Determining Business Risks for Testing

Fiona Charles

Originally published in the EuroSTAR newsletter for October 2009

Testers know that there will never be enough time or money to test everything. Testing must always be selective.

Risk-based testing is a logical method of directing where to focus test efforts—on the system areas where undetected defects could do the most harm. Done well, risk-based testing is reality-based testing. It directs testing efforts to what could go wrong in a system and the harm or loss that could result, while keeping testers from wasting time on low-impact areas.

But "risk-based testing" can be a buzz-phrase with little substance behind it, as testers are left to assess the risks without management and business support. Most testers are not trained to do this, and don't always have the detailed domain knowledge to know the true cost of a defect in the field.

At the very least, testers need to know what questions to ask about system risk.

What Do We Mean by System Risk for Testing?

Risk-based testing targets:

The potential for

…some fault, failure or other unintended happening

…in the implemented system

…to cause harm or loss

…to one or more persons or organizations.

To do risk-based testing, you need to identify and assess risks in the system, and then base your test strategy on the resulting system risk assessment. You also need to get key stakeholders to buy in, ideally by participating in the risk assessment process and taking ownership for the result. If you can't get their input and participation, you should at least get their review and agreement.

Identifying Stakeholders

Before you can assess the risks, you need to understand who the key stakeholders are for the system. Stakeholders come in two principal flavours: business (including outsiders to the organization) and technical (including IT and support).

To identify the **business** stakeholders, ask:

Who has an interest or concern in this software?

Who will benefit from it?

Who could be victimized by it (suffer harm or loss)?

You should consider (at least):

- Management—those who pay for the system and therefore have a "stake" in its successful operation
- Hands-on system users
- Users of system outputs
  - Reports
  - Data
- Customers
- Bystanders

To identify the **technical (or IT)** stakeholders, ask:

Who is responsible for designing, developing and delivering this system?

Who will operate this system in production?

Who will support this system?

People to consider include:

- Project manager
- Development architect and technical leads
- Computer operations (batch jobs, backups, etc.)
- Maintenance leads
- Help desk

Risk Assessment

Identifying and assessing system risk can be done as formally or informally as fits your context. On some projects, conducting a workshop with all the key stakeholders is the most effective and efficient way to develop a system risk assessment. On other projects, you may get a better result by interviewing stakeholders in small groups with common interests, or even individually. In still others, you may have to answer most of the questions yourself and then get the stakeholders to review your assessment.

Regardless of the interview format and participants, it's essential to conduct a systematic examination of the risks inherent in the system as it will be implemented. That means asking a good set of questions to identify and assess the risks—even if you have nobody but yourself to ask.

Business Representation of System Components

Come equipped to each discussion with a model of the application that will make sense to the stakeholders, and then spend a little time making sure they are comfortable with the model. If you're conducting a risk assessment workshop, discuss your model with participants before the meeting, so you aren't surprising anyone.

You can create a spreadsheet to capture the application model and corresponding system risk assessment.

Each row will represent one feature or component of the application. Before creating a matrix, it is important to think about how to structure it so that it represents the application features and components in a way that will be meaningful to all of the stakeholders and project team members. Otherwise, it may be difficult for them to see what the impacts of failures might be.

Some things to consider in creating the spreadsheet:

- The most meaningful model for everyone will likely represent the application from the business stakeholders' point of view. That might be by function, or screen, etc., with additional non-user components, such as interfaces to other systems, added on.

- Make sure your model covers all significant components of the application, and not just those that are high or medium risk. Rather than skipping the lower-risk areas, it is better to list them and document the stakeholders' agreement that they really are low risk. Consider for inclusion all the:

    - Pages/Screens
    - Functions
    - Reports
    - Interfaces
    - Etc.

- Don't drive to too low a level of detail, or you could make the risk assessment process excessively time-consuming and even counterproductive.

Why Ask Business People about System Risk?

The short answer is "because they know what will hurt them, and how it will hurt", (and typically, you don't). It's not for testers to identify those risks—get the experts, and then facilitate and/or ask questions.

Getting an accurate assessment of risk can be as difficult as getting any other type of requirements. Listen to the stakeholders. Ask them questions designed to stimulate their thinking.

You also need their buy-in to a risk-based testing strategy. If you don't get it up front, you risk after-the-fact rejection of your test efforts, even potential rejection of the system.

You are asking people to buy in to:

- The concept that you cannot test every condition in every component of the application. Of the ones you can test, you can't treat them all with equal rigour.

- The idea that directing testing according to risk is a way to contain cost, both upfront project cost, and the cost of significant bugs in production. This can be a way to sell the idea in organizations where it is new.

- Agreement that you have collectively identified the risks adequately.

- Agreement that directing your testing according to the identified risks will provide adequate coverage.

Where the Technical Participants Come In

Business people can tell you the impacts of system failures or bugs, but only the technical experts can tell you what could go wrong in a given component (though you may have some ideas about this yourself), and how probable it is that there will be problems in that component when you test it.

A Framework for Business Impact Questions

It's a good idea to begin each risk assessment session with a general discussion of what risk means to the participants. (Obviously, this is more important if you are new, or an outsider to the organization.) You don't have to nail it completely. You can get an idea of key issues to help discussion as you go through the application.

Sometimes it's useful to frame the conversation by asking **"What is the most catastrophic thing that could happen?"**

Could any failure or fault in this system or feature:

- Kill someone?
- Send your CEO to jail? e.g., by making regulatory reports wrong
- Hurt the stock price?
- Damage the bottom line?
- Hit the news?
- Cause you to rip off your customers?
- Drive away your customers?
- Violate customer privacy?
- Provide wrong data for critical corporate decisions?
- Disrupt a mission-critical process, e.g., prevent your bills from going out?

Once you've established the worst that can happen, you can work through the application one business component at a time, first asking the technical participants what could go wrong, and how likely it is that there will be a problem, and then asking the business how it could hurt.

Conducting the System Risk Assessment

General questions to ask for each business component are:

What could go wrong?

How likely is it that particular kinds of faults/failures could occur?

How might faults and failures that could occur hurt the business?

Could these faults or failures cause data loss or corruption, or other technical impacts that would be difficult or impossible to recover from?

Of course, these are just the high-level questions. You'll need to get much more specific when you assess system risk on a project, and that takes both practice and the experience of looking at risk in different business and project contexts.

Come to My Workshops on Determining Business Risk!

In my workshops and tutorials on Determining Business Risk (e.g., at CAST and EuroSTAR 2009), we explore different contexts and we practice doing risk assessments. We go into detail on each of these broad questions (and any others we identify), and work interactively together to build a list of useful questions everyone can draw on in their work. We also explore the whole question of stakeholders and how to identify them.

Some of our risk assessment practice uses high-level system bug descriptions contributed by workshop participants. So come prepared to describe a bug or two. And don't limit yourself to only high-risk bugs. It will be much more useful and fun for everyone if some bugs turn out to be medium or low risk—more like real life in risk-based testing, in fact!