

CITS5501 Software Testing and Quality Assurance

Semester 1, 2022

Workshop 8 (week 9) – Risk

Reading

It is strongly suggested you complete the recommended readings for weeks 1-8 *before* attempting this lab/workshop.

A. Risk classification

Every project, no matter how small, involves *some* risk. In the lectures, we looked at one classification of risks:

Project risks Risks that could affect the project plan or schedule, because they alter what resources (people, equipment, software, specifications) you have available to complete the project, how effective those resources are, or when they become available.

Technical risks Risks resulting from the problem being harder to solve than we thought. Often when these eventuate, they show up as problems with our system design, or with our decision on whether to code some component ourselves as opposed to outsourcing this (using an existing library or paying for an off-the-shelf software component). These risks alter the amount of work required to finish the project (given your available resources).

Business risks These are risks which if they occur would not make the project late, nor more difficult/expensive – but they would destroy the *business value* of the project (i.e. the benefit to be obtained from it).

For instance, suppose we were developing a software system to be used by the warehousing department of our organisation. If, part-way through, all warehousing responsibilities are outsourced, that would probably destroy the value of the project.

Work through the following list of risks for a software project, and decide which of these classifications each risk would fall into. (Some might fall into more than one). Justify your decisions, and then share as a class how you categorized each one and why.

- a. Management change: Your organization is bought by a larger company, and undergoes significant restructuring. The new management has very different priorities, and the project you are working on is low on that list.

- b. Documentation problems: You are working on a project which will create sales reports to be used by management. One requirement of the reports is that they will include complex statistical charts so that management can easily visualise sales data, and you decide to use an open-source library to create them, Shoshone Cybercharts. However, it turns out the documentation for this library is not adequate: your developers often can't be sure how the API is intended to be called, and what the resulting charts will look like, so it takes much longer to finish the implementation.
- c. Inter-team delays: Your company is building an Internet of Things (IOT) enabled toaster. Your team is developing an Android app which will be used to control the toaster, and another team is developing the toaster itself. By the time your team is ready to start implementation, you are still waiting on specifications from the toaster team which will describe the API your app should use to control the toaster. The specifications arrive three weeks later, causing delays in the project.

B. Risk identification

If you are attending this workshop face to face, then for this exercise you should split up into groups of 3-5 people. If you are attending online via Teams, your facilitator should be able to create “breakout rooms” for you, by following the [MS Teams instructions](#); but failing that, instead do a solo review of the code, and report back to the class as a whole with your findings.

Select one member of the group to describe a *project* they have worked on (or even better, are currently working on). If the group member is currently involved in a group design project, for instance from a CITS555X unit or CITS3200, that would be ideal. However, it needn't be a software project – moving house, buying a car or planning a holiday could all count (although you will need to be creative in applying the “Risk checklist” at the end of this worksheet, as it is geared towards software projects).

That student should describe the project, the resources available and the team involved to the other students in the group.

Then, read through the “Risk checklist” at the end of this worksheet. Each member of your group should try to identify one risk that could apply to the project. As a group, decide what sort of risk it is (project, technical, business), and what impact it could have if it eventuated (minor, major, show-stopper).

C. Pre-mortem

One reason projects can fail is because team members or stakeholders may be reluctant to speak up and identify possible problems, or reservations they have about the project plan. So it can be helpful to put in place an environment where people with the best knowledge about potential problems feel safe in speaking up about those problems.

One way is to undertake a “project pre-mortem” (first described by Gary Klein).¹ In a

¹Klein, Gary. “Performing a project premortem.” *Harvard business review* 85.9 (2007): 18-19. Available at <https://hbr.org/2007/09/performing-a-project-premortem>

pre-mortem, the participants imagine that the project has failed, and then work backward to identify causes that could have led to that failure.

For this exercise, split back up into your groups from exercise B, and consider the same project. Start by imagining that the project has failed spectacularly. Then, each person should (on their own) write down every reason they can think of for the failure, for 3-5 minutes. Then see if you have identified any risks which weren't on your list from exercise B.

This exercise doesn't give the full flavour of a project pre-mortem, because most members of your group are likely to be project outsiders, and may already feel safe about suggesting potential problems. Nevertheless, it introduces you to ways in which we can overcome reluctances or biases which might otherwise prevent us from identifying project risks.

Finally, share with the class as a whole whether your pre-mortem identified new risks you hadn't thought of in exercise B.

Risk checklist

Consider whether the following questions when trying to identify risks.

Scope

- Is the project scope stable? Is it incomplete or unclear?
- Do stakeholders demand additional scope, or have differing ideas of the scope?

Clients

- Is the client technologically sophisticated? Do they communicate in a timely manner?
- Do end-users have realistic expectations?

Requirements

- Are the project requirements stable? Are they unclear or incomplete?
- Have customers been involved fully in the definition of requirements?
- Are the requirements fully understood by the project team?

Project resources and schedule

- Does the project team have sufficient staff? Do they have sufficient experience with the technologies required? Do they have the right mix of skills?
- Are software or hardware items required easy to obtain? Do they meet expectations? Are they well-documented?
- Is the project behind schedule?
- Can resources be secured when required?
- Might staff become ill or otherwise unavailable?

Software development processes

- Is the design reasonably stable? Is it feasible? Practical? Does it contain the features and/or flexibility required?
- Does the organization have a well-understood software development process?
- Are software and project artifacts tracked by version control?
- Can the project team prototype technologies or areas which are poorly understood?
- Does the project team have access to tools of the quality needed for the project?
- Is the technology being used well-understood? Is it novel? Is it well-tested?
- Is the system ...
 - ... especially large?
 - ... especially complex?
 - ... required to be especially secure/safe/reliable?
 - ... unusual in some other way?