**HW 4 – CS 4321, Fall 2015**

|  |  |
| --- | --- |
| Name: | Tyler Cabutto |

**Questions – Lesson 5: Requirements Engineering (43 minutes)**

Watch videos, type answers (leave questions), print before class and turn-in in class.

1. What is one of the major reasons project fail?

* Because people didn’t get the requirements right in the first place.

1. What is the definition the author provides for a software intensive system?

* The combination of software, hardware it is used on, and context in which it is used.

1. Software quality is a function of what two things?

* Combination of its software, and its purpose, where purpose is how the software is going to be used.

1. Briefly mention four reasons that identifying the purpose of software (e.g. identifying the requirements) a hard task?

* 1. Sheer complexity of the purpose/requirements:
* 2. Often, people don’t know what they want until you show it to them.
* 3. Changing requirements. Meaning the customer change their minds, stuff evolves, etc.
* 4. Multiple stakeholders with conflicting requirements.

1. To avoid the lack of completeness in requirements what do analysts sometimes do that often doesn’t help?

* They end up having features that do not work to its fullest extent and also have extra bloatware that is not necessary for the program to work.

1. List three problems that can contribute to stakeholders signing off on requirements without fully understanding them.

* Documents are unstructured and have a lot of information.
* Don’t have enough time.
* Overwhelmed with the information given.

1. Requirements engineering acts as a bridge between what two things?

* Acts as a bridge between the real-world users, customers, and other constituencies affected by a software system and the capabilities and opportunities afforded by software intensive technologies.

1. A software requirements specification is a bridge between what two worlds? Briefly discuss those two worlds.

* The machine domain which is the computer hardware, operating system, libraries, etc. where the software will run, and the application domain, which is the world where the software will operate.

1. Briefly discuss the difference between function and non-functional requirements.

* Functional requirements has to do what the system does with the computation. An example could be finding the square root in your program.
* Non-functional deals with things such as security, reusability, cost, accuracy, performance, etc.

1. What is wrong with this requirement: “The system needs to be easy to use”. How could you fix it?
2. Briefly discuss the difference between user and system requirements.

* User requirements are explained in a natural language and do not typically contain technical details.
* System requirements are written for the developers and have detailed functional and non-functional requirements.

1. Explain how the application domain is a source of requirements.

* It sets the properties and the requirements that we aim to collect.

1. List four issues that add to the complexity of collection requirements.

* 1. This spread of domain knowledge
* 2. Knowledge is tacit.
* 3. Limited observability.
* 4. Might be bias.

1. What is meant by the “thin spread of domain knowledge”?

* The knowledge is almost never written down and split among multiple people. There can also be conflicts.

1. Why/how is bias an issue in eliciting requirements?

* Because people might mislead you off the actual goal of the software you are trying to develop by giving you the wrong picture.

1. What are the three main steps in analyzing requirements?

* 1. Verification
* 2. Validation
* 3. Risk analysis

1. Why do we prioritize requirements?

* Because we may not have enough manpower, limited resources, etcetera, that would not allow us to satisfy all the requirements, thus, some need to be prioritized.

1. Describe the requirements engineering process.

* 1. Elicitation: Extract requirements from various sources.
* 2. Negotiation: Can happen between the stakeholders and the developers, during which requirements are discussed and modified until an agreement is reached.
* 3. Modeling: Represent the requirements with one or more notations.
* 4. Analysis: Identify problems we may have with our requirements.

1. In the very last video, the author mentions 2 properties of requirements (simple and testable) as well as two properties for the presentation of requirements in an SRS (organized and numbered). What the other properties of requirements that he mentioned in this lesson? (I don’t know why he didn’t mention the here).

* Machine and Application domain.