CS 311 - Introduction to Software Engineering

**Assignment 01: due (Friday, Sep 28, 2018)**

# Part 1: Software Process | Software Development Life Cycle

**Q1)-** The government is planning to develop a new **"cloud-based electronic voting system"** The project manager wants to choose a process model what do you think is the suitable one? The software requirements are clear, but some how-to tasks related to security and data privacy need some research and risk management. Which software engineering process model (Software Development Life Cycle model) do you recommend for that project and why?

**Sample Answer (no need to answer Q1, it only serves as an example to show you how to answer the remaining questions in Part 1)**

Based on the given information it seems that the V-model is an appropriate software process model for the cloud-based electronic voting system. An electronic voting system is a critical system which needs an accuracy of data collection and privacy of data collected and how to manage fraud cases and detect them, which need deep business understanding and risk management as well as excellent testing cases.

This system cannot be operational in phases or iteration. For example, the voters cannot start casting their votes to candidates if we do not have an identity management module or fraud detection and these modules will not operate until we have the vote casting module implemented. Therefore, the use of Agile or iterative/incremental models is not reasonable. In addition, the size of the development team for this system is expected to be large and possibly from different organizations. Finally, the requirements of an electronic voting system are unlikely to change (major change) during the project.

**Q1.2)-** Smart Apes is a software house that is currently working on an app to help seniors and elders living in retirement homes to manage their daily activities. Mainly the app will help them to schedule medical appointments, medication reminder, order grocery, and transportation services when needed. The company never developed apps for seniors and elders before. They are concern about the requirements and mainly the usability of the app. In the context of this above scenario as a project manager what will be the choice of the software lifecycle model? **[5 POINTS]**

Based on the given information, it seems that the Agile model would be an appropriate software process model for a senior’s assistance app. A senior’s assistance app is a simplistic system which could be subject to constant feature and requirement changes as new issues and needs arise. An Agile model would allow for these new requirements to be accommodated easily. Additionally, Smart Apes is not very specific with their requirements (i.e. For scheduling appointments, should it be able to contact the medical professional directly to schedule? Should it just schedule a pre-existing appointment in the user’s calendar? etc.) and an Agile model would allow changes to be made easily in the software. Finally, this simplicity of making changes would also be useful for the implementation of a UI, which is necessary due to the developers being concerned with the usability of the app.

**Q1.3)-** Suppose you are developing a software product in a new and growing market (e.g., cryptocurrencies) with your competitors who are also developing a product will be the same product. Which software process model to select and why? **[5 POINTS]**

Based on the given information, it seems that the Iterative and Incremental model would be an appropriate software process model for a given software under a strict time constraint. Since the software being developed is also being developed by the competitor, being able to develop it more quickly than the competitor is vital. The Iterative and Incremental model is one of the most effective software process models when time is of the essence. Additionally, it is known that the software is being developed for a ‘new and growing market.’ Due to this, I would choose the Iterative and Incremental model over the Agile model since the technology we are working with could be unfamiliar depending on the market and how close to the introduction of the market we begin developing our software.

**Q1.4)-** We are creating an online system to manage and visualize security alerts generated by different open source security appliances such as intrusion detection system (e.g., Snort, Bro), firewalls (IP Tables, firewalls) and it is needed immediately, what kind of software- development-life-cycle model should be used and why? **[5 POINTS]**

Based on the given information, it seems that the Integration and Configuration model would be an appropriate software process model for a security software under a strict time constraint. The software is dealing with security, but according to the description it is only ‘managing and visualizing security alerts.’ With this being the case, the software being developed does not need to handle security as it is essentially only a hub to view security notifications, not to deal with or determine the security issues. It does however need to be completed immediately, and the components required are already in a working state. Therefore, this leaves the Integration and Configuration model as the best option.

# Part 2: Requirement Analysis and Engineering

**Q2.1)** You are building a highly interactive website to replace the current website of the school of computer science at the University of Far-Far-Away. Identify the different groups of primary and secondary users of the website. For each group describe between 3-5 functional requirements. How could you verify these requirements? In your opinion what are the possible non-functional requirements for this website and which ones are the most important? **[25 POINTS]**

Primary Users Functional Requirements (FRs):

Students FRs:

1. Ability to plan and commit course schedule, similarly to myUWindsor with course planning and scheduling.

2. Access a schedule of all due dates and test dates, which would be inputted to a single calendar by the professors and would give students a single place to track all due dates and test dates.

3. See which semesters each course is available

(e.g. Fall-Yes, Winter-No, Summer-Yes)

CS Faculty FRs:

1. Ability to overwrite courses, which would be a simpler alternative to the Add/Drop Form.

2. Ability change course times/rooms, another simplification to the current processes which would be more for professor/faculty benefit rather than for students.

3. Access student’s info in a simplistic page. (schedule, grades, personal info, etc.)

Professors FRs:

1. Sign students into classes, ability to bypass the Add/Drop process with professor approval.

2. Upload marks directly to an overall report card, showing current grades and past grades in a simplistic page.

3. Add test dates and due dates to a program schedule, which would be for all CS courses to viewable by students and staff.

Secondary Users:

None based on the definition derived from my Primary User’s requirements.

Verification:

To verify these requirements, I would implement the prototyping technique, producing executable models and checking that the features involved match the needs of the users and the requirements outlined. If the current features match and work, then we would proceed to the next prototype and implement further features, otherwise we would update the current prototype to ensure that the current features are working.

Non-Functional Requirements (NFRs):

* Security Most Important
* Stability l
* Scalability l
* Portability l
* Maintainability l
* Performance l
* Usability l
* Effectiveness V
* Simplicity Least Important

Most Important NFRs (In Order of Importance):

* + Security
    - Certain pages, such as student info page, personal schedules, and the grades page, should be password locked so that only the relevant users can view the given page.
  + Stability
    - This should easily be able to handle up to hundreds of users at any given time with low risk of the software going down, since this is a software that should be used consistently by students to manage their day to day studies.
  + Scalability
    - This software needs to be scalable since the School of Computer Science at the University of Windsor has been growing in recent years, so the software should be scalable for further increase in the School of CS’s size.
  + Portability
    - Since this software is going to be used by mostly students, having portability would be a very useful feature. Students rely on their different devices, so being able to port the website to mobile or tablet as well would be a very welcome feature.
  + Maintainability
    - This software will be used regularly by students and they is the possibility of new features being recommended or even required, thus maintainability would allow for these changes/updates to be implemented simply.

# Part 3: Agile Software Development

Please read the paper **“Impact of Agile Methodology on Software Development Process”** by Gaurav Kumar, Pradeep Kumar Bhatia and answer the following questions [there is a link on BlackBoard to download the paper]

Q3.1) What are the advantages and the limitations of the Agile methodology according to the authors. Do you agree or disagree with the authors? **[5 POINTS]**

The advantages of Agile methodology as indicated by the authors are the ability to easily handle changes in requirements, the ability to detect faults earlier in development, improved performance of the developers through frequent interactions and communication, iterative and incremental delivery so that progress and feedback are monitored frequently, flexibility of design allowing for changes in direction of the software, and an improvement in quality due to the test-driven development and refactoring use.

The limitations of Agile methodology as indicated by the authors are that it is focused on the development of the software rather than the design and end user, high testing lead times and low testing coverage, requires a lot of coordination and communication between teams/developers, poor scalability to larger projects due to the many iterations, possibility to overcommit time to any given feature, possibly high opportunity cost on large scale projects, and a large management overhead due to agile relying on teamwork and communication, thus relying on strong and involved management.

I do not disagree with any of the advantages put forward by the authors in this paper. I would add that Agile methodology also benefits from its capacity to complete software quickly. If there is a strict time schedule, Agile methodology can accommodate this through the ability to push working software in phases. You could have a very basic working software and be able to fluidly change and improve upon it in the following phase.

Q3.2) Do you think the authors in this paper as a positive, negative, or neutral opinion of adopting Agile methodology in software development and why you think their view is positive, negative, or neutral? **[5 POINTS]**

I believe that the authors of this paper have a positive opinion of adopting Agile methodology. My first reason for believing this is that even though they considered advantages and limitations, the difference in emphasis and detail put into the advantages versus the limitations was apparent. The advantages were each given a separate section with a subtitle and accompanying blurb, whereas the limitations were all listed as a singular point form list. Also, in the conclusion most of the discussion revolves around the advantages with only a brief mention of limitations. The overall conclusion that I felt they are giving is with regards to the benefit of adoption Agile methodology.