Assignment-1

1. What are the symptoms of present software crisis? What factors have contributed to the making of present software crisis? What are the possible solutions to present software crisis.

Ans: Software crisis refer to a state or situation in the software development and development process where there are a significant challenges, issues or problems that hinders the successful delivery of software projects.

Symptoms of the current software crisis are:

1. project Delays: Software projects frequently exceed planned schedules, leading to frustration and increased costs. This leads to project delay.

2. Budget overruns: Budget for software projects are often exceeds due to unforeseen challenges and scope creep.

3. Quality Issues: Software may suffer from frequent bugs, errors and inconsistencies, impacting its reliability and user experiences.

4. Security vulnerabilities: Increasingly complex software system may have vulnerabilities that can be exploited, compromising data security.

5. Lack of scalability:
Software architectures may struggle to handle increasing user loads or evolving business requirements.

6. Communication challenges:
poor collaboration and misalignment between stakeholders, developers, & users can lead to misunderstandings and ineffective solutions.

Factors contributing to software crisis are:

1.) Rapid Technological Advancements:-
The fast pace of technological change can make it challenging for software teams to keep up and incorporate ~~for software team to be~~ the latest tools and frameworks effectively.

2.) Increasing complexity:
Modern software systems often have intricate architectures, dependency and integration challenges making development and maintenance more difficult.

3.) ~~Inadequate~~ Inadequate testing:-
Insufficient testing practices and limited test coverages can result in undetected bugs and decreased software quality.

4.) poor project management:
Ineffective planning, inadequate resource allocation and improper risk management can contribute to project failures.

5.) Changing user Expectations.
Evolving user preferences and rising expectations for intuitive interact and seamless experience can create challenges in meeting user needs effectively.

6. Lack of Standardization:
The absence of widely adopted industry standard and best practices can lead to inconsistent approaches to software development.

possible solutions to the software crisis:

1.) Agile and DevOps practices.
Adopting Agile methodologies and DevOps principles can enhance collaboration, improve flexibility and deliver software iteratively with faster feedback loops.

2. Test-Driven Development:
Emphasizing automated testing and employing practices like Test-Driven Development (TDD) can help identify issues early and improve software quality.

3: Continuous Development, Integration & Development. Implementing automated build, test and deployment process can streamline development, Integration and workflows and reduce errors.

4. Robust & Requirements Engineering:
Throughly understanding and document user requirements, involving stakeholder and conducting regular reviews can improve project success.

5. Secure Development practices:
Incorporating Secure coding practices, performing regular security audits and maintaining awareness of potential vulnerabilities can enhance software security.

6) Adoption of Industry Standard:
Following established software development standard and best practices can increase consistency, interoperability and overall software quality.

Q.N.2) Explain how do the use of software engineering principles help to develop software products cost-effectively and timely. Elaborate your answer by using suitable examples.

→ The use of software engineering principles plays a crucial role in developing software products cost effectively and timely. These principles provide a systematic approach to software development, ensuring that best practices are followed throughout the process - there are some ways software engineering principles contribute to efficient and timely development :

i.) Requirement Analysis and planning :
Software engineering and principles emphasize thorough requirement and effective planning. By clearly understanding the project requirements upfront, development teams can avoid costly rework and ensure that the software product meets the desired objects. For eg :-
using techniques like requirement elicitation, prototyping and user feedback, engineers can gather accurate and comprehensive requirements, reducing risks of misunderstandings and delays.

ii.) Modularity & ~~Readability~~ Reusability
SEP promote modular design and code reusability. Breaking down the software into smaller, manageable modules make it easier to develop, test and maintain. ~~modules make it easier to develop, tes~~

iii) Iterative & Incremental Development:
Agile methodologies, which are rooted in SEP, advocate for iterative and incremental development. For example, using Scrum framework a software project can be broken down into short sprints, where each sprints delivery a potentially shippable product increament.

iv. Testing and Quality Assurance.
SEP emphasizes the importance of regular testing and quality assurance practices. By implementing various testing techniques, such as unit testing, integration testing and system testing, software engineers can fix and identify defects early in development.

v.) Version control and configuration management:
using version control systems like Git allows developers to manage changes to codebase, collabrate efficiently and revert to previous versions if needed. Additionally, configuration management tools help track and manage software dependencies, ensuring consistent and reproducible build.

Q.N.3.) Assume that a software development Pt company is already experienced in developing payroll software and developed similar software.

Ans: The waterfall model seems the ideal choice here. No other model seem a resonable alternative

Strengths:

a. considering that the company is software development, Pt is, large and bureautic, requiring, documentation and good project visibility. Waterfall documentation ~~and good project visibility. Water-fall document~~ satisfies this.

b. Since, they have already developed the software many times, the application type is well known to company, so a though a user requirements analysis should be possible early on.

c.) The company has good reputation to maintain in this area as they have built product many times, so quality control will be essential. waterfall emphasis of requirement, before design and design before coding will emphasize quality.

Weakness:-

The waterfall has no obvious weakness in this project.

a.) We may endup swimming upstream, but the changes seem minimal since the requirements should be clear from the start.

b) Late arrival of code may be one but on other hand, our client may be able to continue using existing application until new system arrives.

Q.No.4.) The problem stated by client have uncertainities which lead to loss. So it not planned and solved.

Ans: project risks are caused by the problem statements uncertainability. Consequently the spiral model can be applied to the creation of such a project.

pros :-

i) later on, it is possible to add functionality or make adjustments.

ii.) Risk management benefits from continuous or repeated development.

iii.) With spiral development, features are developed quickly and methodically.

cons :-

i.) Risk of failing to the budget or schedule.
ii.) only large project benefit from spiral model, which also necessities proficiency in risk assessment.
iii.) Because there are intermediary steps, documentation is more.

Q. No. 5.) What is requirement engineering ? Explain it's steps.

Ans: Requirement engineering is the proces of identifying, eliciting, analysing, specifying and managing the needs and expectation of stakeholder for a software system. It involve following steps.

i.) Requirement Elicitation:
This is the proces of gathering information about the needs and expectations of stakeholders for the software system. This steps involves interviews, surveys, focus groups, and other techniques to gather information from stakeholders.

ii. Requirement Analysis :
This step involves analysing the information gathered in requirement elicitation step to identify the high-level goals and objects of software system.

iii.) Requirement Specifications :
This step involves documenting requirements identified in the analysis step in clear, consistent and unambiguous manner.

iv. Requirement validation :
This step involves checking that the requirements are complete consistent and accurate. It also involves checking that the requirements are testable and meet the needs the needs and expectations of stakeholders.

v. Requirements management :
This step involve managing the requirements throughout the software development life cycle, including tracking and controlling changes, and ensuring that requirements are still valid and relevant

Q.No) Although the industry is moving towards component based construction, most software continues to be custom built. Explain ..

Ans: Component based construction offers several advantages such as faster development, reduced costs, improved quality and increased reusability

However, despite custom-built software continues to be prevalent in many scenarios for following reasons.

i) Unique Requirements.
- Many organizations have specified unique requirements that can not be fully addressed by pre-existing software components.

ii) Competitive Advantage:-
Custom-built software can provide a competitive advantage to organization by allowing them to differentiate themselves from competitors.

iii) Legacy Systems Integration:
many organization already have legacy systems in place that are deeply ingained in their operation. It may not be feasible or cost-effective to replace these systems with pre-built components.

iv) Domain-Specific Requirements:
certain industries or domain have specific requirements that may may not adequately addressed by existing software components.

v) Scalability and future Growth:-
Custom built software can be designed with scalability in mind, allowing

organization to accommodate future growth and changing needs. It provides flexibility to scale and adapt as the organization evolves.

vi) Intellectual property and ownership :-
Some organizations consider their solutions as part of their intellectual property and source of competitive advantage. It allows them to protect their proprietary technologies and maintain full control over their software assets.