

Assignment 1

SE Introduction & Software Processes

Total – 50 pts.

1. (3 pts.) Give an example of a project that would require software engineering principles rather than just the knowledge of Computer Science. Explain why?

Response: Creating any product for a company would require at least a passing understanding of software engineering principles, these principles would allow a programmer to be organized in creating a lasting, structured, well-documented, and maintainable project.

2. (3 pts.) For generic software that is sold to a wide market, it is estimated that the system testing costs is more than 50% of the total development costs. Explain? Why is this not the case for custom software?

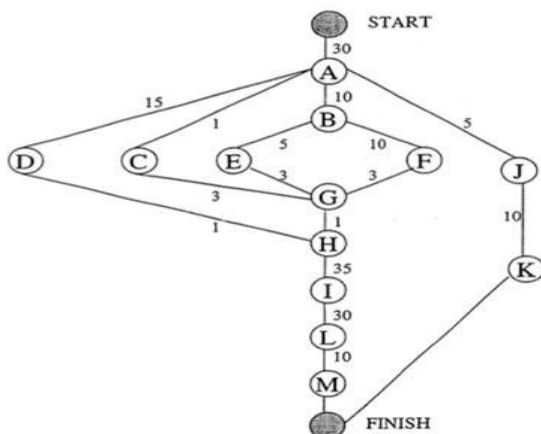
Response: For generic software that is sold to a wide market, system testing needs to attempt to cover every single possible situation in which a user could find themselves, with a variety of system builds in an astoundingly diverse number of environments. For custom software, the testing costs are greatly reduced as the amount of testing needs to be done is limited only to the specific hardware that a given company will be using, in the given environment.

3. (3 pts.) List at least three ACM/IEEE code of ethics (complete sentences, please).

Response:

- 1) Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
- 2) Software engineers shall be fair to and supportive of their colleagues.
- 3) Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

4. (5 pts.) Consider the following activity diagram. The arrows (which are missing) go downward from top to bottom.



Provide the following for this diagram:

- a) the critical path
- b) the minimum amount of time it will take to complete the project
- c) the real time for activity K
- d) the slack time for activity L
- e) the predecessors of activity G

Response:

- a) A, B, F, G, H, I, L, M
- b) 129
- c) 45
- d) 0
- e) A, B, C, E, F

5. (4 pts.) Pick three out of eleven McCall's quality factors and briefly explain what they are. Pick the three that you think are most appropriate for your class project and discuss why they might be important to the project and to your client.

Response:

- 1) Integrity – protecting the product from unauthorized access. This is the most important for our project as we are creating an application that allows personal federally protected information to be accessed by the student.
- 2) Portability – the ability to transfer the software from one environment to another. As this project is a mobile application to be written only for one mobile platform, the design needs to be portable enough to be transferred to other mobile platforms as needed.
- 3) Flexibility – the ability to make changes required as dictated by the business. As the Honor's College, and John in particular, is prone to adding new plans and coming up with new ideas, the project needs to be able to adapt to the College's changing needs.

6. (4 pts.) Pick two out of the top ten risk factors from chapter 8 that may affect your class project.

- a. Explain briefly why you believe that these are risks to your project.
- b. If you were asked to calculate the risk exposure for these risks what measurement would you use to represent the effects of these risks.
- c. Calculate the risk exposure for these risks. Make appropriate assumptions.

7. (4 pts.) Mention three different approaches to modify the waterfall process model to address its weaknesses. How do these variations help the model?

Response:

- 1) Prototyping – in the waterfall model, the customer is only communicated with for requirements and not again until the product is complete; with prototypes immediate and accurate feedback is gained.
- 2) V-model – testing is concerned from the beginning; while gathering requirements, creating designs, and beginning implementation of code developers are also considering how everything should be tested and test what they can when they can.

3) Feedback loops – going back to previous phases allows developers to make changes if something is not going the way it should be.

8. (6 pts.) Differentiate between incremental and iterative (evolutionary) process models. Considering your semester project, discuss:

- a. at least three different increments you will be developing, if you were using the incremental process model
- b. at least three different iterations you will be developing, if you were using the iterative process model.

Reponse:

Incremental process models break the project down into pieces or phases and all pieces come together in the end to form the final product, while iterative process models start with a base prototype and changes are made as feedback is given.

Incremental:

- 1) Student side allows for 'check-in' for attended meetings
- 2) Student side allows for auditing Honor's requirements
- 3) Admin side allows for staff to adjust requirements

Iterative:

- 1) UI of user and admin side
- 2) Admin side can see 'check-in' and student-side can 'check-in' to meetings
- 3) Admin side can adjust requirements and student-side can audit requirements

9. (5 pts.) Which process model is appropriate for each of the following software development projects. Explain briefly.

- a. Loan processing module for a bank's customer relationship management system.
- b. Patient registration system for a hospital's emergency room.
- c. An embedded system in a children's hand-held game.
- d. An iPhone application that will control the thermostat in your home.

Response:

- a) evolutionary model would allow for constant feedback to exist between developers and stakeholders, best choice for the different types of users that would be using it
- b) if price is no object, spiral model is the model best used for minimizing risks; in a critical system such as a hospital ER, risks need to be near eliminated
- c) iterative model, developers could just build a basic game system, making adjustments and advancing over time as needed
- c) incremental model would allow for first an iPhone app that makes a connection to the house, then the controlling of the thermostat would come as a different phase

10. (2 pts.) If you were allowed to create a hybrid model, which two or more models would you combine and why? Pick any one system from question 9 to make an argument.

11. (2 pts.) Explain why programs that are developed using evolutionary development are likely to be

difficult to maintain?

Response: With evolutionary development you are constantly building on an already existing code base, as developers are trying to get the next update out they aren't necessarily trying to write the best code; this can result in a subpar final product because developers are trying to fit in new code with old code without having to completely start over.

12. (2 pts.) What are SE methods? Provide an example.

Response: Software Engineering methods are the methodologies used to divide up a project with the intent of better planning and management; the waterfall model is an example.

13. (2 pts.) What are SE tools? Provide an example.

Response: Software Engineering tools are any tool used during the software development process; version control systems such as GIT or an IDE such as Eclipse are both examples.

14. (5 pts.) **Scrum** is an agile methodology. Research this topic and provide a detailed description of this methodology. Make sure you address the following: how it is unique and different from the other software development methods, its principles/characteristics, its strengths and weaknesses.

Response: Scrum focuses on real time change when planning projects; similar to the incremental process model, Scrum breaks the project down into pieces, called sprints, which range from one to three weeks in length. At the end of each sprint, the team members and stakeholders meet to discuss progress and plan the next step; the project's direction is adjusted based on progress not predictions.

Strengths include: earlier development start time, continuous feedback between stakeholders and team members, and changes are very easy to cope with as they arise.

Weaknesses include: product owner has most authority and must not micro-manage, projects must be able to be broken down into digestible parts (sprints), and because the team is so small, large projects may be difficult to manage.