passed / failed / not run tests

# Wanting to Try Something Different

## **Management in own organization**

- with limited people hours available, buy-in for exploratory testing was not difficult (1:3 investment ratio)
- Most relevant factor: reputation for test management

## **Individual contractors**

- Acceptance testing is customer territory, reporting
- Most relevant factor: contract not requiring detailed test cases with fixed data

## **Customer procurement facilitator organization**

- no buy-in – request denied. Everyone must work with same methodology and formats.

Coaching

# Framework of Management



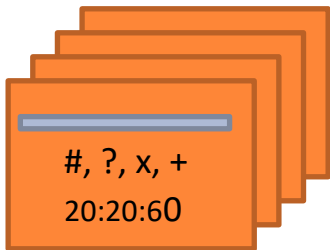
Test Manager

Debriefing

Past  
Results  
Obstacles  
Outlook  
Feelings

Metrics  
summary

Session sheets of the past testing



Playbooks

Perception of  
quality and  
coverage



Tester

"A day's work"

Vision ("Sandbox")

Current Charter

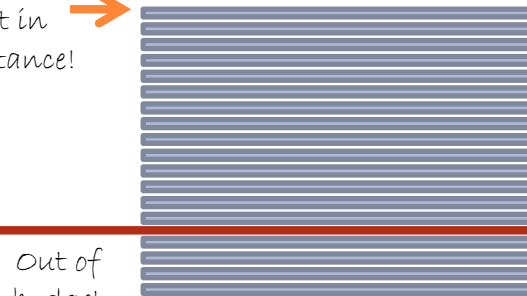
Other Charters

Details

#  
?  
x

Charter backlog of the future testing

Next in  
importance!



out of  
budget

Idea of  
exploration

Quality  
Report

Bug  
Reports



# The Pieces in Management Framework

- A disciplined tester replanning on various levels
- Session with charter that provides a report
- Classification of information created as metrics
- Prioritizing of what test idea comes next
- Supporting reporting by debriefing
- Supporting skills development by coaching
- Supporting future needs to remember with playbooks
- Creating a combined judgement of quality by quality reporting

# What We Did

- Phase 1: Inform and negotiate
  - Suggested test data oriented metrics, different form of documenting (not from spec but from data)
  - Result: ultimatum - our way or not at all
- Phase 2a: Scope down the visible scale of testing & documenting
  - Example area: 53 test cases (P1 – visible) and 184 (P2 – not visible)
  - Minimal energy principle in reviews: "thanks for feedback, we decide if we change or not"
  - Disobeying: "add test case before writing a bug report"
  - Our written test cases were bad – just as anyone else's except for one
- Phase 2b: Defer commitment with test cases
  - Vacation as an excuse, then delivering 2 documented that included 255 individuals after execution
- Phase 3: Add visibility to the hidden
  - Start showing P2 numbers in reports, without explaining they don't mean the same
  - One-on-one discussions within customer organizations to avoid duplication
  - Supporting our own manager's continuously with information of value vs. cost. They needed to know more than usually, playing politics.
- Phase 4: Followup
  - Continuously emphasizing the opportunity cost
  - Explaining what really happened & results

# Lessons for Improving Our Testing

## (1/2)

- Adding and dropping tests
  - Vague idea of number of tests made it easier for testers to not do things that were not valuable and add things that were needed
- Testing halves-of-tests
  - "Start of process" could be tested much earlier in non-data destroying way than the whole process
    - The user interfaces may just not be as ready as they should...
- One test to find things in the masses to guide other tests
  - Identifying one out of "essentially same" is different helped in focusing more effort on the one that needed the effort without wasting it equally on all
- Creating unplanned scenarios with production-like use
  - Identified crashing problems that did not exist until enough times the system was inaccessible for days during testing

# Lessons for Improving Our Testing

## (2/2)

- Working with data that was inaccessible due to rights until the system under test would deliver it
  - Identified data that we did not seem to ever get
  - Efficiency in test data selection process (avg 2 man-days / data sample)
    - data outdates easily
- Spec-based test cases miss relevant information for us
  - Spec → defect – contractor pays ; Other → change – customer(s) pay
- Regression testing became relevant for us
  - Not focusing on checklist, but check after various changes
- Replacing "test case" with a "test day"
  - Time for testing if did not need to stop for bugs
  - Guiding experts to disagree on scope and resolving the disagreement

# The Excuses

- It's not us that want these information, it's ***your*** managers
  - Managers did not say how they want they want the information, they just wanted to know if we'll make it.
  - Lack of creativity in the test management layer is eminent, not enough practical examples how how (without detailed session based testing) you could control it.
- Coordinator had little idea of how the customer organizations will use the information system provides, could not control based on contents
- End users as tester's will not tell what they did if not planned in advance and provided as reference
  - Test cases needed in bug reports

# The Real Barriers

- Collaborating with various groups that are testing simultaneously
- Controlling the schedule
- Contractual allocations of responsibilities and costs

Many possible solutions without compromising these.

# To End This With

- A long way to go
  - Allowing us to do exploratory testing constantly
  - Enabling our organization not to use the permission as not preparing
  - Enabling other customer organizations to do exploratory testing within their acceptance testing
  - Enabling the contractors to exploratory testing in any scale
- When asked to do a routine that does not provide value, try first doing less of it