[![Review Assignment Due Date](https://classroom.github.com/assets/deadline-readme-button-24ddc0f5d75046c5622901739e7c5dd533143b0c8e959d652212380cedb1ea36.svg)](https://classroom.github.com/a/-ucQIGTc)

[![Open in Visual Studio Code](https://classroom.github.com/assets/open-in-vscode-718a45dd9cf7e7f842a935f5ebbe5719a5e09af4491e668f4dbf3b35d5cca122.svg)](https://classroom.github.com/online\_ide?assignment\_repo\_id=15242953&assignment\_repo\_type=AssignmentRepo)

**# SE-Assignment-2**

Assignment: Introduction to Software Engineering

Instructions:

Answer the following questions based on your understanding of software engineering concepts. Provide detailed explanations and examples where appropriate.

Questions:

Define Software Engineering:

It is a way of coming up with softwares or programs that are packaged once ready to achieve certains tasks. its a combination of codes to achieve innovative solutions.

What is software engineering, and how does it differ from traditional programming?

software engineering is the process of coming up with a software to solve real life issues. it takes the SDLC process to come up with a software. Traditoin programming is where the programmer writes explicit instructions that tell the computer what to do to solve a specific problem or perform a certain task.

Software Development Life Cycle (SDLC):

Explain the various phases of the Software Development Life Cycle. Provide a brief description of each phase.

The SDLC is a process of software development.

The different stages are;

requirements gathering : gathering and documenting user and system requirements

design : coming up with the a graphical desing of the system depicting what the end product should look like

implementation : writing and building the software according to the drawn design

testing : conducting various tests to ensure the software meets the quality standards and functinal requirements

deployment : release the software to the users. Making it available for usage.

maintenance : providing ongoing support,enhancement and updates to the software after deployment. This fixes bugs and adds new features to the sofwater with time.

Agile vs. Waterfall Models:

Compare and contrast the Agile and Waterfall models of software development. What are the key differences, and in what scenarios might each be preferred?

Agile is iterative and incremental in nature with flexibility in collaboration and change. It is best for projects where the significant contraints are well defined. Will mostly suit projects in which a completely new product is expected as the deliverable.

Waterfall is sequential in nature and has disctinct phases. it is best for projects with concrete timelines and well-defined deliverables.

Requirements Engineering:

What is requirements engineering? Describe the process and its importance in the software development lifecycle.

requirements engineering is the systematic process of identifying, eliciting, analyzing,specifying,validating and managing the needs and expectations of stakeholders for a software system.

the process has the following steps in it :

Feasibility Study : current resources both hardware software along required technology are analyzed/assessed to develop the project

Requirements elicitation : It is related to the various ways used to gain knowledge about the project domain and requirements. The various sources of domain knowledge include customers, business manuals, the existing software of the same type, standards, and other stakeholders of the project. The techniques used for requirements elicitation include interviews, brainstorming, task analysis, Delphi technique, prototyping etc.

Requirements specification : This activity is used to produce formal software requirement models. All the requirements including the functional as well as the non-functional requirements and the constraints are specified by these models in totality

Requirements for verification and validation : Verification: It refers to the set of tasks that ensures that the software correctly implements a specific function.

Validation: It refers to a different set of tasks that ensures that the software that has been built is traceable to customer requirements. If requirements are not validated, errors in the requirement definitions would propagate to the successive stages resulting in a lot of modification and rework.

Requirements management : Requirement management is the process of analyzing, documenting, tracking, prioritizing, and agreeing on the requirement and controlling the communication with relevant stakeholders

Software Design Principles:

Explain the concept of modularity in software design. How does it improve maintainability and scalability of software systems?

Moduöarity is the division of the parts of a complex software into small manageable independent modules. this facilitates easier management. By dividing software into distinct modules, developers can manage, update, and scale parts of the application independently, leading to more efficient maintenance and easier scalability.

Testing in Software Engineering:

Describe the different levels of software testing (unit testing, integration testing, system testing, acceptance testing). Why is testing crucial in software development?

Testing is crucial in the life cycle of a software because it assists developers maintain quality while ensuring the software is free of defects and bugs.

unit testing : this teste the individual units of the software developed. its mainly used to test individual components of a software to ascertain that they each individually meet the requirements and delivers the expected output.

Integration testing : individual software modules are logically integrated and tested as a group. thid is to idnetify which the individual modules which work together while itegrated or as a group.

System testing : this level identifies bugs and challenges while ensuring the software meets all specific requirements and runs smoothly in its operating environment.

Acceptanc testing : This determines whether or not the software should be released to the public. This level evaluates whether the software complies with the end-user requirements and whether its ready for deployment.

Version Control Systems:

What are version control systems, and why are they important in software development? Give examples of popular version control systems and their features.

a version conrol system tracks changes made to the source code by diffeent engineers, and helps cordinate work within a team of engineers/developers. The VSC are compatible with any environment. examples of such systems are git and subversion. it keeps track of modicifation in the code making it easy to track mistakes if there are any.

VCS allows long-term change history of every software,brnching, merging and traceability of the codes.

Software Project Management:

Discuss the role of a software project manager. What are some key responsibilities and challenges faced in managing software projects?

Software project managers are responsible for prepartion and implementation of the software projects. they analyse project constraints,establish project objectives, cordinate teh projects internal and external teams, construct the project timelines and monitor the project’s key performance indicators.

a project manager does project esetimation(cost, time,effort),staffing,scheduling,risk management and miscellaneous plans.

Software Maintenance:

Define software maintenance and explain the different types of maintenance activities. Why is maintenance an essential part of the software lifecycle?

Software Maintenance refers to the process of modifying and updating a software system after it has been delivered to the customer. it is necessary to ensure that the software continues to meet the needs of the users over time.

Corrective maintenance : fixes bugs and errors in the software

patching : its an emergency fix done due to proessure from management, but always results in bugs due to lack of proper analysis.

Adaptive maintenance : this involves modifying the software system to adapt it to changes in the environment, such as changes in hardware or software, government policies, and business rules.

Perfective maintenance : This involves improving functionality, performance, and reliability, and restructuring the software system to improve changeability.

Preventive maintenance : This involves taking measures to prevent future problems, such as optimization, updating documentation, reviewing and testing the system, and implementing preventive measures such as backups.

Ethical Considerations in Software Engineering:

What are some ethical issues that software engineers might face? How can software engineers ensure they adhere to ethical standards in their work?

examples of ethical issues faced by developers are addictive design;, corporate ownership of personal data; algorithmic bias;weak cyber security and personally identifiable information (PII) protection; and overemphasis on features.

Collaboration between engineering leadership and legal teams can avoid ethical shortcomings.

Submission Guidelines:

Your answers should be well-structured, concise, and to the point.

Provide real-world examples or case studies wherever possible.

Cite any references or sources you use in your answers.

Geek for geeks

Atlassian

Submit your completed assignment by [due date].