CITS5501 Software Testing and Quality Assurance
Semester 1, 2020
Week 9 workshop – Risk management

Overview

You will complete a short risk identification and assessment exercises in small groups.

Break into groups of 3-5 people.

Select one member of the group to describe a software project they have worked on (or even better, are currently working on). (If they are currently involved in a group design project, for instance from a CITS555X unit or CITS3200, that would be ideal.)

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

Then try one of the two following risk assessment processes, assigned by the lecturer.

Method A

- Identify possible risks (at least five), and try to classify them. (A possible set of questions to consider is included at the end of the worksheet, but you are welcome to use another if you wish.)
- Attempt to estimate the probability and impact of each risk.
- Identify the top three risks, in terms of probability × impact.

  Rather than assigning a numeric probability, it's recommended you use one of the following categories: very unlikely, likely, somewhat likely, likely, and highly likely (with a score from 1 through 5). Then assess the risks based on their likely impact on the project: insignificant, minor, moderate, major and catastrophic (again, with a score from 1 through 5). (See, for some illustrations of these, https://www.scu.edu.au/staff/risk-management/risk-process/risk-descriptors/)
- Identify some options for mitigating, monitoring or otherwise managing those risks. See if you can rank the likely effectiveness of those options.

Method B

Attempt to carry out what is called a "*pre-mortem*" on the project:

- Assume you are at a point in the future after the project was due, and that it was a disaster.
- Each person should write a brief history of that disaster.
- Then, compare notes, and see if you can combine them into a list of risks, and identify any others that might occur.

Then discuss as a class:

- What risks were identified?
- What risk strategies were identified?
- Did the two methods come up with any different sorts of risk?

Sample solutions:

Different groups will come up with different risks. However, in general:

- the "pre-mortem" is good at getting people to think outside the box, and identify risks they might not have thought of
- the standard risk assessment method is better for getting people to consider the more common risks, more thoroughly.

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?

Assessed exercise

You are part of the software testing team for TimeOff, a business which books adventure and recreation tours. You have been tasked with the testing of their new online booking system, which is currently being developed.

For each item below, provide:

- A description of what sort of test you would write to test it, and what techniques from lectures you would apply.
- Based on any reasonable assumptions about the system requirements, one example test case.

Justify your answer.

Items:

a. Test that the `addNewTourMember` method for the `Tour` class works properly in combination with the system database.
b. Test that the system adequately protects customers' credit card details.
c. Test that the system remains responsive under high load.

This workshop exercise is worth 5% of your final grade. All work is to be done individually.

Sample solutions:

One possible answer:

a. We are asked to test that the `addNewTourMember` method for the `Tour` class works properly "in combination with" the system database. Therefore, this is an integration test – we are checking that one component (the `Tour` class) works correctly with another (the database).

We aren't told exactly what the method does, but it seems reasonable to assume that if we call it with some sort of `TourMember` object, then that object will get added to the database.

We would not use a mock object for the *database*, since we want to test that it actually works. We *would* use fixtures (that is: state that needs to be set up for the test) – namely, some sort of test database containing bogus data, and a connection to that database.

b. We would need to perform security testing. This is often done through *penetration testing*: the tester adopts the role of an attacker, and attempts to obtain the secured data.
[NB: Other ways of validating the system's security are:
- thorough review of design and implementation
- use of security "best practice" checklists
- formal verification of portions of the system.]

c. We would need to perform *stress* or *load testing*. Typically this is done by generating high levels of random, but realistic, input. For instance, for this system, we might generate random web access to the system, which are as similar as possible to ones that might be performed by real users.
[NB: Specialised software exists to generate such traffic: see for some examples https://www.dnsstuff.com/network-traffic-generator-software and https://www.icir.org/models/trafficgenerators.html.]

Submission

You should submit your answers via `cssubmit` by 8 p.m. on Wednesday 6 May.

Hand-written and scanned answers are fine, as long as they are legible. Remember to write your name and student number on the answer sheet.

You should submit a 1-page A4 pdf file. (PDFs can be exported from MS Word and most other word processing software.) The font for body text should be between 9 and 12 points and easily readable when printed out.