



# TESTING IN CHALLENGING PROJECTS

ANTHONY KLING

CONSULTANT

SMS MANAGEMENT & TECHNOLOGY

# MALFUNCTION 54

- 1985 – 1987
- Therac 25 Radiation Therapy machine
- 6 accidents – patients overexposed to radiation
- 3 died as a result of radiation poisoning

# MALFUNCTION 54 - CAUSES

- Lack of testing and quality assurance
- Poor design, user documentation
- Reuse of code from older model
- Assumption that previous models worked safely

# DOES HISTORY REPEAT

- 2000-2001 National Oncologic Institute - Panama
  - 17 patients die, 11 injured
  - Operators used a shortcut – a software condition never tested
  - Physicians charged with manslaughter
- Ariane5 explosion
  - Reused code from Ariane 4
- Mars Climate Orbiter & Mars Polar Lander
  - NASA motto 1990s: Faster, Better, Cheaper
  - 30% reduction in project costs – reduced testing
  - Lack of end to end testing

# SO WHY TEST?

- What is the real impact if software fails?
- What is the real cost if software fails?
- Do organisations really care?
- Why is testing so hard at times?

# TESTING CHALLENGE 1 – TINY TEST EFFORT

- Testing already estimated
- Tester comes in late in project
- 1 week planning,
- 1 week execution
- 1 day to fix defects
- And it's an 18 month project!
- Who came up with that estimate?

# TESTING CHALLENGE 1 – TEST ESTIMATION

- What are all the test tasks required for a successful test?
  - Plan for test case development / design
  - Sizing – how many tests need to be run
  - What is the most effective sequencing / order of tests
  - Parallel vs sequential execution
  - Environment set up
  - Test tool set up – includes stubs, harnesses, automation
  - Training

# TESTING CHALLENGE 1 – TEST ESTIMATION

- Do we estimate for test failure?
  - Anticipated test pass rate - Not all tests run the first time
  - Not all bug fixes work first go
- Test cases are not static
  - Plan for test revisions
  - Additional tests (unplanned) may be required
  - Factor in a Level of exploratory testing
- Provide the facts – document test assumptions
- Perform a risk assessment – what can be left out



# TESTING CHALLENGE 2 – ERROR PRONE DEVELOPER

- To often assumption – developer is OK
- To often – developer delivers poor quality code
- What is the impact to the test effort?
- What additional testing is required to point out build deficiencies?
- Code fixes continuously break other parts of the system
- How much regression testing is required?
- UAT is too late to conduct unit testing

## TESTING CHALLENGE 2 – TIPS

- Provide the facts – defect tracking
- Be honest – more testing will now be required
- The customer is your ally
- Beware the change request argument
- The need to increase Regression testing
- The need to increase Exploratory testing
- The need for Post Verification Testing (PVT)

# TESTING CHALLENGE 3 - VERY LATE PROJECT

- Project is very late – way over schedule & budget
- No solution delivered yet
- Project team turnover
- Pressure to test quickly – get it over the line
- You discover
  - Little evidence of previous testing
  - Lots of testing, but does not relate to requirements / project objectives
- What is required to test the solution?

# TESTING CHALLENGE 3 - TIPS

- Traceability – what were the original objectives, requirements
- Need to show that objectives have / haven't been met
- Re-estimate remaining test effort
- Stand Firm
- Need to demonstrate the risk
- Project may fail

# TESTING CHALLENGE 4 – DELIVERY BULLYING

- Project has fixed deadline
- Requirements and design are delivered late
- Build is delivered late
- Your 3 months of planned testing is now 1 week
- You will deliver or else!
- Pressure to cover up the defects

# TESTING CHALLENGE 4 – DELIVERY BULLYING

- Understand the risk of NOT testing
- Use what if it fails scenarios - risk
- Assess risk priority
  - what needs to be tested first
  - What testing can be dropped
- Clearly document the issues
- Stand firm
- Test Summary Report is your friend
  - What could go wrong because we didn't test
  - What doesn't work now and the consequences

# KEY TOOLS TO SUPPORT TESTING

- Learn to adapt – projects are different
- Traceability – the basis for testing
  - Demonstrate that the business need has been tested / validated
  - Shows the gaps
- Test Estimation toolkit
  - Detail what really needs to be done to test adequately
  - Assumptions / Arguments to support your estimate
- Test Summary Report
  - Is the solution ready?
  - Risk assessment on what hasn't been tested – What if it fails
  - Risk assessment on what doesn't work



# QUESTIONS

TESTING IN CHALLENGING PROJECTS

5/6/2015

16

