Jae Ung Kim (37007135) – Test 1 (Written Questions)

1. *What is Software Architecture and what are the key concerns?*

* Software Architecture focuses mainly on non-functional parts of the system.
* Its key concerns include: “Cross-cutting concerns” and “Conceptual integrity”

1. *What is different between Architecture and Design? (in terms of decisions and focus, best to illustrate it with diagrams)*

* Architecture is more focused on non-functional requirements.
* Architecture is basically **what the system is.**
* Design is more focused on functional requirements of the system.
* Design is basically **what the system does.**

1. *What are the conditions to have a “just enough” architecture condition to start your project?*

* The term “just enough” is extremely important in software architecture as it focuses on “Simplicity,” which is one of the 12 principles of Agile development.
* It is also known as the “art of maximizing the amount of work.”
* With “Just Enough.” In class, Dr. Yu mentioned “happy-path” which also describes the ideal “path” software development should take and is critical in software development.

1. *How does the software design activities fit into software development?*

* Software design activities include “planning, refining, and making decisions.” These fit into software development in “designing” phase to generally have a “blue-print” of where the development should go.

1. *Requirement gathering is an important stage as you (and team) start the project, what are the General 3 categories of information gathered. What are some of the problems usually happen?*

* Requirement gathering is definitely important in software development. In my COSC 499 Capstone project right now, requirement gathering is basically 30% of total grade indicating it is extremely important. General 3 categories are: **functional requirements, non-functional requirements, and Constraints**. Obviously, requirements can change over the process of development. Similar to our day to day lives, when we plan something, it never goes as we expect and can change. Similar to software development, as we develop, there might be better approach to something, system environment changes, or just simple personal problems (sick, vacation and etc.).

1. *What the three object perspectives in OO design as we discussed in class?*

* 1. Conceptual = responsibilities.
* 2. Specifications = methods.
* 3. Implementation = actual coding and data.

1. *What is different between abstraction and encapsulation? How do they relate to the three perspectives discussed in the above question?*

* Abstraction includes “conceptual” and “specifications” of object perspectives; whereas encapsulation includes “specifications” and “implementation.”

1. *What is Agile in software development*?

* Agile in software development is one of the major software developments process these days. It is open to changes and flexible. There are also 12 principles of agile development that are crucial. Software development should follow these principles to be considered “agile.” Agile development is one of the most popular development tools these days with scrum and Kanban.

1. *What are the four Agile Manifesto? Give a brief description for each of them?*

* **Individuals and interactions**
  + Individual work and interactions with the team, always team first over individual.
* **Working software**
  + make sure the software works as developed – testing is a must
* **Customer collaboration**
  + always make sure customer or whoever needs the software is satisfied.
* **Responding to change**
  + responding to changes requirements from clients. Plan can always change in software development.

1. *What is the key different between traditional and agile software development approach in term of the triple constraints (Scope, Time, Cost) of project development.?*

* Triple constraints are very crucial in agile development, as it is described with a triangle kind of an idea. Basically, if you focus on any of the two in development cycle, you will definitely fall behind in the other one. For example, if you focus on scope and time, you will fall behind in cost. In terms of triple constraints, traditional development approach was fixed most of the time and was not subject to changes unlike agile. Thus, time and cost could be extremely high if the development process failed in some ways.