**SE 361 – Practice Exam #3 Jenny Zhen; 11.11.12**

1. For each, indicate the best team member to handle the task.  
   (Project Lead, Quality Lead, Configuration Lead, Testing Lead, and Development Lead)
   1. Making sure the team adheres to good software process. **Project Lead**
   2. Documenting key design decisions (and the rationale behind them). **Development Lead**
   3. Maintaining a Defect Report. **Quality Lead**
   4. Tracking the progress of the project. **Project Lead**
   5. Verifying that the team follows coding standards. **Quality Lead**
   6. Writing and maintaining a build script for the code deliverables. **Configuration Lead**
   7. Assigns defects from Defect Report to developers. **Testing Lead**
2. For each requirement, specify whether it is a functional or non-functional requirement.
   1. The system shall allow the user to print any drawing which it can create.  
      **Functional – feature**
   2. The system shall support 9001 concurrent users without significant user experience degradation. **Non-functional – performance**
   3. The system shall interface with any number of the model MSP430X processor boards.  
      **Functional – external hardware interface requirement**
   4. The system shall allow a user to specify a logic file as a command line parameter.   
      **Functional – feature**
   5. The system shall support the use of a screen reader.  
      **Non-functional – software quality assurance**
3. Examine the following statement, “The system can never crash.”
   1. State how this statement fails to meet at least two properties of good requirements.
      * **Not well-defined, not testable, not clear, ambiguous (what is a crash?), not realistic.**
      * **Needs to be specific, measurable, attainable, relevant, and timely.**
   2. Rewrite the requirement to make a good requirement.
      * The system will provide 5.9s of availability (99.999%)
4. What is the term for unexpected behavior exhibited by a system when observed by the user?

* **Defect/fault – problem found by end users after the software is released**
* Bug – problem found during unit testing by the developer
* Error – problem found by the QA/Testers before software release

1. Which of the following are indications of a system that is loosely coupled?
   1. Difficult to see at a glance how some components work.
   2. Harder to reuse components in other systems.
   3. Changes in one component will typically require changes in other components.
   4. **None of the above.**
2. Which of the following are indications of a system that has low cohesion?
   1. **Individual components are responsible for a variety of unrelated services.**
   2. Computation performed by a component has no side effects.
   3. The classes use inheritance.
   4. The component is likely to be reused.
3. Which of the following are characteristics of glass/white box testing?
   1. **Designing the tests based on the software requirements document.**
   2. **Using knowledge of the code to identify boundary conditions that must be tested.**
   3. **Developing a testing strategy to ensure proper coverage of the code in the module.**
   4. **Observing at run-time the steps taken by algorithms and their internal data.**
4. What is the purpose of acceptance testing?

* Ensuring all the requirements of the customer are met.

1. Consider the following specification for a program:  
   **The input shall be one of the strings Citizen or Resident.**Based on this specification, define three distinct equivalence classes that should be used to test the program. For each class, tell whether the elements of the class should be treated as valid or erroneous.
2. **“Citizen” and “Resident”**
3. **Other things**
4. **“citizen” and “resident” – case sensitivity**
5. What is integration testing? Why are errors often found during integration testing? Name one practice that can help to reduce the number of defects found during integration testing.

* **Testing what happens when you put the different components/pieces together.**
* **One component might work by itself, but when you bring them together, they mess with each other. Interfaces might change.**
* **Solid unit tests**
* **Test each feature strongly independently**
* **Continuous Integration (CI)**
  + **Run integration constantly on your code**
  + **Automated**
  + **Working incrementally**

1. One quality for usability that can be measure is Ease of Learning. Give examples of two ways that you can design your user interface to be easy to learn.

* **Wizards/tutorials for first use or complex tasks**
* **Tool tips**
* **Follow system paradigms**
* **Meaningful icons**
* **Documentation (read-me file)**
* **Use interface conventions from similar applications**
* **Accurate message dialogues**

1. List three reasons why unit testing is a good thing to do during software development.

* **Reduces integration bugs**
* **Builds confidence in code**
* **Downstream bugs are less costly**
* **Automated**
* **Easy to run**
* **Catch regressions**
* **Forces developers to verify assumptions**

1. List any two design principles and briefly explain how the MVC architectural patterns help to adhere to them.

* **Separation of Concerns (“divide and conquer”) – The three components have their own roles and are different parts of the system.**
* **Don’t repeat yourself (DRY) – code is very well laid out so you can reuse it (use multiple views with a single model)**
* **Testability – test each component individually**
* **Reduce surprises – you expect each component to do what it is supposed to**
* **Increase cohesion – each unit works independently**
* **Reduce coupling – don’t rely on each other**

1. Describe two advantages of an incremental development software process model over the waterfall process model.

* **Allows you to adapt to change more quickly, get feedback sooner, adapt to feedback sooner.**
* **Feature driven development; work on specific features per release.**
* **Test along the way. you test each increment, so that you don’t wait for testing until the end.**

1. What is the difference between verification and validation?

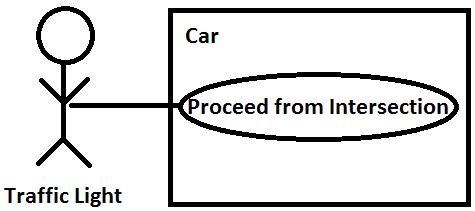
* **Verification – Did we build the product right?**
* **Validation – Did we build the right product?**

1. Using this diagram:

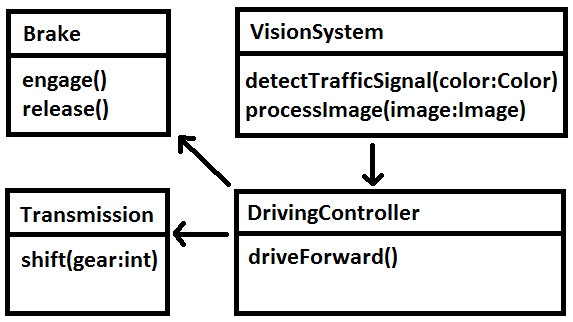
C:\Users\Serenity\Dropbox\College Docs\4010-361 Software Engineering (Robin Louvain)\Exams\se03-1.png

* 1. Describe how the UML association between Secretary and Manager would read.
* **There are many secretaries to one or more managers**
  1. According to the UML association, is it possible for a secretary to temporarily not have a manager? Why?
* **No, it specifies at least one manager (1…\*)**
* **Yes, manager can have no secretaries**
* **Yes, one secretary can have multiple managers**

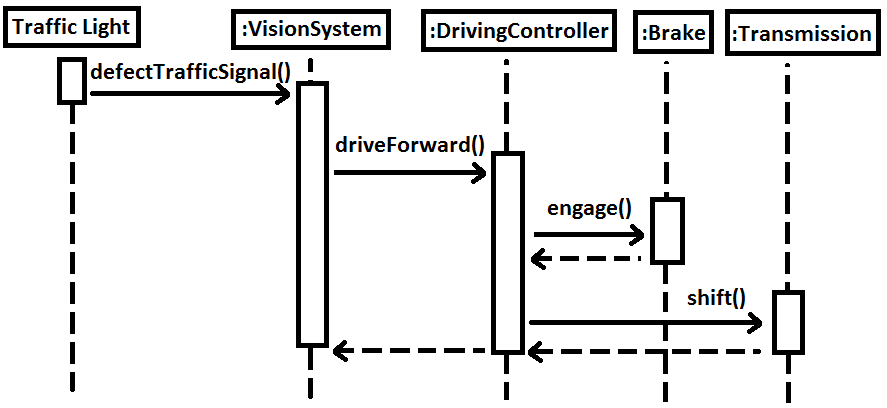
1. A self-driving car is stopped at an intersection. Given the following use case:



* 1. Create a class diagram for the car. The diagram should include any important vehicle components that contribute to realizing the “Proceed from Intersection” use case.
* Classes:
  + Camera
  + Motor Controller
  + Brake Controller (maybe)
  + Transmission (you need to reverse)
* Non-classes:
  + Car



* 1. Using the class diagram from part A, create a sequence diagram that illustrates the “Proceed from Intersection” use case.

****

* **Backwards arrows are “return arrows;” used to show that it is waiting for it to finish.**
* **Vertical lines indicate time; an “x” indicates that it has died (the “lifeline”).**

1. Read the following description of the following taxi dispatch system, and create a state diagram of the process where a user requests a taxi. Additionally, create a use case diagram outlining different aspects of the system.

A taxi company in a large metropolitan area desires to set up a new taxi dispatch system. The company’s management has two objectives. It wants to provide its customers with better service and it wants to save costs by limiting fuel consumption and trading carbon caps. The system will rely on the use of customer smartphones to request cabs and pay fares.

The customer requests a taxi by submitting a pickup request specifying the pickup location and time, the trip destination, and the number of passengers. These must happen in sequence and must allow for a manager to override each individual action. The pickup location may be an address or the customer’s current location (determined via GPS coordinates).

