****

**Business Programming**

Lesson 2: Software Development

**OVERVIEW**

In this handout, you will learn the concept of software development methodologies. You might have encountered the term before but may be not bothered to find its meaning. If you managed to find its meaning that it great. All the same, by the end of the lesson, you will have mastered software development methodologies. Individuals working for professional software development organizations find that it is a big part of their work environment. There is currently a trend of managing other types of work following the general scheme of a software development management practice.

GETTING ASSISTANCE

It is now a common practice to take you through to different sources in search of knowledge specific for the topic. In this lesson, I encourage you to maintain all the networks you have been using in the previous lesson unless if you were not finding enough joy from the sources then you can try others. Why break something if it works?

SO WHAT DO YOU NEED FOR THIS MODULE?

Basic knowledge to computer systems.

So let us get started

**Introduction**

A software development methodology is a way of managing a software development project. This typically address issues like selecting features for inclusion in the current version, when software will be released, who works on what, and what testing is done. No one methodology is best for all situations. Even the much maligned waterfall method is appropriate for some organizations. System methodologies keeps evolving just like computer programming / generations.

In practice, every organization implements their software development project management in a different way, which is often slightly different from one project to the next. None the less, nearly all are using some subset or combination of the ones discussed in this lesson. Choosing an appropriate management structure can make a big difference in achieving a successful end result when measured in terms of cost, meeting deadlines, client happiness, robustness of software, or minimizing expenditures on failed projects.

There are several terms which are often used interchangeably when it comes to business process such as:

* Systems vs Information systems
* Application Lifecycle Vs Software Development Lifecycle.

So, let us start by differentiating these terms as applied in business programming.

A System is set of related components that produces specific results and an Information system is a system that combines information technology, people, and data to support business requirements.

A system development methodology refers to the framework that is used to structure, plan and control the process of developing an information system. Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Systems development is mainly guided by the analyst and designer. Software development is guided by developers or analyst programmers. A methodology is, a set of steps, guidelines, activities and/or principles to follow in a particular situation. Most methodologies are comprehensive, multi-step approaches to systems development.

Another term which is often misused is application Application lifecycle management. (ALM) is the set of activities that revolve around a new software product, from its inception to when the product matures.

**Characteristics of systems**

1. System interact with other systems e.g. purchasing system generates orders that become input for a production system.

2. System have boundaries. Boundary indicates where one system ends and another system begins.

3. System handles specialized business needs. Specialized systems for information management that is unique to their company or industry, e.g. college handle class registration.

4. Systems requirements. The size and growth of an organization will determine the system requirement.

**List of different types of systems found in an organisation**

The current section discusses the possible types of systems found within an organisation. We are using the term systems at this point because these are used in developing the different information systems. Table 1 provides a list of some common systems. You are encouraged to master the current content to enable you to understand the type of systems you will be building.

**Table 1:Types of systems**

|  |  |
| --- | --- |
| Type of systems | Description |
| Enterprise computing systems | support company wide data management requirements, e.g. airline reservation and credit card billing systems. |
| Transaction processing systems | process data generated by day to day business operations, e.g. customer billing, accounts receivables and warranty claim processing. |
| Business support systems | provide job-related information support to users at all levels of a company, also called management information systems (MIS), analyze transactional data and generate information needed to manage and control business processes. |
| Knowledge management systems | These are also referred to as [also called expert systems, simulate human reasoning by combining a knowledge base and inference rules that determine how the knowledge is applied] |
| User productivity systems | User productivity systems provide employees at all organizational levels with a wide array of tools that can improve quality and job performance. |

There are a number of software development methodologies. In this lesson we are going to mention seven and these are:

* Waterfall
* Software Development Lifecycle (SDLC)- waterfall, V model
* Agile Methodologies-Scrum XP, FDD, DSDM
* Prototyping
* Incremental
* Spiral
* Rapid Application Development

For a solid understanding, we will start by explain the traditional methodologies.

**Software Development Life Cycle (SDLC)**

Software is a complex product that is delivered through a series of steps. Whether a document. diagram or working software, the artefact created in one step becomes the input to the next step. Eventually, the software is delivered to the customer. The sequence of steps used by these methods is commonly referred to as the software development lifecycle. (SDLC)

SDLC is a more formalized process for handling large projects where documentation, training, integrity, and security are vital to the project success. SDLC projects typically use object oriented analysis and design. Multiple models will be prepared for use cases, relational data, user interface, and a more abstract conceptual model. Multiple types of software testing are employed. It assumes that different management teams may be handling requirements, implementation, deployment, and monitoring. Table 2 provides the stages of SDLC.

|  |  |
| --- | --- |
| **Stages** | **Description** |
| **Planning and requirements** | to identify the scope of the new system, ensure that the project is feasible, develop a schedule, allocate resources and budget for the remainder of the project. |
| **Analysis** | to understand and document the business needs and the processing requirements of the new system |
| **Design** | In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived. |
| **Implementation or Coding** | Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase. |
| **Testing** | Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed. |
| **Deployment** | Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation. |
| **Maintenance and review** | After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers. |

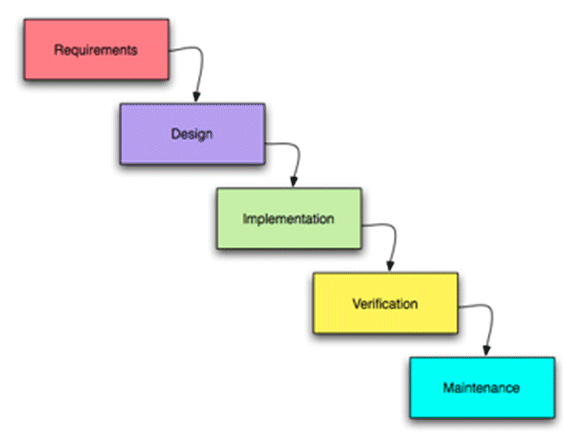
The common SDLC methods are the Waterfall and V-model.

Task1:

**List 5 advantages and 5 disadvantages of V-model methodologies**

**Waterfall methodology**

Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. 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 Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. Figure 1 illustrates the common stages of Waterfall methodology.



**Figure 1:Waterfall methodology**

As indicated in Figure 1, stages in waterfall can start while the previous stage is still halfway through. Let us briefly discuss each of the stages.

**Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

**System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

**Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing

**Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment of system**: Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

**Maintenance**: There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**Strength of Waterfall methodology**

1. It is ideal for supporting less experienced project teams and project managers or project teams whose members fluctuates.
2. The orderly sequence of development steps and strict controls for ensuring the adequacy and design reviews helps ensure the quality, reliability, and maintainability of the developed software.
3. Progress of system development is measurable
4. Conserves resources

**Weaknesses of the waterfall methodology**

1. Project progresses forward with only slight movement backwards.
2. Little room for use of iteration, which can reduce manageability if used.
3. Depends upon early identification and specification of requirements, yet users may not be able to clearly define what they need early in the project.
4. Requirements inconsistencies, missing system components and unexpected development needs are often discovered during design and coding.
5. Problems are often discovered until system testing.

**Prototyping**

A prototype is a system that captures the essential features of a later system, is the most appropriate definition of a prototype. A prototype system, intentionally incomplete, is to be modified, supplemented, or supplanted.

**Task1:**

**List 5 advantages and 5 disadvantages of the prototyping methodologies.**

**Agile Methodologies**

Agile methods are meant to adapt to changing requirements, minimize development costs, and still give reasonable quality software. Agile projects are characterized by many incremental releases each generated in a very short period of time. Typically, all members of the team are involved in all aspects of planning, implementation, and testing. This is typically used by small teams, perhaps nine or fewer, who can have daily face-to-face interaction. Teams may include a client representative. There is a strong emphasis on testing as software is written. The disadvantages of the Agile methods are that they work poorly for projects with hundreds of developers, or lasting decades, or where the requirements emphasize rigorous documentation and well documented design and testing.

The different agile technologies include:

* Xtreme Programming (XP)
* SCRUM
* Agile Software Development (ASD)
* Feature-Driven Development(FDD)
* Crystal Methods
* Dynamic Systems Development Method (DSDM)

The above listed techniques are discussed in detail in the next lesson in detail