



Department of Electrical and Computer Engineering

SENG 637 - Dependability, Reliability and Testing of Software Systems

Project Assignments

Behrouz Far
Winter 2010
(Revision 1.00)

Delivery Date for all projects: 13/April/2010 (Tuesday)

Outline

1. The deliverable pages should be numbered and each assignment should have a cover page bearing the assignment number, student(s) name and student(s) ID number(s), and submission date. The cover page is excluded from the minimum and maximum required pages.
2. Assignments must be delivered in electronic form either by email or on a floppy, data CD or DVD disks. For electronic submission formatted documents in MS Word, HTML or PDF are only acceptable. Using other word processing software is encouraged but the output should be saved or converted to one of the above mentioned formats. Unformatted text or LaTeX files should be compiled or converted to PDF before submission. HTML formatted files are acceptable only if accompanied by all the links and inline graphics, etc., in a single file (.mht file) or archive (i.e., zipped web archive).
3. Email submissions should bear the course number, e.g., SENG 635 in the **Subject line** of the message. The maximum size of attachment is 2 megabyte. Attachments larger than 2MB should be delivered on disks. Automated virus scanners may omit certain attachment file types (e.g., exe files, ZIP or RAR archives, etc.) for emails initiated by senders from outside of the UofC. It is always safer to send email with the attachments using students' UofC email account. Submissions not complying with this may be discarded with no action. All successful submissions by email will receive an acknowledgement receipt within 3 days of submission.
4. For electronic submission, the student is responsible to have the electronic media made readable. Defective archives will be rejected and marked zero.
5. For electronic submission, the student is responsible to have the files and attachments scanned for viruses before submission. Virus infected deliverables will be deleted automatically and marked zero.

Project: Software Reliability Project

Part 1: Defining Necessary Reliability

Grade: %25 of the total labs mark.

This assignment is a group assignment.

You must select your team members (1 or 2 members) and a project to build (list of projects is posted separately). Then each team should prepare a journal that the analysis, design, development and test documents will be added to it gradually. This helps communication among team members as well as reviews by the Instructor and/or TAs (if any).

Similar to many other projects, you start with the requirements analysis and then design. However, a big difference is that you must first finalize the requirements by defining what functions the system offers (functional requirements) together with what reliability criteria must be satisfied (non-functional requirements). In this assignment the focus is on the latter, which is usually ignored in conventional requirements analysis. To do so you must concentrate on defining the necessary reliability for the product. You can have a better understanding of the reliability centered requirements by answering the following questions:

1. How will you define failure for the product (by severity class, as product-specific as possible)?

Answer in detail what the failures that you expect the system handle are and how severe they may be.

2. Choose the natural or time unit you will use for the product.

Explain why you select such unit(s) and how to convert one to the other if you have more than one measurement unit. Time unit is useful when the operations of the system can be measured with respect to time, e.g., a server must be up 98% of running time. Other units, such as number of transactions, jobs, etc., may be used otherwise.

3. Set the product failure intensity objective (FIO).

Define FIO target for your product. If impossible to define it now leave it open for a later stage.

4. Find the expected product acquired failure intensity, based on the failure intensities of the hardware and acquired software components (if applicable).

Check whether your software should run on a specific platform (IBM PC + Windows xx, or SUN Sparc + Solaris 2.x, Linux, etc.), find the failure intensity for that platform if you can. Check if the maker gives the FI for its product. If not, try to guess or use the values given in the textbook.

5. Determine the product-developed software failure intensity objective (if applicable).

Straightforward by subtracting platform's FIO from target FIO. However, if you start by dividing your project into a number of independent modules, you should define FIO for each module and then add them up.

6. How will you balance fault prevention, fault-removal, and fault tolerance strategies?

Consider your specific requirements for reliability, timely delivery, and cost and allocate your resource percentages among the following six activities. You may add other activities to the list if they are significant, but the percentages must total 100.

- Fault prevention
 1. Requirements reviews
 2. Design reviews
- Fault removal
 1. Code inspection
 2. Unit test
 3. System test
- Fault tolerance (design for fault tolerance)

If you select fault prevention as the main strategy, you should demonstrate how you could adopt the ISO 9000-3 guidelines in your project. If you select fault tolerance, you should start your design to account for fault tolerance, i.e. redundant modules, etc.

Write a summary of your selected strategies and how you are going to implement them.

Deliverables:

- 1) The requirements document (textual, 1-2 pages) explaining what is to be built and what are the functions that the system offers.
- 2) Use-cases and use-case specification document.
- 3) Reliability requirements document containing answers to the above questions.
- 4) Submit one report per team.

Notes:

Some of the decisions may be revised or updated in the later phases of the project. If you are not sure of a decision (failure intensity for a component, etc.), document all possible candidates and leave the decision to be made later time.

Part 2: Developing Operational Profiles

Grade: %25 of the total labs mark.

This assignment is a group assignment.

Here we go one step ahead and define the modes and profile of operation for the developed software. You should answer the following questions and complete the following tasks.

1. What factors are likely to yield different operational modes for your system?

Construct a list of possible operational modes. Reduce the list to a set of operational modes that are significantly different from each other and frequently executing. Limiting the number of operational modes is important; otherwise, you may create excessive work for yourself later in the software reliability engineering process.

2. Pick one operational mode of the product of broad scope and identify its operation initiators.

The operational mode that you select may be usually the one executed the most in the field. And operation initiators may be among user types, external systems, own system.

3. Decide between tabular or graphical representation for the operational profile.

4. If you are using the tabular representation, list the operations for the operational mode you have selected.

Consider the different initiators to generate the list for each operational mode.

5. If you are using the graphical representation, draw the network.

For large network, depending on the size of operations and attributes, you may limit the size to not more than 10 operations or attributes or draw part of the network.

6. How will you determine occurrence rates and occurrence probabilities for the operational profile of the operational mode you have selected?

Is it possible to estimate them? Can you obtain any data from already available resources, log files, etc? Can you measure or collect data?

In either case represent the occurrence probabilities using simple numbers.

Deliverables:

- 1) The documented operational mode(s) and operational profile(s). This document provides answer to the above mentioned questions and completion of the above mentioned tasks.
- 2) Design documents (collaboration or sequence diagrams, class diagrams for the analysis and design phases)
- 3) Submit one report per team.

Part 3: Preparing for Test Workshop

Grade: %25 of the total labs mark.

This assignment is a group assignment.

At this stage you identify how many test cases you need; how much time you want to spend on testing the product; and finally define your test-cases and develop your test suite. You must perform the following tasks:

1) Estimate the number of test cases you need to prepare

If this is the first release, estimate the total number of test cases required. If this is not the first release, estimate the number of new test cases for this release. Your estimation can be based on time and/or cost criteria.

2) Allocate the number of test cases among the associated and sub-systems to be tested.

How many new test cases should be created if this is not the first release? How many test cases will go to the other infra-structure systems? How many to the OS? How many will be assigned to the acquired and developed products?

3) Allocate the number of test cases for the developed product to its operations

Make sure you identify critical operations properly and assign enough test cases to them. New test cases will naturally be assigned to new operations in the first and subsequent releases.

4) Specify the test profile for one of the operational modes

If you have more than one operational mode (e.g., prime time, peak, off time, etc.) select the one most frequently used and specify its test profile.

5) Document your test cases

You need to document them all for your own use. Include a few (5-10) test cases in your report.

6) Determine how you will divide hours of test among the associated systems you have defined

7) Determine the number of hours you will devote to feature, regression (if needed), and load test for the product

8) Allocate the hours of load test among the operational modes.

Deliverables:

- 1) The documented test profile(s). This document provides answer to the above mentioned questions and completion of the above mentioned tasks.
- 2) Submit one report per team.

Part 4: Release & Reporting Workshop

Grade: %25 of the total labs mark.

This assignment is a group assignment.

During this final step you run your program and test it and record its reliability growth. You must accomplish the following tasks:

- 1. Report how you are going to collect the test data (if the project was to be implemented)**
- 2. Report how you can verify the actual reliability growth of the project**
- 3. Determine how to achieve your target reliability and may/may not release your software**
- 4. Prepare a 8-10 minutes slide presentation of the project**

Deliverables:

- 1) Answer to the Q1- Q3 above.
- 2) Powerpoint or PDF file for the presentation.