SOEN 384
Management, Measurement and Quality Control

Tutorial 2 on Quality Control

SRS Review Exercise

SRS review exercise overview

- **Objective:**
  - Inspect SRS Document for Quality
    - Formal Technical Review (FTR) , forms were **posted on Moodle**
- **Steps:**

1. Find Defects/Errors individually this week **before the tutorial**, approx.. 1h
   - Record the defects in the provided Form 1
2. During tutorial 2, your TA will separate you in break rooms (2-3 people)
   - Verify the defects found by your teammate
   - Compare defect reports, eliminate those that are duplicated
   - Complete a team report
   - **Each student will submit individual reports, team report, and team metrics by the end of the week through Moodle**

Requirements Inspection Rules

(From Sylvie Trudel's slides)

- TESTABLE :
  - A requirement shall be testable by any means.
- ELEMENTARY :
  - Requirements shall be stated at the lowest possible level.
  - A requirement shall contain only one testable item. As a guideline, if two or more things have to be tested in order to validate the requirement, then it is not expressed at the elementary level.
- NEED :
  - A requirement shall be stated in term of final needs, not perceived means (state the « What », not the « How »).

Redaction guidelines for requirements

(From Sylvie Trudel's slides)

Karl Weigers, Software Requirements, Microsoft Press, 2000

- Write short sentences and paragraphs.
- Do not write passive sentences.
- Write complete sentences with correct grammar and punctuation.
- Use consistent terminology that is defined in the glossary.
- Always use the same sentence pattern : « The system shall » followed by an active verb and an observable result.
- Avoid vague expressions such as « user friendly », « easy », « simple », « fast », « efficient », « superior », « robust », etc.
- Avoid comparison verbs such as « optimize », « maximize », « minimize ». Instead, quantify the desired improvement level or specify a lower limit and an upper limit.
- Avoid design level details such as "press button"

Types of defects

C = Critical:
defect that is likely to cause rework, will prevent understanding or desired functionality

M = Minor:
information is **wrong**, but does not prevent understanding

S = Syntax or spelling

G = Guideline:
agreed guideline has not been followed

I = Improvement:
the product can stay as is but would be better with the improvement.

Q = Question:
clarification or explanation required, often leading to a defect (critical or minor)

Defect category to use

F = Functional defect

- Defect that may affect functionality
  - E.g. missing or unclear functional requirement

N = Non-functional defect

- Defect that is not likely to affect functionality
  - E.g. inconsistence in document format

U = Undetermined defect category

- Defect for which it is not obvious whether it has an impact or not on functionality
  - E.g. a syntax error in a functional requirement
- Why?
  - Should be categorized into F or N by consensus among participants

Form 1: inspection-Form1-individual.doc (download Tutorial 2 package)

- Filled up by each member of the team.
- Combine them all in one file before submission.
- Example:

2.UOBERSERVE DESCRIPTION

2.1uObserve perspective

uObserve is a tool that supports usability testing of user interfaces. It records audio and video data of a user running a spied application. uObserve synchronizes a log of events sent from the sniffer (records of keyboard and mouse movements). uObserve is composed of two (2) subsystems: the uSleuth server executing on a different workstation than the uSpy client. uSpy is a sniffer library that would be integrated to an existing Java client application to be observed. uSpy sends a log of keyboard and mouse events to uSleuth via a Local Area Network (LAN). uSleuth records and playback audio/video files.

At least one camera shall be used, as shown on Figure 1.

Defect #	Page	Line	Seve-rity[1]	Type[2]	Description

Form 2: inspection_Form2_team-leader.xlsx (download Tutorial 2 package)

- Filled up by the team leader to finalize the work.

CRIM

Inspection Form

Request		*Completed by the author*
Name of the author(s):		
Request date (yyyy-mm-dd):		
Identification of the product to inspect:		
Identification of "source" documents, drawings or other:		
Rules & checklists to use:		
Product size:	Quantity:	
	Type:	
Date & time required (yyyy-mm-dd; hh:mm) :		
Project name:		
Activity/Iteration/Life Cycle Phase:		Effort (to plan):
Charge code:		

Opening	*Completed by the inspection leader*			
Name of Inspection Leader:	Inspection ID:	# Participants	Kick-off Duration	Effort (kick off)
Mr. Abcd Efgy				

Form 2: inspection_Form2_team-leader.xlsx (download Tutorial 2 package)

- Filled up by the team leader to finalize the work.

Inspection										*Completed by inspectors*
Inspection Mode: Serial ☐ Parallel ☑		(C)	(M)	(S)	(I)	(?)			yyyy-mm-dd	15 min. = 0,25 hr
Inspectors' names	**Role(s)**	**Critical/Major**	**Minor**	**Syntaxe/ Spelling**	**Improvement**	**Question**		**Initials**	**Date**	**Effort**
Xyz Pqrs	Standard	4	8	2	6	6		XP	17/09/2008	55 min
Mnop Qrst	Standard	2	5	4	5	4		MQ	18/09/2008	49 min
Abcd Efgh	Standard	7	2	5	3	3		AE	17/09/2008	1 hr 10min
Logging Meeting	All								19/09/2008	x
Correction Verification	All								19/09/2008	x
Number of anomalies found:										
Correction										*Completed by the author(s)*
Number of anomalies NOT corrected:										
Fermeture										*Completed by the Inspection Leader*
Total number of corrected defects:										
Disposition: Accept as is ☐		Re-inspect ☐			Accept if modified ☐				Total effort:	
Comments or notes:										

Process for the Review and Revision of Requirements

- Inner consistency checking of all related documents
- Verifying against defined best practices (rules, checklists)
- One efficient way to provide feedback on the author's work (Inspections and peer reviews)
- Plan for a managed-change approach
  - Change traceability analysis and risk analysis

Review Process Metrics

Inspection Rate

- This metric can be used to get a rough idea of the required duration to perform a code review. The inspection rate is the number of pages reviewer can cover per unit of time.
- This rate should not be used as part of a measure of review quality, but simply to determine duration of the task.

Defect Detection Rate

- This metric measures the defects found per unit of time
- can be used to measure performance of the code review team, but not to be used as a quality measure.
- Defect detection rate would normally increase as the inspection rate (above) decreases.

Tutorial 2 on inspections: individual work

- **Goals and guidelines:**
  - Concentrate on Critical and Minor defects
  - note any Critical defect observed
- **Work that must be completed before the tutorial:**
  - Read the SRS document
  - Annotate all defects by Severity and Type
  - Fill in Potential defect list (Form 1)
  - Record inspection effort in minutes (Form 1)

Tutorial 2 on inspections: Team work

- **Teamwork:**
  - Review your team member's defect list,
  - Remove duplicate entries
  - Complete inspection form for the team
  - calculate **Inspection Rate** of your team
  - calculate **Defect Detection Rate** of your team
  - Add the metrics data to your Report
  - Submit through Moodle the individual reports (Form 1), Team report (Form 2), and review metrics results

Remarks

- If by any reason you cannot attend the tutorial live session, talk to your TA to find you a teammate for the team work