**Final Exam – Software Engineering Management**

1. Explain what is wrong with the notion that computer software does not need to evolve over time.

Answer (Section1.1):

Computer software must be revised as errors are discovered and corrected. Software must be updated to accommodate changes in the computing environment. Many times a customer will request changes to add new functions to an existing product or to accommodate changes in the business environment. Sometimes an older system will need to be reengineered to provide benefits to the user in a modern context. The bottom line is that software that does not evolve will eventually become unusable.

1. Describe the relationships among software engineering process, methods, and tools.

Answer (Section 2.1):

Software process defines the framework that must be established for effective delivery of software engineering technology, by providing a context by which the software engineering methods are applied. Software engineering methods provide the technical how-to’s for building software. Software engineering tools provide automated or semi-automated support for software engineering process and methods.

1. Why it important for software processes to be agile?

Answer (Chapter 3 Overview):

Software process provides the stability, control, and organization to an activity to prevent it from becoming chaotic. Yet, modern software processes must only demand the activities, controls, and work products that are appropriate for the team and product to be produced – to ensure that it can accommodate changes easily and deliver a high quality software product.

1. Why are evolutionary models considered by many to be the best approach to software development in a modern context?

Answer (Section 4.1.3):

Because time lines for the development of modern software are getting shorter and shorter, customers are becoming more diverse (making the understanding of requirements even harder), and changes to requirements are becoming even more common (before delivery), we need a way to provide incremental or evolutionary delivery. The evolutionary process accommodates uncertainty better than most process

1. Describe the three key assumptions regarding software projects that every agile software process must address.

Answer (Section 5.3):

It is difficult to predict in advance which software requirements and customer priorities will change and which will not.

For many types of software design and construction must be interleaved, it is difficult to predict how much design is needed before construction can be used to prove the design.

Analysis, design, construction, and testing are not always predictable processes and this makes planning difficult.

1. List environment characteristics that can be considered toxic to software teams.

Answer (Section 6.3):

Frenzied work atmosphere

High frustration that causes friction among team members

Fragmented or poorly coordinated software process

Unclear definition of roles on the software team

Continuous and repeated exposure to failure

Differing and incompatible team member human traits

1. Discuss how poor management decisions can impact software quality?

Answer (Section 19.3.6):

* Estimation decisions – irrational delivery date estimates cause teams to take short-cuts that can lead to reduced product quality
* Scheduling decisions – failing to pay attention to task dependencies when creating the project schedule may force the project team to test modules without their subcomponents and quality may suffer
* Risk-oriented decisions – reacting to each crisis as it arises rather than building in mechanisms to monitor risks and having established contingency plans may result in products having reduced quality

1. What is a formal technical review and why is one conducted? Outline the steps required to conduct a successful FTR?

Answer (Section 20.6):

The purpose of an FTR is to have a group of software engineers examine a discrete work product and determine whether on not the product is free of defects using the software specifications and standards as the review criteria.

To perform a successful FTR, the steps described in Section 15.6.3 are conducted.

1. Describe statistical quality assurance?

Answer (Section 21.6):

* Information about
* software defects is collected and categorized
* Each defect is traced back to its cause
* Using the Pareto principle (80% of the defects can be traced to 20% of the causes) isolate the "vital few" defect causes
* Move to correct the problems that caused the defects in the “vital few”

1. Describe steps needs to create a threat model.

Answer (Section 27.5):

* Identify assets
* Create architectural overview
* Decompose application
* Identify threats
* Document threats
* Rate threats

1. Describe the process of writing a formal specification for some system function.

Answer (Section 28.6):

* Define a system state in terms of the objects manipulated by the function
* Using a simple mathematical function as an example, define the data invariant by writing data relations that will not change during the execution of the function
* Write the precondition and postcondition for the function using mathematical notation to show the system state before and after the

1. Describe the change control process for a modern software development project.

Answer (Section 29.3.3):

A change request is submitted for evaluation for a change report is submitted to the change control authority (CCA). The CCA makes the final determination as to the status and priority of the change. An engineering change order (ECO) is generated for each approved change. Items to be changes are checked out of the project database subject to its access control parameters. The modified object is subjected to SQA procedures and returned to the project database. Version control procedures are followed to produce the next version of the software. Synchronization control is used to make sure that parallel changes made by different people do not overwrite one another.

1. How does software configuration management differ for Web and Mobile Apps?

Answer (Section 29.4)

The “code and go” philosophy dominates App development. So SCM for Apps must be an agile process. Documentation and review of changes is done on an as needed basis depending on the risk associated with the work products being changed.

1. Describe the five activities associated with the software measurement process.

Answer (Section 30.1.3):

Formulation - derivation of metrics and measures that are appropriate for the software representation being considered

Collection - mechanism used to gather the data used to derive the metrics

Analysis - metric computation and use of mathematical tools

Interpretation - evaluation of metrics results to gain insight into the quality of the software representation

Feedback - recommendations derived from the interpretation are transmitted to the software team

1. Describe the role of class-oriented metrics in assessing the quality of an OO system.

Answer (Section 30.3.3):

The class is the fundamental unit of an OO system. The number of methods and their complexity are directly related to the effort required to test a class. The depth of the inheritance tree can be used to estimate the complexity of the class hierarchy. It is important to strive to have low coupling between classes and high cohesion within each class. If large class sizes are detected during OOA review, the classes should partitioned to improve the modularity of the system and make it easier to maintain.

1. How is software scope defined?

Answer (Section 31.3.1)

By defining how the software to be built fits into a larger systems, product, or business context and the constraints imposed by the context. Determining what visible objects the customer expects to see as output and what input objects are required to produce them. Determining the software function needed transform input to output and any special performance characteristics. Attempting to bound all information quantitatively, when possible, and descriptively, when numbers cannot be used.

1. Explain how size-oriented metrics differ from function-oriented metrics. Discuss the pros and cons of each.

Answer (Section 32.2.3)

Size-oriented measures are computed by normalizing direct measures of the software engineering process (e.g. effort or defects) over the product size, measured in lines of code. Function-oriented measures are indirect measures that are computed from measures of the information domain of a business application and an assessment of its complexity. Size-oriented metrics are relatively easy to collect, but can present problems when component-based or visual programming methods are applied. Function-oriented metrics can be determined much earlier in the software cycle, but are an abstraction that is open to interpretation.

1. Why is the "make-buy" decision and deciding whether or not to outsource software development an important part of the software planning process?

Answer (Section 33.10):

It maybe more cost effective to acquire a piece of software, rather than develop it. Similarly, deciding to outsource software development frees resources for other purposes (or reduces expenses) but it makes outsourcing can make it harder to control and manage delivery times and development costs.

1. Approximately what percent of the project time line should be devoted to each of the activities listed below?

Planning \_\_\_\_\_\_

Analysis \_\_\_\_\_\_

Design \_\_\_\_\_\_

Coding \_\_\_\_\_\_

Testing \_\_\_\_\_\_

If you could expend more time in one activity, which would have the highest likelihood of improving software quality? Why?

Answer (Section 34.2):

Planning 2-3%

Analysis 10-25%

Design 20-25%

Coding 15-20%

Testing 30-40%

Expend more time in design because design is the place where quality is built into a product.

1. Describe all activities that must occur in order to produce a Risk Mitigation, Monitoring, and Management Plan.

Answer (Chapter 35.7):

Risk Identification - determine the risks that are appropriate

Risk Projection - determine the likelihood that each risk will occur and the damage likely to occur

Risk Mitigation - figuring out strategies to avoid the risks

Risk Management and Contingency Planning - assuming each risk becomes a reality determine ways to limit their impact

1. How does business process reengineering (BPR) differ from software reengineering?

Answer (Section 36.4):

BPR defines business goals, identifies and evaluates existing business processes, and creates revised business processes that better meet current goals. The software reengineering process activities have as their intent the creation of new versions of existing programs that exhibit higher quality and higher maintainability.

1. What characteristics need to be exhibited by organization to improve its software process?

Answer (Section 37.1):

Management commitment and support for SPI, staff involvement throughout SPI process, process integration into organization culture, customizable SPI strategy, solid management of the SPI project