**ITMD: 511 Application Development Methodologies**

**Chapter 1: Introduction to Software Engineering**

**Exercise 1**

**1.1: Explain why professional software that is developed for a customer is not simply the**

**programs that have been developed and delivered.**

Certain Design standards are followed for building professional software for customers use. Software includes not just the applications themselves, but also the related documentation, frameworks, support websites, and configuration files required to make these programs work. Often, a properly built software consists of multiple programs. A software system can be built using multiple different programs and the configuration files used to set them. It could not only include system documentation that tells us the system's structure but also user manuals, which explains how to use that system, and websites where users can access the most up-to-date product information. Consider an example of banking software which goes through various stages of software building process such as requirement gathering and analysis, coding and testing, maintenance. This banking requires having various configuration files to be setup as per the countries as languages differ. The banking software is also provided with documentation on transaction screens and their functionalities.

**1.2:  What is the most important difference between generic software product development and custom software development? What might this mean in practice for users of generic software products?**

**Generic software**: These are stand-alone software systems that are created by a development firm and sold on the open market to anyone who can afford them. Consider below examples of apps built for mobile devices, as well as PC applications. Building an expense management software is an example of generic software used by many students to manage their daily/monthly expenses and are available at cheaper costs.

**Custom software:** Custom software are systems that are designed and developed especially for a specific customer/end user. Building control systems for electrical based companies wherein the system is built to assist a certain business process and controls and developing traffic management systems for police are all examples of this sort of software. Above software is built by considering specific requirements provided by the customer i.e., banks/electrical companies.

I think the most important difference between generic and custom software is that with generic products, the software specifications are controlled by the developers of the software firm that actually built the system. It means that software building firms can rethink what has to be built if they come under any development issues. This means the users using generic software has to accept changes provided by software building firms. Here the end user cannot ask for changing the specification of software in generic software. Consider an example of Facebook where we have to update applications whenever Facebook sends upgrades. Vice-versa, the organization/end user that is buying the software, has full control over the specification for custom software, for example a bank can ask for changes in the specifications and the standards provided by the bank must be followed by the software developers.

**1.3: Briefly discuss why it is usually cheaper in the long run to use software engineering methods and techniques for software systems.**

The majority of costs involved for most types of software systems are the software CR (change requests) costs when it has been delivered to the end user. Because Software Engineering approaches are thorough, they increase the initial cost, duration, and efforts associated with a new project. However, the same methods lead to a stronger base, creating software which is easier to support, modify, or expand upon; more accessible for new programmers to come in and be able to understand due to a concrete design process having been used; more accessible to users (or auditors) due to the extensive documentation; and generally, more tested properly. As a result, using software engineering processes and techniques for software systems is always less expensive in the long term.

**1. 4. Software engineering deals not only with problems such as system heterogeneity, business and social change, trust and safety, but also with ethical issues affecting the sphere. Give some examples of ethical issues that have an impact on the software engineering domain.**

**Diversity:** Race, gender and age distributions of testers/developers in the development team of software firms can have an impact on the software engineering domain. Ethics should be in place that prevents such distinction and brings harmony in the workplace to work in coordinated fashion allowing multicultural environments.

**Work Ethics:** Software engineering team should follow proper development model which includes requirements gathering, analysis, development and testing, maintenance. The end product must be tested for bugs before releasing quality software ensuring the quality of code before release.

**Accountability:** At first, every software product delivered to the end user is not 100% complete and can have significant failures when used by stakeholders/end users. The software engineering domain should follow the ethics in accepting accountability for the harms caused by their product in case of any failures at the end user and should work on fixing the issues and improving usability.

**Transparency:** Software organizations should be transparent about decision making procedures used while building software products. Consider an example of Facebook/Google where they show us ads/suggestions based on our search history. The software organizations should be transparent about decision making procedures used and ethics policies should be made publicly available on how they are using customers' search data to provide them with different suggestions.

**Business Ethics:** End user/customer of software engineering firm can have business impact if not provided with changed business model especially including revenues models. Engineering domain must notify the stakeholders about changed business models to build trust.

**Privacy:** Storing and handling user information only under certain circumstances and for the purposes that the user sets while giving his/her personal information. Consider an example of PII given to insurance firms (BCBS/Aetna) should be used for claims or by doctors. Breach of PII by any software developers/testers working in insurance firms can adversely impact someone’s life. Hence software engineering should follow engineering ethics for the protection of user data. Engineering domain should focus on implementing proper cyber security for the protection of PII data.

**1.5: Based on your own knowledge of some of the application types discussed in Section 1.1.2, explain, with examples, why different application types require specialized software engineering techniques to support their design and development.**

Because each application has its own set of difficulties to handle, it necessitates the use of specialized methodologies. For example, as a compiler specialist, I am quite familiar with certain types of directed graphs and binary arithmetic, but if you ask me to solve a graph problem that isn't in my area of expertise, such as max-flow or shortest path, I'll have to search up the algorithm. Also, some software applications have a limited lifespan, while others can last for tens of years. Consider building an application of an insurance firm that requires the handling of covid related claims, in such cases the software systems become of prime importance, and they must be delivered rapidly. Building of such functionality in an application can be done by agile methodology rather than following a waterfall model. Agile methodologies are getting used mostly to deliver quality software in a short period of time. However, software systems that runs for long run require methodologies that will need long-term support, system design and analysis, maintenance, change request handling, configuration changes, regional language support if used across globe, which is not ideal for short-span quick delivery systems. Example being, building functionality of airplanes when we consider every aspect before delivering a product as it involves protecting human lives.

**1.6: Explain why the fundamental software engineering principles of process, dependability, requirements management, and reuse are relevant to all types of software systems.**

Software is a collection of programming instructions that are executed to carry out a certain task based on the user's requirements. There are several fundamental rules that must be followed by every software. User requirements management, platform dependability, development process definition, updating, and maintenance are only a few of these principles. Both generic and custom software follow these concepts. In both generic and customized software, the process of gathering requirements is essential. Every sort of software program requires a document that defines the development process. Every form of software requires updating. New versions of each sort of software are released to do this. Software development is incomplete without maintenance. Every form of application development necessitates it. In order to run software, it must meet certain basic requirements. As a result, platform dependability is taken into account in every program development.

**1.7: Explain how electronic connectivity between various development teams can support software engineering activities.**

Communication is a critical component of team collaboration. Using electronic communications such as outlook, WebEx, Microsoft teams at workplace, allows various development teams to share their files and communicate without being physically present in the workplace. It saves lot of travel time of employees and especially when there is an emergency employees can communicate electronically so that there is no harm/loss to the team. Electronic communication, whether it is in the form of an email or a video conversation, can boost productivity and enhance work patterns. Employees are no longer needed to sit in long hour meetings or drive large distances to discuss ideas/thoughts or make decisions. Consider an example of GITHUB as it is very popular tool for communication between the different team members by contributing their code to the single repository and can have backup of their code changes over the time. We have seen in this covid pandemic, remote work is getting popular and electronic communication is getting of prime importance for every software product development process.