

Software Engineering Midterm Examination (May 2010)

※ 請按題目順序依次作答，可用中、英文來答題

※ 本次考試採 open-book 方式，但禁止彼此交談討論或交換資料

1. (10%) Answer each of the following briefly.

(a) What is Software Engineering? ~ 23

(b) What is Software Reuse?

2. (10%) Why are incremental process models considered by many to be the best approach to software development in a modern context? $= 54$

3. (10%) What is Six Sigma? Can we use it to improve software quality? How? $1.45?$

4. (20%) Evaluate the following sentences by answering either "True" or "False".

① Function Point (FP) can be used to size software applications. Sizing is an important component in determining productivity, predicting effort, estimating cost, etc.

② For a software module, measuring faults per KLOC is an example of direct measurement and measuring the number of faults in that module is indirect measurement.

③ In the software cost models, effort is usually plotted against time for a series of projects and in the model cost is calculated as a function of project duration.

④ Productivity is usually defined in term of program size divided by effort. Experimental results show that there is a linear relationship between effort and size, leading to a constant productivity.

⑤ SQA is a style of management aiming at achieving "long-term" success by linking quality with customer satisfaction.

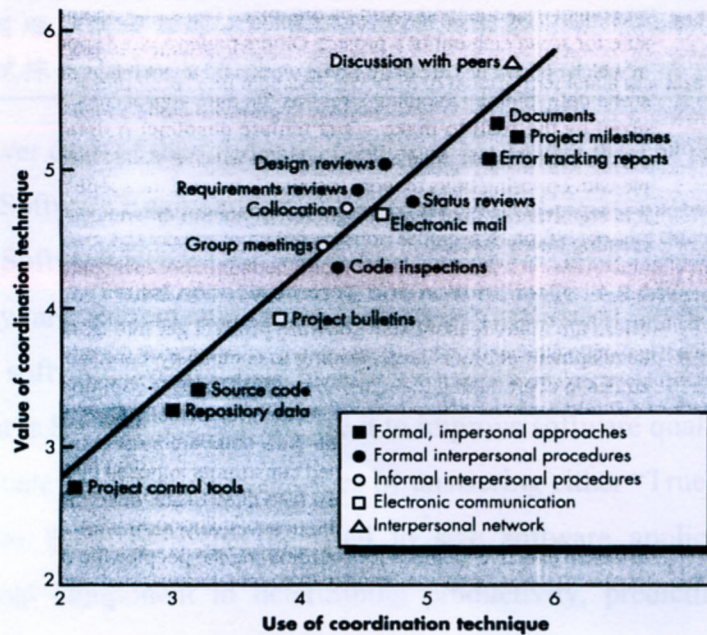
5. (10%) Defect Analysis: The following table depicts the pre-release defect rate vs. pre-release inspection effort in typical software systems.

		Pre-release inspection effort	
		Higher	Lower
Pre-release defect rate	Higher		
	Lower		

(a) Considering the quality of the code, explain which one of the 4 combinations is the best and which one is the worst case and why.

(b) Fenton and Ohlsson reported that the hypothesis of "Modules with higher defect rate in pre-release testing likely to have higher failures in post-release operation" is invalid. Do you agree with him? Explain why.

6. (10%) Below is the plot of value and use of coordination and communication techniques. Please explain it.



7. (15%) Answer to the following questions:

- (8%) DeMarco states that "You Cannot Control What You Cannot Measure". What does this mean for software project managers?
- (7%) If we define the productivity as program size divided by effort, what metrics do you suggest to measure productivity?

8. (15%) Several measures have been defined for software length measurement, including the line of code (LOC), Halstead's measure of size, and function point (FP). Please compare these 3 measures against the criteria by writing "yes" or "no" or "both" in the table below.

Criteria	LOC	Halstead's	FP
Applicable to early stages (requirement phase) of software development?			
Subjective measure?			
Derivable from system requirements?			
Language dependable?			
Has clear and understandable physical meaning?			

9. (10%) Software is often a part of various devices and systems where malfunction can have a very dramatic effect on people or the entire society—consider for example banking systems, power distribution or hospital equipment. (What is (or, what should be) the responsibility of software providers (organizations or individual software engineers) for their work?) (In some countries software engineering is a certified profession in a similar way as doctors: what are the advantages and disadvantages of this solution?) (Do you think that it solves the problem above: why/why not?)