UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING
FINAL EXAMINATION, December, 2008
CSC444H1 F – Software Engineering I
Examiner – D. Penny
2½ Hours - 60 marks total
Write all answers in the exam books provided.
No aids allowed other than non-programmable calculators

Answer all questions succinctly, but including all relevant points. Think deeply about your answers, demonstrating your insight and understanding of software engineering with each one, and make convincing arguments to back them up. If you make a statement, for example "a certain practice is inefficient," then be sure to explain why.

- The points below are adequate to get the full marks
- If alternative good points are present, they can be rewarded marks in lieu of the listed points.

1. (10) What are the potential problems with using Gantt charts to plan the next release of a software product?
    - Gantt charts generally contain
        - Implementation tasks (1)
        - Who is assigned to what task (1)
        - The pre-requisite information between tasks (1)
    - This is too much detailed information that is (2)
        - Not required at an early stage (1)
        - Takes too long to develop and is too difficult to change when planning (1)
        - Cannot be manipulated by a non-technical person (e.g., product manager). (1)
    - For planning,
        - We do not need to know who exactly is working on what task – coders are to a great extent interchangeable (1)
        - We do not need to work out pre-requisites during the planning stage as generally pre-requisites can be worked around in software (e.g., create an API to break a dependency) (1)
2. (10) Why is using only a single number for an "estimate" (e.g., estimating a work factor is 0.6) not terribly meaningful? What is the most accurate way of providing an estimate? What is generally considered an acceptable compromise?
    - A single number can contain no notion of the degree of uncertainty around that number (4)
    - The most "accurate" is for a PDF function to be drawn out for us. (3)
    - We generally consider two numbers that define a Normal or Log Normal curve that best approximates the PDF that would have been drawn to be an acceptable compromise (3)

3. (10) Why do we recommend that a source code control system be used as a buffer between the coding group and the build/QA group?
    - This is to guarantee that nothing gets shipped out the door that has not been built and tested by QA (3)
    - It also guarantees that the source for what gets shipped is present in the source code control system. (3)
    - The above is vital in order to be able to debug an issue found in the field. (2)
    - Also vital in order that a patch to what is in the field can be produced. (2)
4. (10) What is UML used for in software engineering?
    - UML has two main uses – one for requirements and one for design (3)
    - For requirements, UML is used to name domain concepts and the relationship between these concepts. (2)
    - The UML from requirements is the starting point for database schemas, UI designs, and class design (3)
    - For design, UML describes the class structure of an OO program. (2)
5. (10) Why do we use defect arrival rate as a shipping threshold?
    - This rate is assumed to be proportional to the total number of defects remaining in the product. (5)
    - The threshold should be established based upon previous experience with the perceived quality of the release by customers at equivalent arrival rates. (3)
    - Setting an arrival rate threshold is the same as setting a threshold on the quality as perceived by customers. (2)
6. (10) In the feature workflow presented in class, there were three states called ‘Valid’, ‘Valid Ready’, and ‘Valid Verified’. Why were they there? How did they work? Should they always be present in a feature workflow? How are they used by a manager to affect behavior?
    - Those states were there because the particular organization using it was having a problem getting well defined feature requests into the release planning process, thus delaying the start of release planning. (3)
    - It need not always be present, only when the organization is having a problem as described above (2)
    - A defect was in Valid state when it was accepted as a reasonable idea that was not a duplicate by the product manager (1)
    - It was moved to Valid Ready by the product manager to signify to QA that the feature was filled out with all the necessary criteria of a well-defined feature as defined in the process document (1)
    - It was moved to Valid Verified by QA after checking that the feature was well-defined (1).
    - Managers will create reports showing the number of defects in each of the stages. If the numbers are too high in a particular stage, the manager can go to the group responsible and ask for them to improve, and then monitor whether or not they have improved (2)