Software development life cycle

Key Concepts

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

SDLC Activities

SDLC provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps

1. Communication
2. Requirement gathering
3. Feasibility analysis
4. System Analysis
5. Software Design
6. implementation or coding
7. Testing
8. Integration
9. Implementation
10. Maintenance
11. Disposition

- Different approaches to life cycle methodology are: waterfall model, Prototyping Model, Spiral Model, Incremental Model RAD model.

Relevance

- In order to maintain a good working phase, SDLC is incorporated.
- SDLC can help build the product with better quality and estimate the cost beforehand.
- SDLC can help build a more dedicated team.

Real World Contexts

- SDLC can be applied to both hardware and software.
- SDLC produces high quality systems that meet customer expectations by estimating schedules and estimated costs.
- SDLC helps in planning and execution of the project in chaos free manner.

Interdisciplinary Connections

- SDLC evolves at many phases that can help build a better product.
- SDLC bridges the gap between customer and technical aspects of the project.
- SDLC takes many parameters into consideration to arrive at a solution.

Engage in critical thinking	**Technology, Tools and Test Case**
Analyze and build a solution for the problem indicated by the customer.	Slack Gmail GitHub Google Docs
Plan project management	**Project specification**
Project management can be efficient if the chooses the correct life cycle model.	● Helps to gather specification and requirements for the project. ● The team follows an accepted life cycle, cost is estimated and role assignment is done.

Various Software Development Methodologies:

Key Concepts	**Relevance**
● Different approaches to life cycle methodology are: waterfall model, Prototyping Model, Spiral Model, Incremental Model, RAD model, SCRUM Model.	**Waterfall Model:** Waterfall model is the simplest model of software development paradigm. It says the all the phases of SDLC will function one after another in linear manner. That is, when the first phase is finished then only the second phase will start and so on. **Prototype Model:** The Prototyping Model is a systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. **Spiral Model:** Spiral model is a combination of both, iterative model and one of the SDLC model. It can be seen as if you choose one SDLC model and combine it with cyclic process (iterative model). **Incremental Model:** The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. **RAD Model:** RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. **SCRUM Model** Scrum is an agile software development model based on multiple small teams working in an intensive and interdependent manner.

Real World Contexts	Interdisciplinary Connections
In real world SCRUM model will be more helpful as the project is developed independently by developers and .	The model takes into consideration various parameters like the number of employees, requirements, cost estimated, time of every sprint, etc.

Engage in critical thinking	Technology, Tools and Test Case
● It involves co-ordination. ● Every sprint needs to be carefully planned. ● Requires more of SCRUM Master involvement. ● Integration of the entire project has to be planned.	No specific tools.
Plan project management	Project specification
The team is decided, the scrum master is choosen. Requirements are gathered. Cost is estimated, etc	The team has understood their roles and the time limits they have. They have peer pressure and undertake responsibilities.

Software Requirement Specification

Key concepts	Relevance
A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide and provides Problem analysis, requirement specification, validation, metrics, monitoring and control.	The Software Requirements Specification (SRS) is a communication tool between stakeholders and software designers. The specific goals of the SRS are: ● Facilitating reviews ● Describing the scope of work ● Providing a reference to software designers (i.e. navigation aids, document structure) ● Providing a framework for testing primary and secondary use cases ● Including features to customer requirements ● Providing a platform for ongoing refinement (via incomplete specs or questions) Reliability Availability Security Maintainability Portability.
Real World Contexts	Interdisciplinary Connections

● Every member's ideas and requirement must be considered at this stage. ● The requirements must be well documented so as to avoid chaos.	● Every team member are aware of its peer's idea. ● Helps in resolving any conflicting requirements issues. ●
Engage in critical thinking	**Technology, Tools and Test Case**
● Requirements analysis needs to solve the problems addressed by every team member regarding the requirements. ● However since the scope of critical thinking has been very diverse.	● For management Slack and gmail were used and the diagrams were made by Creately software tool.
Plan project management	**Project specification**
● It includes planning Characteristics Implementation Operations Interfaces	● Defines all the functionality. ● All the requirements are documented. ● Requirements are analyzed.

System Design

Key concepts	**Relevance**
● **Systems design** is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.	● System design is made at this stage ● Helps in proper making of the project ● Dependability and Interdepend ability is checked
Real World Contexts	**Interdisciplinary Connections**
● Effective testing and debugging by taking care of unnecessary details. ● Level of interaction among modules with each other is explained. ● Requirements are discussed beforehand.	● Maintains an abstract design and integrates functionalities of the software. ● Optimum results are passed and not the entire result. ● Proper working of the component and its relation with other components are checked.
Engage in critical thinking	**Technology, Tools and Test Case**
● Increasing number of modules creates problem in integrating them.	● Creately, Photoshop.

- Better coupling can be achieved by less coupling. - Higher the coupling, high the performance of the program.	
Plan project management	**Project specification**
- The sprint for the design were for short term and repetitive terms. - Clients views were considered at every meeting	- The architecture of the system had to be considered.

Coding

Key concepts	**Relevance**
- Information protection, structured programming, top down approach was accounted.	- Modules triggered other component of other modules. - The difficulty to trigger between modules was to be accounted.
Real World Contexts	**Interdisciplinary Connections**
- Program is separated into several blocks in order to understand particular segment of code in a software code. - Entry points , multiple exit and statements are reduced in a program. - Detail abstraction between modules.	- Functions and procedures are globally visible. - Modification and abstraction in done.
Engage in critical thinking	**Technology, Tools and Test Case**
- Dependent modules are hard to implement - Information hiding doesn't take part in creation of modules. Rely on other modules.	- Visual Studio - Github. - Ionic - Koha Library System - REST Easy - Firebase - Slim Framework - Self-Analysis.
Plan project management	**Project specification**
This phase needs lot of management. This has been taken care by the Scrum master to co-ordinate with the team and assign the roles accordingly.	Coding was given preference and maintain good standards of programming.

Testing:

Key points	**Explore concepts significance and relevance**
● Software testing is done to uncover errors that were made during designing and construction. It includes test planning, test case design, test execution, data evaluation and collection. ● It involves alpha testing, beta testing, class testing. ● Project manager, software engineers, and testing specialists develop the strategy for software testing. ● Levels of testing: White Box and Black Box testing.	● Testing basically provides the developers the about how to conduct the tests, how we should develop a plan for testing, and when to involve a customer in the testing process. The testing process incorporates two phases which are as follows: the ***early testing*** that generally focuses on a small group or single component of related components and testing is done to detect the errors in the data and processing logic. Once the developer is through the early testing phase they proceed to ***later testing phase***. Here, a series of high-order tests are conducted to detect errors in assembling customer requirements. When errors are found, they must be corrected by a process, called debugging. ● Testing of the application often require more project physical and mental effort than any other SE action. If it is conducted irregularly, unnecessary effort is expended, time is wasted, and even sometime errors remain undetected. ● The main benefit of having a test specification document is that we can know the types of tests and specific testing steps that will be conducted. ● Testing is a way of checking if the application that we have developed is working properly and providing expected output.
Find Real life contexts	**Find Interdisciplinary connections**
● Karma, Jasmine, JS complexity analyzer are some of the tools used for testing. ● The tests can be non-functional or functional. Testing design techniques includes: Cause Effect Graphing, Boundary Value Analysis, Equivalence partitioning.	● There is no one particular way to test software. ● Not only we should understand various features of technology, but also we have to understand how that technology is applied as per different set of users and different perspectives.
Engage in critical thinking	**Technology, Tools, Techniques**

● Process of thinking is basically of two types : a) System 1 thinking b) System 2 thinking. ● **System 1**thinking is helpful for making quick decisions about effortless matters ● **System 2** thinking is helpful where software testers want to focus their efforts on. This can be termed as critical thinking. ● **Critical thinking** helps testers in remaining conscious about the software they're analyzing and when to start test runs and the nature of the defects they uncover. ● It suggests, until and unless testers have a firms understanding of overall mission, they shouldn't conduct test. ● It is the duty of testers and he shouldn't take anything for granted. It is the job of the testers to notice the defect or performance issue. By this the underlying problem can be noticed, otherwise which would have gone unnoticed.	● Karma, Jasmine and JS Complexity analyzer are some of the tools used in the testing process of this application. ● There are mainly 3 type of testing: white box testing (in this tester is completely aware of internal working of the app), black box testing(in this tester is unaware of internal working of the app), grey box testing(in this tester has limited knowledge about internal working).

Plan project management	**Project specification and project brief**
● Project plan management is one of the most important part. Inaccurate project planning can lead in degradation of the overall quality of the product as a whole. ● It includes distribution and allocation of tasks, Resource management, tools usage for reporting.	● Test plan reflects the entire project testing approach and schedule. ● Generally, a test plan includes tasks and objectives, testing strategy (White Box and Black box Testing), test schedule, dependencies, risks/assumptions, tools and approvals. These are the specifications for testing.

Software project management:

Key points	**Explore concepts, significance and relevance**
Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned,	● Cost estimation is accounted. ● Hiring of people, management of team, keeping the team encouraged. ● Analyzing the performance of the project.

implemented, monitored and controlled.	
Find real life contexts	**Find inter disciplinary connections**
● Software project management mainly focus on production aspects of software development, than the technical aspects, such as software tools. ● It helps in understanding progress reports for stakeholders, projection and time tracking, percentage complete and quality assurance. communicating updates with local and distributed items, feedback, assignments and shared task lists. Identify and resolve projects obstacles , appraise financials and evaluate performance.	● Introduces key risk-reduction models and the basic software development process . ● Explores recent swings in software process models and examines the subject of process improvement . ● Explain how software tools influence problem-solving and focus of development has shifted to business contexts from technical. ● Focuses on the economics and role of costs in software engineering.

Engage in critical thinking	**Technology , Tools and Techniques**
● New approaches in the specific field and knowledge of the advancement in the particular field is important. ● Understanding the Problem, Analysis and reducing the cost of the overall project.	Tools used in making this project are as follows: Slack Whatsapp Gmail GitHub

Plan project management	**Project specification and project brief**

● Although the software production is less costly as compared to the hardware production. But it do requires a much more rigorous development efforts, management. ● One of the major problems faced by every developer is the difference between the delivered software and the user specified software. ● To avoid these problems, software project management methods focused on matching user requirements to delivered products. This can be done by software models like waterfall model, spiral model and agile model. ● Analysis of software project management failures has shown that the following are the most common causes: involvement of insufficient end-user, Poor reporting of the project's status.	● Software project management is one of the important parts of planning and leading software projects. ● The main purpose of project planning is to recognize the scope of the project, estimate the work involved, and create a project schedule. ● Project planning starts with requirements that defines the required software to be developed. Then the project plan is developed to report the tasks that will lead to completion. ● This is the Controlling phase of management which helps the developer be on track. ●