**CSC354, Software Engineering I**

**Kutztown University, Fall 2015**

**Chapters 1 to 6 Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mid Term Exam One:** Answer all questions.

1. [15pts] In our discussion of developing and supporting **complex systems** there are problems or technical issues that must be considered for developing such systems. List and explain these issues. What are three ways to attack these issues?

First, the size and complexity of the system being developed must be considered.

Second, we must consider what technology and tools to use.

Third, we must consider the coordination and communication of people and resources needed to complete the work.

Three ways to attack these issues are:

* Simplifying the large, complex problem into smaller ones
* Choosing a programming language and development environment
* Deciding on a software development process to follow

1. [15pts] As part of project planning, **risk management** is an important element of software project management. Explain the different elements of risk management.

The elements of risk management that must be considered as part of project planning include:

* Risk **identification** – identify the risks that might impact the project
* Risk **specification** – formally writing them down
* Risk **assessment** – assess the impact of each risk
* Risk **prioritization** – prioritize all the risks
* Risk **re-evaluation** – review and analyse them
* Risk **monitoring** – monitor and track risks at all times
* Risk **mitigation** – have plans ready to mitigate their impact
* Risk **management** – have a formal change process

1. [10pts] The release of 2014 **CHAOS** **Report** divides software projects into the categories of successful, challenged, and failed. What are five main reasons for **challenged** and **failed** projects?

The CHAOS Report looked at hundreds of software projects in industry and identified reasons for project failures. They include (top five in red):

* **Lack of user input**
* **Lack of user involvement**
* **Incomplete requirements & specification**
* **Changing requirements & specification**
* **Lack of resources**
* Poor planning
* Lack of executive support
* Unclear goals & objectives
* Failure to communicate as a team
* Unrealistic time or resource estimates
* Technology incompetence

1. [15pts] Today’s software is intricately involved in our daily lives and it is impossible to ignore the significance of software. As professionals, software engineers must conduct their practices at some level of professionalism to ensure that their work results in no harm to society. Explain three **principles** that are recommended by the IEEE-CS/ACM for ethics and professional practices in **software engineering**.

The IEEE-CS/ACM recommends specific principles for ethics and professional practice in software engineering. They include:

* Software engineers (SEs) should act consistently with public interest.
* SEs should act in a manner that is in the best interests of their client and employer.
* SEs should ensure that their products & related modifications meet the highest standard possible.
* SEs should maintain integrity in their professional judgement.
* SEs managers and leaders should subscribe to and promote an ethical approach to the management of software development and maintenance.
* SEs should advance the integrity and reputation of the profession consistent with public interest.
* SEs should be fair and supportive of their colleagues
* SEs should participate in lifelong learning regarding the practice of their profession and promote an ethical approach to the practice of the profession.

1. [15pts] The **Waterfall Model** is the classic software lifecycle (process) model. It is a fundamental model that can be modified to fit many other process models.
   1. Explain three positive aspects and three negative aspects of the waterfall model.
   2. Describe a modified process model of the waterfall model.

Advantages of the Waterfall model include:

* Requirements must be completed in the first step
* The start and end of each phase is defined; no phase starts until the previous phase is complete.
* The output from one phase is fed into the next phase in sequence.
* The output from one phase to the next is clear defined.
* It is easy to track the progress of the software project

Disadvantages of the Waterfall model include:

* A later phase cannot start until the previous phase is complete
* Unable to iterate back to a previous phase if a problem is found.
* No concurrent work can be done; resources stay idle
* Lots of documents are produced (both good and bad)

Two modified examples of the Waterfall model are the Iterative Model and Incremental Model.

1. [15pts) List and explain five fundamentals characteristics of **Agile methods**.

Agile processes are a family of software development methodologies that produce software in short iterations and allow for greater changes in design.

Characteristics of Agile methods include:

* Short releases and iterations – divide work into small pieces and release software to customer often
* Allow for incremental design – create a simple design, work with it and improve the design when more knowledge is acquired
* Lots of user involvement – work with users to receive constant feedback
* Minimal documentation – do only the necessary amount of documentation; focus on the people
* Informal communication is encouraged – communicate with other people; often for better understanding
* Embrace change – change is expected; find ways to deal with it

1. [15pts] **Software Requirements**
   1. What are the two general classifications (types) of software requirements?
   2. Give an example of each type of requirements for an Automated Teller Machine (ATM).
   3. List and explain the steps in the Requirements Engineering process.

The two general classifications of software requirements are Functional and Non-functional requirements.

For an Automated Teller Machine (ATM):

an example functional requirement would be “User wants to withdraw a certain amount of money.”

an example of non-functional requirement for ATM would be “There must be a response to each user request within 5 seconds.”

Steps in the **Requirements Engineering** process include:

* Elicitation – talk to users about requirements for the system.
* Documentation & Definition – documenting and defining terms and concepts.
* Specification – formally writing down the requirements in SRS document.
* Modeling & Prototyping – use models to communicate terms & concepts to other project members; create a prototype for demonstration/presentation.
* Analysis – analyse the requirements for accuracy, consistency, and completeness.
* Review & Validation – review requirements with users to confirm them.
* Agreement & Acceptance – after agreement with users on the requirements, sign a contract on the scope of the system.