**SE 361 – Practice Exam Jenny Zhen; 09.23.12**

1. What is the definition of software engineering?

* Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software.
* The delivery of high quality software in a timely manner at effective cost.
* **The on-time delivery of high quality software that meets the customer’s budget and needs.**

1. For each user story, indicate the process model that would best fit.
2. A local softball league needs a web-enabled application for tracking standings and statistics. You and your IT friend volunteer to take on the project, since you already play on a team on the league and have good experience with web applications.

* **Code-n-fix** (hacking your way at it; cowboy coding)

1. Boeing has received a government contract for a new line of fighter jets with brand new targeting software. The contract is for five years and there are rigid standards for safety, reliability, and system documentation.

* **Waterfall**, Incremental

1. Biotech Software services the genetic engineering world. Their proposed product will accelerate the pace of gene typing. Their work is R&D oriented using a new software architecture, but the goal is to produce a product within the next year.

* Spiral Model (iterative; allows you to handle changes)  
  **Evolutionary Prototyping (allows for more research)**

1. What is domain analysis, and why is it important?

* Looking at where your software is going to be used
* Looking at other software projects in the domain
* Compare requirements needed
* **The identification, analysis, and specification of common requirements from a specific application domain (and explain).**
* Why? Saves time; customer doesn’t need to clarify; enhances the quality.
* **Because it is hard to develop software if you don’t understand the business.**

1. What is requirements elicitation?

* **The process of getting useful requirements out of the client.**
* **Communicating with the client.**
* **Process of identifying the needs and constraints of the system.**
* Steps:

1. Identify the problem
2. Propose elements of the solution
3. Negotiate different approaches
4. Specify preliminary requirements for the solution

* Always a negotiation.

1. Identify four characteristics of good requirements.

* Precise, attainable, testable, reflects the customer’s needs, complete (comprehensive), unambiguous (specific about a certain topic)
* **Have benefits that outweigh the costs of development.**
* **Important for the solution.**
* **Don’t overly constrain the design of the system.**

1. For each requirement statement below, specify which type it is (functional or non-functional).
2. The system will be complete and tested in time for the 2012 Christmas season.
3. The automated trading system shall provide a public application programming interface (API) for the addition of new financial instruments.
4. The system shall continuously display a list of the ten most actively traded stocks on each of the major stock exchanges.
5. The new system shall be released before the hardware on which the present system is based on becomes unavailable.
6. The system shall be developed in C# on the .NET platform.

* **Nonfunctional, functional, functional, nonfunctional, nonfunctional.**
* **Functional requirements require an additional constraint on the system; things that the system can do.**
* **Do I have to write more code to meet this requirement? Yes? Nonfunctional.**

1. Rewrite the following requirement so that it reflects the characteristics of good requirements: “I don’t want there to be any downtime.”

* Total system downtime will not exceed 4 hours in a month (quantify downtime; put a number to it).
* **The system should maintain 99.9% (“3 nines”) of maintainability;** 99.9999 is “6 nines.”
* This is nonfunctional; expecting the system to not be down.

1. Describe the advantages and disadvantages of evolutionary and throwaway prototypes.

**Advantages:**

* The customer knows what to expect
* Test the feasibility of an idea
* Proof of concept
* Code quality (theoretically, doesn’t matter)
* Take more risks (in hopes of more gain)

**Disadvantages (throwaway):**

* Customer can think that the product is ready way before it actually is.
* Customers don’t like things being thrown away. They get angry.

**Disadvantages (evolutionary):**

* May not be at production level quality.
* Customers may want more features.

1. What is coupling? What is cohesion? In software, we want to **minimize coupling** and   
   **maximize cohesion.**

* **Coupling – the relative connectedness of components (modules or classes) in a system**
* **Cohesion – how well a single unit/component (class, package) performs its job; the relative functional strength of the module.**

1. Anne has written a system that receives a user’s age to determine if they can drink. The valid age for drinking needs to be in between 21 and 85 years. Give that system description.
2. Identify the input equivalence classes to be tested.

* **Under 21, 21 to 85, 86 and above.**

1. Identify at least 4 bounds that would need to be tested to ensure the proper operation of the software.

* **20, 21, 85, and 86.**
* **Any negative value; zero; very large integer.**

1. Compare and contrast black and white box testing.  
   **Black box:**

* Customer’s perspective.
* Don’t know the internals.
* Can’t look inside.

**White box:**

* Developer’s perspective.
* You know everything about the system.

**Same:**

* They are both for testing; verifying the system is working correctly.
* Have access to the requirements.

**Verification:**

* Did I build it right?

**Validation:**

* Did I build the right thing?

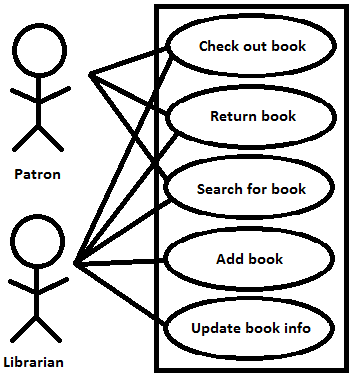
1. Complete the illustration of the V-model.

* **Requirements Modeling -> Architectural Design -> Component Design -> Code Generation   
  \_\_Executable Software\_\_  
  Unit Testing -> Integration Testing -> System Testing -> Acceptance Testing**
* **The V-model is describing how things are; how you build software.**

1. You have been hired to write an application to keep track of patrons that borrow books from a library. The system should keep track of the library’s inventory of books, including information about each book. The following assumptions can be made about the system.
2. Identify the stakeholder for the library system.

* **Library owner, developers, librarians, patrons, developer’s company.**

1. Draw a UML (**Unified Modeling Language**) Use Case diagram for the library system

* **Don’t put a database as an actor.**
* **Arrows are typically treated as bidirectional.** 

1. Draw a UML Class Diagram for the library system

* The arrows represent associations.
  + “One library knows about many customers.”
  + “Zero to one customer knows one book.”
  + “One library knows about many books.”
* The asterisk represents “many.”
* The filled diamond represents “composition.”
  + “The library is composed of many bricks.”
* The unfilled diamond represents “aggregation,” which is different from composition since the objects may still exist if the owner doesn’t exist anymore. A weak link.
  + “The library knows about a lot of books.”
* A customer can go to many libraries, but this UML diagram is for one library.

