



**QUALITY INTELLIGENCE**  
Intelligence about software quality, intelligently applied

---

# **Make Risk-Based Testing A Reality**

**Fiona Charles**

***KWSQA Targeting Quality Conference***

**April 23, 2008**



## Who I Am:

**Fiona Charles** - Test Manager and Consultant, President of Quality Intelligence, Inc.

- 25+ years in IT
- QA & Test Management on software development and integration projects for clients in retail, banking, financial services, health care and telecommunications
- Managed E2E Systems Integration Tests on several large programs, including:
  - A retail frequent shopper program
  - An online store for a major retailer
  - A bank teller application
  - Internet television



## Who I Don't Want To Be

### A Talking Head!

Let's make this as interactive as we can

- Please break in at any time if you have a relevant question or comment, or a tip to share
- If we can address it quickly, let's do that and move on.
- If your issue or question needs more time, I may ask you to park it until the end, when we can give it the appropriate focus.

**We'll all learn more by sharing our knowledge and experiences!**



## Today:

**What's it all about?**

**Stakeholder ownership**

**Identifying & assessing risk**

**Using the assessment**

**Wrap-up**



## What's it all about?

Stakeholder ownership

Identifying & assessing risk

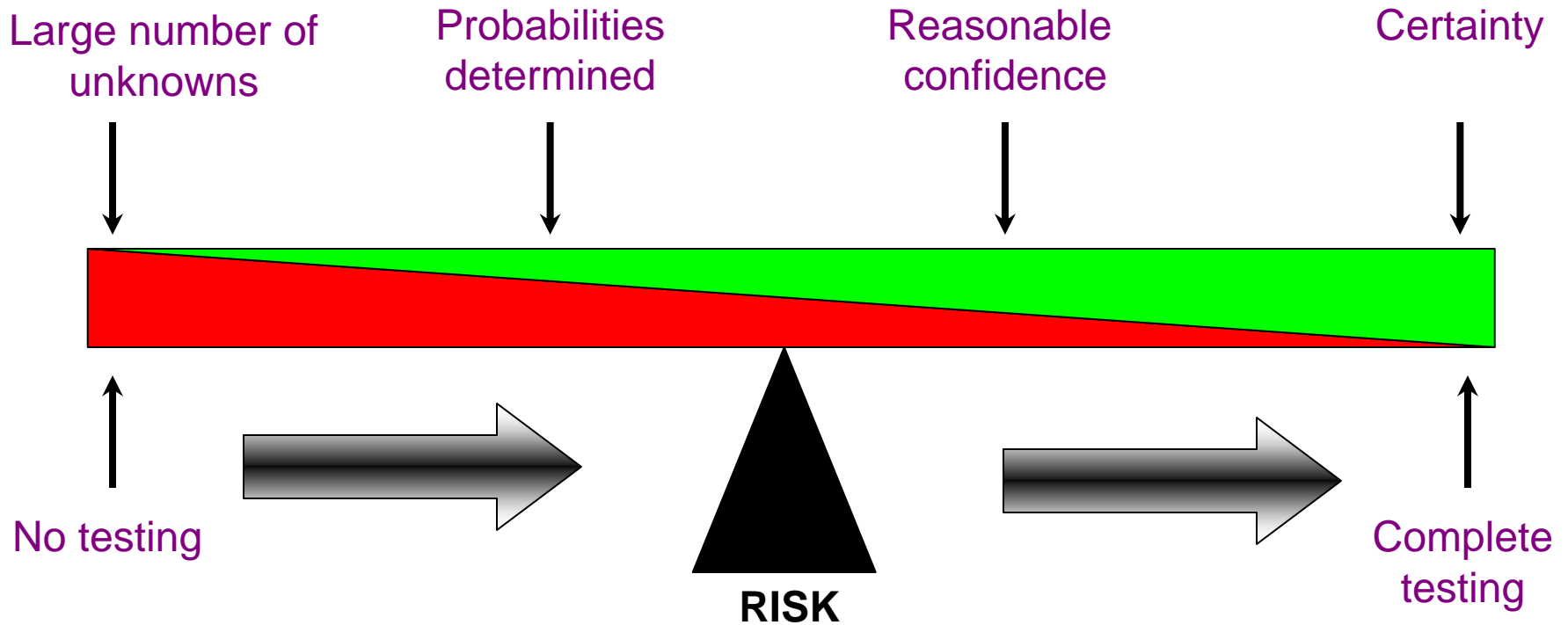
Using the assessment

Wrap-up



# Testing provides information about the risks of implementing software

The challenge is to balance the level of certainty achieved with the level required.





## We can never do “complete” testing, so we look for guides to adequate coverage

- Requirements-based testing
- Path coverage
- Model-based testing
- Scenario-based testing...



# The impacts of software failures can be major—even catastrophic—for an organization or its “victims”

- Patient death (+ hospital sued)
- Massive financial losses
- Customers’ private records exposed
- Banking errors strewn all over the headlines...

**Most impacts fall between catastrophic and annoying**

---

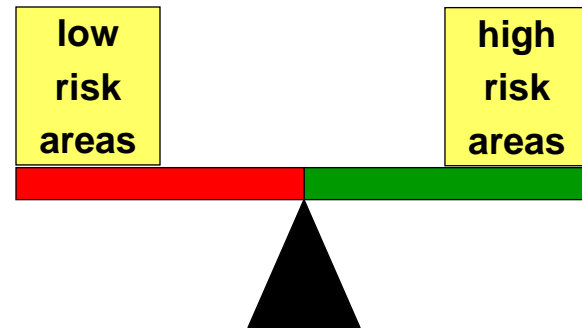
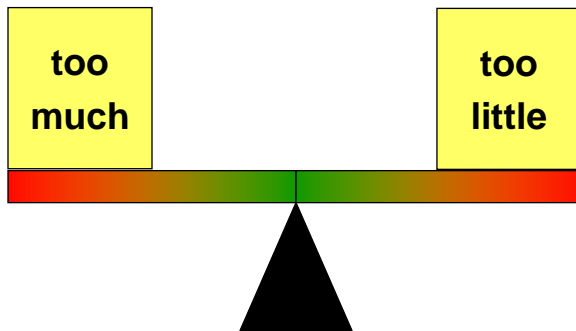


# Risk-Based Testing matches testing effort and coverage to the risk in the system(s) being tested

It helps to ensure that:

- the right things are being tested
- in the right place in the project
- to the required level of rigour

while avoiding unnecessary effort on low-risk components.





## Other benefits

- Understanding what could happen stimulates you to design tests for those things
- Manage stakeholder expectations
  - Not everything is equally risky
  - Take real risks seriously
- Demonstrate due diligence for the historical record



## Risk-based testing addresses only one category of risk

- Not project risk
- Not even testing project risk
- The risks inherent in a software system

**RBT targets the implemented system's potential for causing harm**



## A systematic process for RBT has three essential components

- Risk identification and assessment
- Stakeholder ownership of the assessment
- Test strategy based on the assessment



What's it all about?

**Stakeholder ownership**

Identifying & assessing risk

Using the assessment

Wrap-up



## Developing a risk-based test strategy should be a collaborative effort by the stakeholders:

- Driven and documented by the testers
- Requiring participation by the other experts on the project:
  - Business sponsor and system users
  - Project Management
  - Solution Architecture
  - Development
  - Database Management
  - Business Analysis

**Correctly identifying key stakeholders is critical**

---



## Stakeholders have knowledge to share

- About the system
  - Features and functions
  - Technical components
- About the project
  - Requirements
  - Changes it will introduce to the system
- About their priorities
- About system problems this project could introduce
- About the business impacts of potential problems



## You are asking stakeholders to own:

- The concept that you cannot test every condition in every component of the application with equal rigour
- The identification and assessment of risks
- Agreement that directing testing according to the risks will provide adequate coverage.



What's it all about?

Stakeholder ownership

**Identifying & assessing risk**

Using the assessment

Wrap-up



# Steps for developing a Risk-Based Test Strategy

1. Identify and assess the system risks
  - Opportunities for harm from system problems
  - Probability of potential harm occurring
  - Business impacts of system problems that are likely to occur
2. Identify all the test activities that could provide the required levels of confidence
3. Decide who should perform which test activities and when



## Conducting a workshop with the stakeholders is a good way to do risk assessment

- Makes stakeholders integral part of the process
- Can be scaled for different sizes of project
- You can document the result and get signoff



## Prepare for the workshop

- Get the right people (stakeholders & techies)
  - If it's too big a group, split them up and have one final confirmation session
- Explain why, what and how (high level) in invitation
  - Make people want to come
- Design the workshop approach



## Your approach will depend on the project type

- Prepare questions oriented to what is new or changing
- Design a structure for capturing the assessment that is appropriate to the project type
- Identify key stakeholders



## New application or functional enhancements – risk assessment matrix

- A high-level business view of the system features will be most meaningful to all participants
- Avoid a too-detailed view
- Include functions, features, pages, screens, reports and interfaces
- Include any other systems that may be impacted
- Socialize the matrix before the workshop,



# Matrix example (health care)

Application: Example Release 1 28-Aug-06

		Risk Levels		
Module/Functional Area/Screen	Process/Function	Functional	Usability	Notes (Concerns,etc.)
Live data load	Master Patient Index Backload		Economy	Duplicate patients, patients not listing, wrong information for a patient
	Materials management initial load			Nightly load validation, ensure new data is passing over
(manual load)	MSM Future Appointments	Patient Critical		Validating the appointments from MSM to SIS
Preference cards	building	Clinical Business Normal		
	linking to procedure	Economy		
<b>Scheduling</b>				
Scheduling	Scenario based - patient types	Patient Critical		Not scheduled patients
Scheduling	Remote access to cases -user types	Clinical Business Normal		VPN, check patient scheduling remote to hospital. Not scheduled patients
Scheduling	Scheduling a case and multiple cases.	Economy		
Scheduling	Scheduling a wait time case.	Economy		
Scheduling	Moving a case., waiting area	Clinical Business Normal		
Scheduling	Cancelling a case.	Clinical Business Normal		
Scheduling	Testing the available tabs.	Economy		
Scheduling	Building and maintaining blocks	Economy		Only 1 person building blocks, set blocks
Scheduling	Case maintenance	Patient Critical		manul entry of MRN numbers, some determination on how to handle this is still being worked out
Scheduling	Scheduling reports	Clinical Business Normal		OR Schedule, waittime report



## Conducting the workshop

- Clear roles
  - Facilitator
  - Scribe
  - Tech reps
  - Business reps
- Explain what you're doing and why it matters
- Brainstorm on what risk could mean in this application
- Work systematically through the matrix
  - Project on screen, update as you go
- Everyone owns the result



## Initial brainstorming - risks overall

- What kinds of risk matter in this organization?
- Are there “givens” we can start with?
- Who could be hurt and who do we care about most (risk hierarchy)?
- How could this application hurt each of those?
- What’s the worst thing that could happen?
- Which aspects of quality should we look at in this application?



## Starting points example – retail organization

- We must be certain that pricing is always be accurate throughout the integrated systems (legal, financial, reputational impacts)
- We must be certain of the accuracy of financial data used in regulatory reports or in critical business decision-making (legal, financial, reputational impacts)
- We need a high degree of confidence that inventory numbers are accurate throughout the integrated systems (direct impact on bottom line, etc.)



## Walk through the application in detail

- Ask how each component in the matrix could break
  - What kind of system problem could occur (functional, usability, performance, security)
- If it did break what's the worst that could happen?
  - How likely is this to occur?
- What other things could happen?
  - How likely are these?
- How often is this system/component/function used?
- Where is it used, and by whom?



## Infrastructure changes or upgrades – some questions

- What are we upgrading/changing/removing?
  - Hardware (server, storage mode, client, desktop)
  - Operating system
  - Database engine
  - Compiler
  - Message broker
  - Batch scheduler
  - What else?
- Are we changing any code with the upgrade(s)?
  - Only as required for the upgrade(s)
  - Functional enhancements
- What configuration changes are required?



## For Infrastructure projects, focus on questions about technical risk

- Which technical components could be affected by each platform upgrade? How?
- Could any of the upgrades change or add restrictions to:
  - How data is stored? (compression algorithms, date formats, zeroes, etc.)
  - How data is transmitted? Between components? Between applications?
  - How data is accessed? (locking rules, etc.)
  - How calculations are done? (rounding, sequence, etc.)
  - Command or process sequences?
  - Interface mechanisms?
  - Permissions? (user and system)
  - Reserved words?
- If any of these could change, which system technical components depend on the existing modes, and could break or behave unpredictably with the upgrade?



What's it all about?

Stakeholder ownership

Identifying & assessing risk

**Using the assessment**

Wrap-up



## Risk assessment becomes a project document

- Can guide design and development
- Guides testing coverage and rigour
- Provides information for the Test Manager and Project Manager to make decisions if plans change, or project dates slip
- Guides bug fixing
- Informs go-live decision



## Using the risk assessment to drive testing

- Walk through the matrix with the technical team
- Decide what kind of test is best suited to each significant risk identified
  - Might NOT be a dynamic test; might not be you
- Map your testing to the matrix
- Use the matrix for reporting



## Some testing activities to consider

- Unit testing
- Data validation
- Under-the-covers technical verification (functionality or performance)
- Near-neighbour integration testing
- Exploratory testing
- Black/grey box system testing (functionality or performance)
- Black box end-to-end systems integration testing
- User Acceptance Testing (UAT)
- Post-implementation verification



What's it all about?

Stakeholder ownership

Identifying & assessing risk

Using the assessment

**Wrap-up**



## Risk assessment is a thinking process

- Getting an accurate assessment of risk is as difficult as getting any other type of requirements
- Tools, spreadsheets, etc. can be useful, but they are no substitute for thought. You should approach each situation as if for the first time, and not let canned questions stifle fresh thinking.
- In a risk workshop, the most vocal and the best negotiators can prevail, regardless of actual risk.
- Both probability & impact are subjective conclusions. It's the credibility of the source that matters.



## Things to consider

- Assess technical impacts as well as business impacts.
- The value of a risk can change over time.
- A risk assessment should be a living document throughout the project, kept up-to-date with the latest knowledge the team has about risks in the application or its interfaces.



# Wrap-up Questions and Discussion





## Credits

This presentation benefited from discussion at TWST 3 (the Toronto Workshop for Software Testing), held June 9-10, 2007, with the participants:

Josh Assad, Michael Bolton, Fiona Charles, Peggy Collier, Michael Cookson, Pierre Garrigue, Morven Gentleman, Kristin Goetz, Adam Goucher, Sherry Heinze, Paul Holland, Cem Kaner, Yuri Makedonov, Tomas Marchese, Robert Sabourin, Jon Steinberg, Adam White