Name: Instructor – Marking Scheme Student #: N/A

_____/50

CSC444F'05 Midterm Test

50 minutes – No Aids Allowed – 50 points total

Answer all questions in the spaces provided. Use the backs if you run out of space.
Write your name and student number on each sheet.

_____/3 1. What is "maintenance" as it applies to software? Explain using an example why maintenance is economically important.

Maintenance is fixing defects (1) in, and adding features (1) to, previously released software.

The initial release may take 1 year and 5 people. Maintenance on a successful product may go on for 10 years and involve 50 people. (1)

There was a problem with this question. I never really defined "maintenance" this way in class or in the book (!), though it is the correct academic definition. Therefore, I was very generous with this question and would accept:

(1) Maintenance is fixing defects
(2) It is important a company fixes problems in the software they ship or customers won't buy it from you.

_____/3 2. What are the three important questions release planning is concerned with? What is more important than any of the three questions?

1) What features will we add to the next release?
(2) 2) How many trained coders to we have to work on coding these new features?
3) When does the release have to be shipped?

(1) Can we satisfy all 3 simultaneously?

_____/3 3. When coming up with an estimate in Effective Coder Days for a feature, what three underlying contributing factors are being estimated?

(1) 1) What is the inherent size and complexity of the feature (e.g., in LOC or FP).
(1) 2) Which coder will be assigned to work on the feature?
(1) 3) How productive will that coder be on that feature?

_____/9

_____/14 4. Describe the phases in a release lifecycle. Identify the start and stop points of each phase.

(1) Requirements Gather: lists of desirable new features are centralized, cleaned up, and the most likely ones are sized.

– Start & stop: goes on continuously

(2) Specification and Design: Specifications defining the externally-visible behavior of the new feature are written where needed. Designs describing how the new feature will be coded into the software are written where needed.

(1) – Start: after a release plan is produced

– Stop: these activities continue into the next phase

(2) Code & Unit test: Coders write the code to implement the new features and themselves test that the feature works as they expect.

(2) – Start: at "fork", the point at which the maintenance codeline for the previous release is branched from the main codeline.

(2) – Stop: at "dcut", "development cutoff", the point at which all coders can think of no further code that needs to be added to complete all the features.

(1) Alpha Test: Testers execute test plans against the code. Coders fix defects identified by the testers.

– Start: dcut

(1) – Stop: when the software is sufficiently stable to be shared outside the development team.

(1) Beta Test: The software is given to friendly customers and other employees in the company for preliminary non-production use.

– Start: end of Alpha (see above)

(1) – Stop: at "GA", "General Availability", the point at which the defect arrival thresholds are sufficiently low that the software can be shipped. After this point, customers buying the software will get this new release.

_____/4 5. Before dcut, what management actions can be taken if the plan is falling behind? What management options are left after dcut?

Before dcut:

(1) The ship date can be delayed.

(1) Features can be cut or reduced in scope

(1) Developer capacity can be increased by adding coders (not usually feasible owing to lack of experience on code base), by increasing w (by having coders work longer hours), or by decreasing v (forgoing vacation).

After dcut:

(1) The ship date can be delayed (The only other alternative is to ship with poor quality owing to a compressed test phase).

_____/18

_____/4 6. What is release proliferation? What is the problem with it and why? How does one combat it?

(1) Release proliferation is having too many simultaneously active maintenance streams out in the field.

(1) Because each defect must be reproduced and fixed in each active maintenance stream it eats into coders' time which takes them away from new feature coding on the next release.

(2) One combats it by minimizing the number of active maintenance streams by encouraging customers to move to the latest release (by making the release compelling, by marketing it effectively, and by contractually refusing to support older releases). Also, one must avoid polluting point releases with new features or else customers will be reluctant to move on to new point releases and hence existing point releases may become maintenance stream of their own. (any 2 good points of the 4 given in this paragraph)

_____/4 7. (4 points) Give your four most important reasons for having source control.

Any of the following (1 point each up to 4)

- **Repository: a known place to safely keep sources.**
- **Structure: one common structure defining the module architecture.**
- **History: history of all changes. Useful to track down problems and understand code.**
- **Control: management has visibility into and can control code changes.**
- **Collaboration: multiple coders can work at the same time on the same application, staying isolated from change or integrating it in as they choose.**
- **Multiple Streams: can support multiple codelines for maintenance and shipping reasons.**
- **Reproducible System State: can reproduce any shipped build in order to reproduce defects found in the field and/or create patches.**
- **Code/Build Communications: acts as the hand-off between coding and build/qa to ensure that proper standards are applied to all shipped software.**

Will also accept anything else that looks reasonable.

Does not require the exact underlined wording.

Will not give mark if one point is just another way of saying a previous point.

_____/3 8. (3 points) What is the shipping codeline used for? Why do we need it?

(1) Branched from the maintenance codeline a few days prior to shipping. Only QA/Build may integrate changes into this codeline.

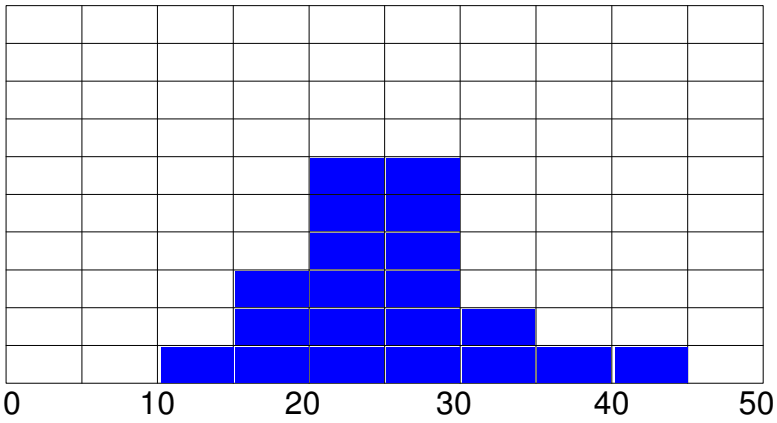
(1) It is used as an extra safeguard to ensure that only show-stopper, management-selected defects are corrected in code about to be shipped.

(1) Without it, we risk introducing a fix for a non-show-stopper defect that introduces a show-stopper that we either need to fix and test completely and delay the release, or that risks escaping detection due to inadequate time remaining to test.

Name: Instructor – Marking Scheme Student #: N/A

_____/7 9.a) Draw a probability density function for a feature sizing estimate consistent with the following statements:

- On average, it will take about 25 effective person days. **(1)**
- Plus or minus 5 days about 60% of the time. **(1)**
- There is about a 10% chance it'll take longer than 35 days. **(1)**
- There is no chance at all it'll take longer than 45 days. **(1)**
- 20% of the time it'll take longer than 30 days. **(1)**
- If everything goes well, it might come in as early as 15 days – there's only a 5% chance of that, though. **(1)**
- Certainly no earlier than 10 days no matter what. **(1)**



(-1) if does not add up to total cumulative probability of 1.
OK if draws a curve and marks probabilities in various areas of it.

_____/2 9. b) Give two reasons why the above feature sizing is inconsistent with a Normal distribution?

(1) It does not extend to ±∞ with non-zero probability.
(1) It is not symmetric about the mean.

_____/3 9.c) If you must model the above as a Normal distribution, what would you choose as the mean and standard deviation?

(1) Mean: 25
(1) Sdev: mean ± σ should encompass about 68% of the probability. Therefore slightly larger than 5 (which encompasses 60%). Around 6 or so.

_____/12