UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2007/2008**	Semester:	**Semester 1**
Module Title:	**Software Quality**	Module Code:	**CS4157**
Duration of Exam:	**2½ Hours**	Percent of Total Marks:	**80**
Lecturer(s):	**Dr. Ita Richardson** **Mr. Pat McElligott**	Paper marked out of :	**80**

Instructions to Candidates:

- Answer any 4 questions.
- All questions carry equal marks.
- Use separate exam script for Question 1.

Q1.

a) Define verification. (1)

b) Define validation. (1)

c) Explain why it is practically impossible to validate reliability specifications when these are expressed in terms of a very small number of failures over the lifetime of a system. (2)

d) Explain why ensuring system reliability is not a guarantee of system safety. (2)

e) Give two reasons why inspections are an effective technique for discovering errors in programs. What types of error are unlikely to be discovered through inspection? (6)

f) Compare top-down and bottom-up integration and testing, giving two advantages. Explain why most large system integration, in practice, has to use a mixture of top-down and bottom-up approaches. (8)

Q 2.

a) Software quality factors can be considered within the classifications of Product Operation, Product Revision and Product Transition. List five factors which should be taken into account when developing software, defining the classification to which they belong. Discuss each of these factors in detail. (10)

b) Within Total Quality Management (TQM) there are four basic quality concepts – fitness to standard, fitness to use, fitness of cost, fitness to latent requirements. Describe each of these and how they apply to software development. (10)

Q3.

a) Pick any software process with which you are familiar. Describe this process. (4)

b) Giving a real life example, discuss

How the correct implementation of this process can result in successful development of software

OR

How the incorrect implementation of this process can result in unsuccessful development of software (6)

c) List five factors that make software succeed, and discuss how each of these can be implemented through the use of a software quality model such as CMM, SPICE or ISO9000. (10)

Q4. Answer any 2 of (a), (b), (c)

a) When implementing Global Software Development, organisations need to be cognisant of barriers and complexities that are introduced, mainly because of distance. Discuss any five of these factors, and how they can be avoided through good global project management. (10)

b) Small companies are not large companies, and therefore, when implementing software process improvement activities, they cannot be in the same manner as large companies. In particular, models such as CMMI and ISO15504 were not developed with small companies in mind. Discuss how small companies can successfully implement software process improvement strategies. (10)

c) Due to Food and Drug Administration requirements, medical device companies face more stringent software process requirements than other industry sectors. Discuss why this is the case for medical device companies, and, in the case of risk management, what they can do to ensure that their software passes regulatory requirements. (10)

Q5.

Software Company Greekgeeks has asked you to analyse their process. They have provided you with the data from 4 of their projects, Alpha, Beta, Gamma, Delta, in the spreadsheet overleaf. The timeframe for these projects is:

Alpha – July 2005 to June 2006
Beta – Dec 2005 to Nov 2006
Gamma – Aug 2006 to July 2007
Delta – Oct 2006 – Sept 2007.

Using statistical charts analyse the processes in Greekgeeks. (12)
What questions does this pose for the company and its processes? (5)
What extra data would you advise should be collected? (3)

You have ascertained that:

Team members on Beta and Gamma were sent on a training course at the end of Q2.

SPI project was introduced into Alpha starting Q3.

Project leader on Delta was sick for Q1 and was eventually replaced by a team member at start of Q2. This team member was not replaced until end Q2.

Table 1	**ALPHA**	**BETA**	**GAMMA**	**DELTA**
No.Errors (Testing)	**No.Programs**	**No.Programs**	**No.Programs**	**No.Programs**
< 20	12	45	33	62
< 40	21	68	88	106
< 60	25	80	130	142
< 80	30	103	139	213
< 100	36	148	158	265
< 120	45	150	172	299

Table 2	**ALPHA**	**BETA**	**GAMMA**	**DELTA**
	No. Errors	**No. Errors**	**No. Errors**	**No. Errors**
Functions Missing	10	17	14	0
Calculations Incorrect	12	15	24	2
Interface Incorrect	14	12	24	3
Accepting Invalid Data	28	21	15	1
Files not Updated	17	25	11	0
Speed of System	2	5	8	31

Table 3	**ALPHA**	**BETA**	**GAMMA**	**DELTA**
No of Classes	330	258	143	187
Team Members	8	4	5	10

Table 4	**Q1**	**Q2**	**Q3**	**Q4**
ALPHA				
Actual Hours (Average per team member)	400	350	300	350
Planned Hours	450	420	400	350
Overtime (as % Actual)	30%	20%	10%	10%
BETA				
Actual Hours (Average per team member)	350	320	320	320
Planned Hours	380	380	370	370
Overtime (as % Actual)	0%	0%	10%	10%
GAMMA				
Actual Hours (Average per team member)	420	300	300	300
Planned Hours	400	400	400	400
Overtime (as % Actual)	30%	20%	20%	20%
DELTA				
Actual Hours (Average per team member)	368	360	375	380
Planned Hours	370	370	370	370
Overtime (as % Actual)	5%	0%	10%	10%

Actual - Time worked on project

Planned - What we planned for