Software Engineering Assignment

MODULE: 1 (SDLC)

• What is software? What is software engineering?

Software is a program or set of programs containing instructions that provide desired functionality. And Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.

Software Engineering is the process of designing, developing, testing, and maintaining software. It is a systematic and disciplined approach to software development that aims to create high-quality, reliable, and maintainable software. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.

• Explain types of software

Application software. The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of [modern applications](https://www.techtarget.com/searchcio/feature/The-rise-of-modern-applications-Why-you-need-them) include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.

System software. These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the [firmware](https://www.techtarget.com/whatis/definition/firmware), computer language translators and system [utilities](https://www.techtarget.com/whatis/definition/utility).

Driver software. Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

Middleware. The term middleware describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

Programming software. Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and [debug](https://www.techtarget.com/searchsoftwarequality/definition/debugging) other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

• What is SDLC? Explain each phase of SDLC

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:

Requirement analysis

Planning

Software design such as architectural design

Software development

Testing

Deployment

This article will explain how SDLC works, dive deeper in each of the phases, and provide you with examples to get a better understanding of each phase.

1. Planning & Analysis

The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders. This phase is when you evaluate the feasibility of creating the product, revenue potential, the cost of production, the needs of the end-users, etc.

To properly decide what to make, what not to make, and what to make first, you can use a [feature prioritization framework](https://theproductmanager.com/topics/feature-prioritization/) that takes into account the value of the software/update, the cost, the time it takes to build, and other factors.

Once it is decided that the software project is in line with business and stakeholder goals, feasible to create, and addresses user needs, then you can move on to the next phase.

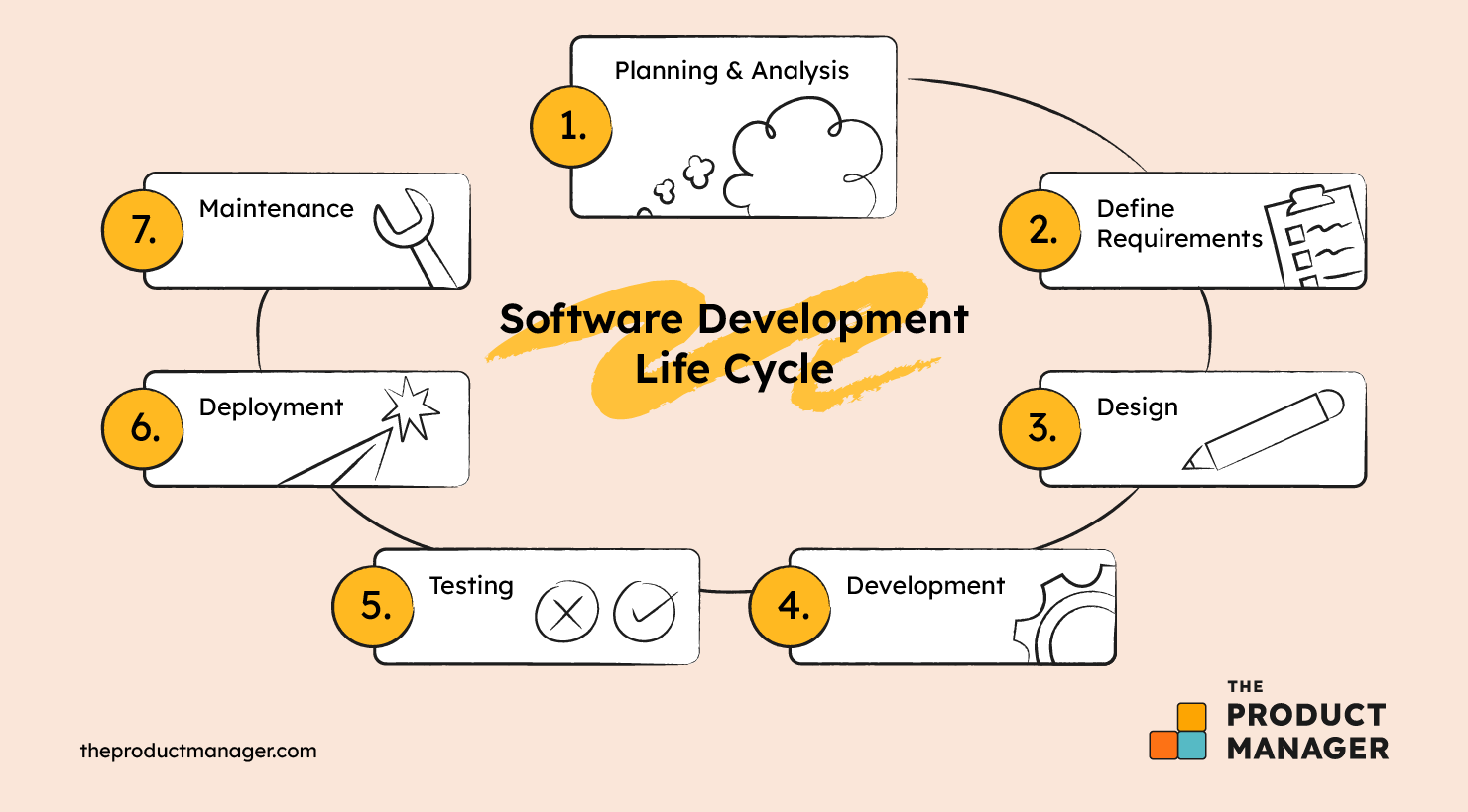
2. Define Requirements

This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team. This process guides the development of several important documents: a software requirement specification (SRS), a Use Case document, and a Requirement Traceability Matrix document.

3. Design

The design phase is where you put pen to paper—so to speak. The original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures. This is also where you can flowchart how the software responds to user actions.

In most cases, the design phase will include the [development of a prototype](https://theproductmanager.com/topics/prototyping-vs-wireframing/) model. Creating a pre-production version of the product can give the team the opportunity to visualize what the product will look like and make changes without having to go through the hassle of rewriting code.



4. Development

The actual development phase is where the development team members divide the project into software modules and turn the software requirement into code that makes the product.

This SDLC phase can take quite a lot of time and [specialized development tools](https://theproductmanager.com/tools/product-development-software/). It’s important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage.

In some cases, the development stage can also merge with the testing stage where certain tests are run to ensure there are no critical bugs.

Keep in mind, different [types of product development software](https://theproductmanager.com/topics/types-of-product-development-software-an-essential-guide/) will have different specialties so you’ll want to pick the one that suits you best.

5. Testing

Before getting the software product out the door to the production environment, it’s important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it’s meant to do. The testing process can also help hash out any major user experience issues and security issues.

In some cases, software testing can be done in a simulated environment. Other simpler tests can also be automated.

The types of testing to do in this phase:

Performance testing: Assesses the software's speed and scalability under different conditions

Functional testing: Verifies that the software meets the requirements

Security testing: Identifies potential vulnerabilities and weaknesses

Unit-testing: Tests individual units or components of the software

Usability testing: Evaluates the software's user interface and overall user experience

Acceptance testing: Also termed end-user testing, beta testing, application testing, or field testing, this is the final testing stage to test if the software product delivers on what it promises

6. Deployment

During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type. For example, if you are only deploying a feature update, you can do so with a small number of users (canary release). If you are creating brand-new software, you can learn more about the different [stages of the software release life cycle](https://theproductmanager.com/topics/software-release-life-cycle/) (SRLC).

7. Maintenance

The maintenance phase is the final stage of the SDLC if you’re following the [waterfall](https://theproductmanager.com/topics/agile-vs-waterfall-methodology/) structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

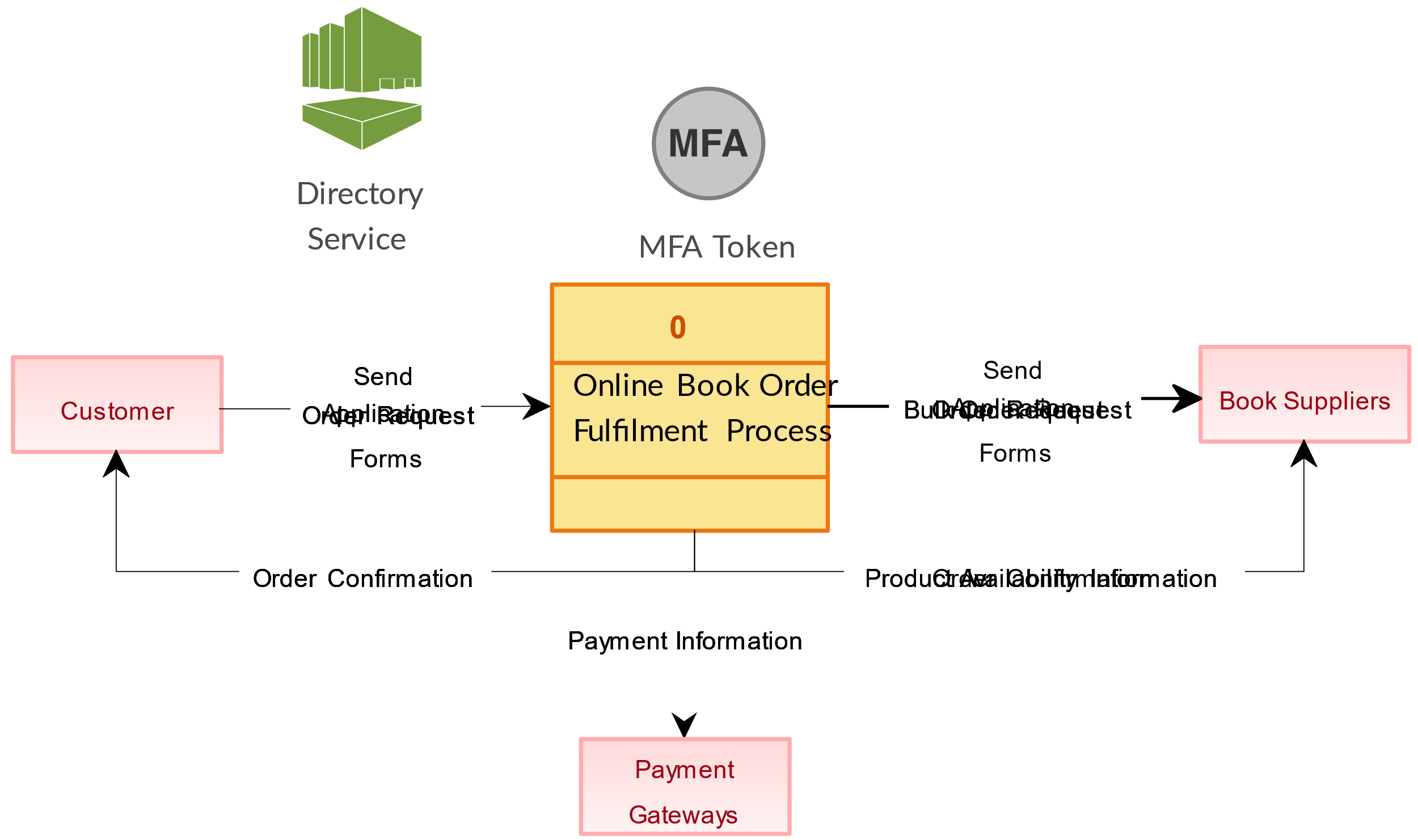
In the maintenance stage, users may find bugs and errors that were missed in the earlier testing phase. These bugs need to be fixed for better user experience and retention. In some cases, these can lead to going back to the first step of the software development life cycle.

The SDLC phases can also restart for any new features you may want to add in your next release/update.

• What is DFD? Create a DFD diagram on Flipkart

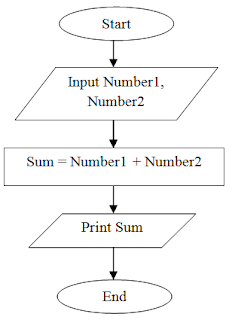
DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users ,managers and other personnel. it is useful for analyzing existing as well as proposed system.

DFD diagram on Flipkart :



• What is Flow chart? Create a flowchart to make addition of two numbers

Flowcharts are nothing but the graphical representation of the data or the algorithm for a better understanding of the code visually. It displays step-by-step solutions to a problem, algorithm, or process. It is a pictorial way of representing steps that are preferred by most beginner-level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. A flowchart is a picture of boxes that indicates the process flow sequentially. Since a flowchart is a pictorial representation of a process or algorithm, it’s easy to interpret and understand the process. To draw a flowchart, certain rules need to be followed which are followed by all professionals to draw a flowchart and are widely accepted all over the countries.



• What is Use case Diagram? Create a use-case on bill payment on paytm.

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

