**PRACTICAL-1**

**AIM**: List & draw at least 7 various Software Development Life Cycle (SDLC) models and preparing the detailed case study on “Microsoft Team” which SDLC model is suitable to develop the “Microsoft Team” desktop application?

**THEORY:**

Software development life cycle (SDLC) models show the ways to navigate through the complex and demanding process of software building. A project’s quality, timeframes, budget, and ability

to meet the stakeholders’ expectations largely depend on the chosen model.

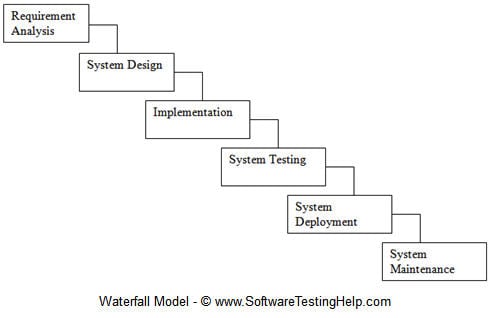
Today, there are more than **50** recognized SDLC models in use. None of them is perfect, and

each brings its favourable aspects and disadvantages for a specific software development project or a team.

**Types of SDLC Models:**

1. **WATERFALL MODEL:**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

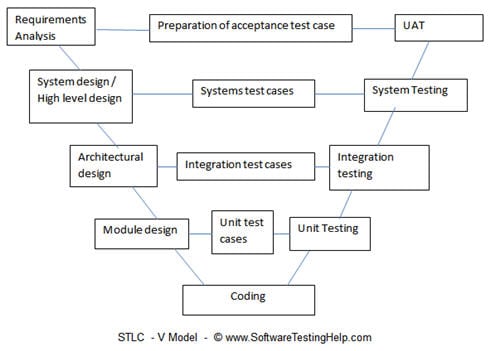


Use cases:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

1. **V-MODEL:**

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.



Use cases:

* Product definition is stable.
* Technology is not dynamic and is well understood by the project team.
* There are no ambiguous or undefined requirements.
* The project is short.

1. **INCREMENT MODEL:**

The development process based on the Incremental model is split into several iterations (“Lego-style” modular software design is required!). New software modules are added in each iteration with no or little change in earlier added modules. The development processes can go either sequentially or in parallel. Parallel development adds to the speed of delivery, while many repeated cycles of sequential development can make the project long and costly.



Use cases:

* Large, mission critical enterprise applications that preferably consist of loosely

coupled parts, such as microservices or web services.

1. **ITERATIVE MODEL:**

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

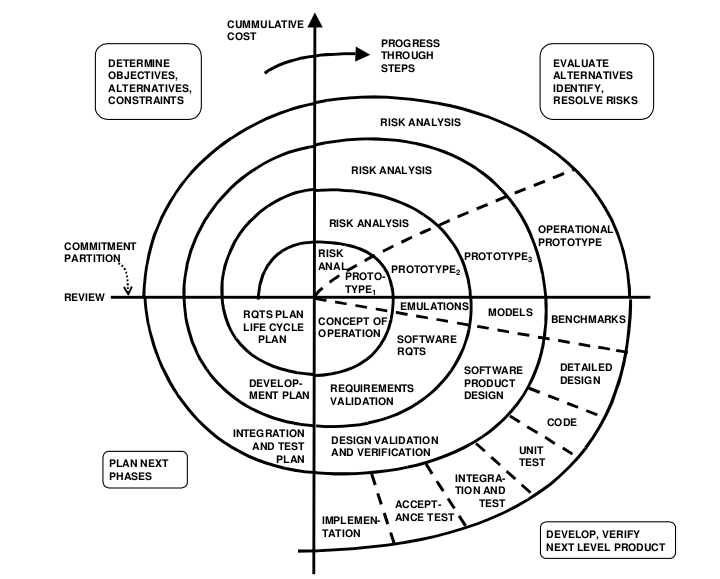


Use cases:

* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learnt by the development team while working on the project.
* Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
* There are some high-risk features and goals which may change in the future.

1. **SPIRAL MODEL:**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.



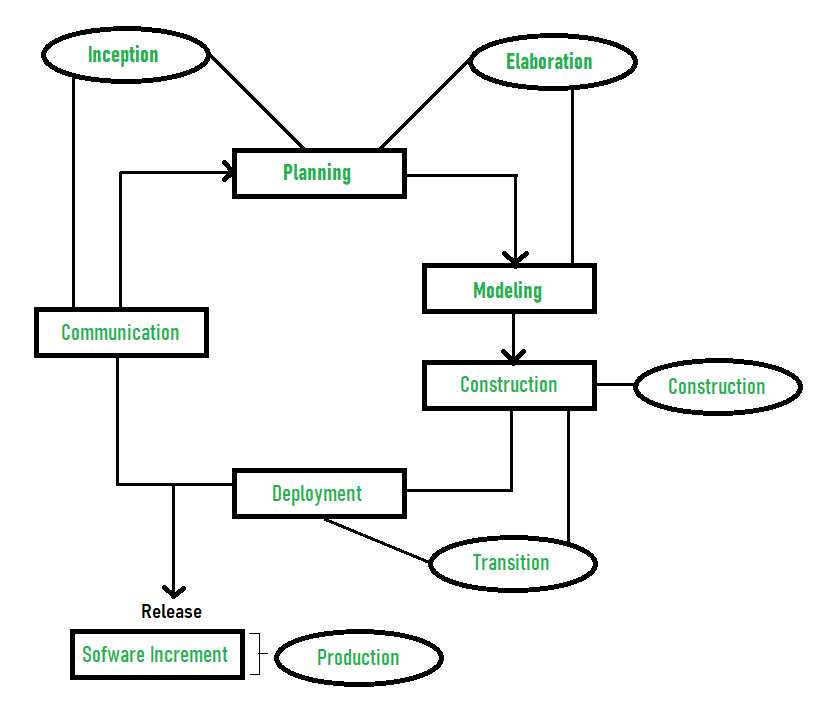
Use cases:

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which is usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

1. **RATIONAL UNIFIED PROCESS**:

The Rational Unified Process (RUP) is also a combination of linear and iterative frameworks. The model divides the software development process into 4 phases – inception, elaboration, construction, and transition. Each phase but Inception is usually done in several iterations. All basic activities (requirements, design, etc.) of the development process are done in parallel across these 4 RUP phases, though with different intensity.

RUP helps to build stable and, at the same time, flexible solutions, but still, this model is not as quick and adaptable as the pure Agile group (Scrum, Kanban, XP, etc.). The degree of customer involvement, documentation intensity, and iteration length may vary depending on the project needs.



Use cases:

* Large and high-risk projects, especially, use-case based development and fast

development of high-quality software.

1. **AGILE MODEL:**

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.



Use cases:

* Practically any start-up initiatives, when end user’s early feedback is required.
* Most mid-sized projects in custom software development were business
* requirements cannot be confidently translated to detailed software requirements.
* Large projects that are easy to divide into small functional parts and can be

developed incrementally over each iteration.

**CASE STUDY ON “MICROSOFT TEAMS”**

**Overview**:

With the ongoing pandemic of COVID-19, mostly all the organisations were left with no option

rather than adopting technologies to make sure the work is done without any barriers. Keeping in

mind Microsoft Teams a collaboration and communication software emerged as vital technology

to ensure not only the industrial sector but it went good with the Education purpose as well.

Microsoft Teams is a tool designed to transform the way people work and interact with each

other, with collaboration firmly in mind. Over the past three years, Teams has grown

significantly in both new capabilities and usage, as the hub for teamwork that brings people

together and fosters a culture of engagement and inclusion.

Starting September 2019, Devang Patel Institute of Advance Technology and Research migrated

to Microsoft Teams to continue the Teaching-Learning process. With this migration the initial

stages were a bit challenging but it went good with the time. It really helped in a few situations

when asked to give advanced features of file sharing and other collaborative things.

Solution Included:

* Provided custom role-based training in step with the migration
* Teams Essential + Conferencing + Telephony
* Teams Advanced
* Provided Virtual Sessions for remote employees
* Provided continuous learning

No. of End-Users:

* 1200+

Result:

* Successful end-user adoption of teams
* Higher user satisfaction

More examples:

* Air France
* Beiersdorf

**CONCLUSION**

From the above theory and case study, Agile Methodology suits best for the development of “Microsoft Teams” desktop application.