

Lecture-1

Introduction

-Nafisa Tabassum
Lecturer, Dept. of CSE
NDUB

What is software?

- Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
- Good software should deliver the required functionality and performance to the user, should be maintainable, dependable and usable and ease of use should be high

What is software engineering?

- Software engineering is an engineering discipline that is concerned with all aspects of software production.

N.B. Engineering means the branch of science and technology concerned with the design, building, and use of engines, machines, and structures.

Software Product

Two kinds of SW products (SW which can be sold to a customer)

- Generic products
 - Stand-alone systems that are marketed and sold to any customer who wishes to buy them. The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
 - Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

Software Product

- Customized products
 - Software that is commissioned by a specific customer to meet their own needs. The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.
 - Examples – embedded control systems, air traffic control software, traffic monitoring systems.
- ★ Many systems are now being built with a generic product as a base, which is then adapted to suit the requirements of a customer. E.g., Enterprise Resource Planning (ERP) – e.g., Systems Applications Products (SAP)

Importance of software engineering

- More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

General issues that affect software

- Heterogeneity
 - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
- Business and social change
 - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

General issues that affect software

- Security and trust
 - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.
- Scale
 - Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

Software Application Domains

- **System software**—a collection of programs written to service other programs. Some system software (e.g., compilers, editors, and file management utilities) processes complex, but determinate, information structures. Other systems applications (e.g., operating system components, drivers, networking software, telecommunications processors) process largely indeterminate data. In either case, the systems software area is characterized by heavy interaction with computer hardware; heavy usage by multiple users; concurrent operation that requires scheduling, resource sharing, and sophisticated process management; complex data structures; and multiple external interfaces.

Software Application Domains

- **Application software**—stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making. In addition to conventional data processing applications, application software is used to control business functions in real time (e.g., point-of-sale transaction processing, real-time manufacturing process control).
- **Engineering/scientific software**- has been characterized by “number crunching” algorithms. Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing. However, modern applications within the engineering/scientific area are moving away from conventional numerical algorithms. Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics.

Software Application Domains

- **Embedded software**—resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system. E.g., SW in a microwave oven to control the cooking process
- **Product-line software**—designed to provide a specific capability for use by many different customers. Product-line software can focus on a limited and esoteric marketplace (e.g., inventory control products) or address mass consumer markets (e.g., word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, and personal and business financial applications).

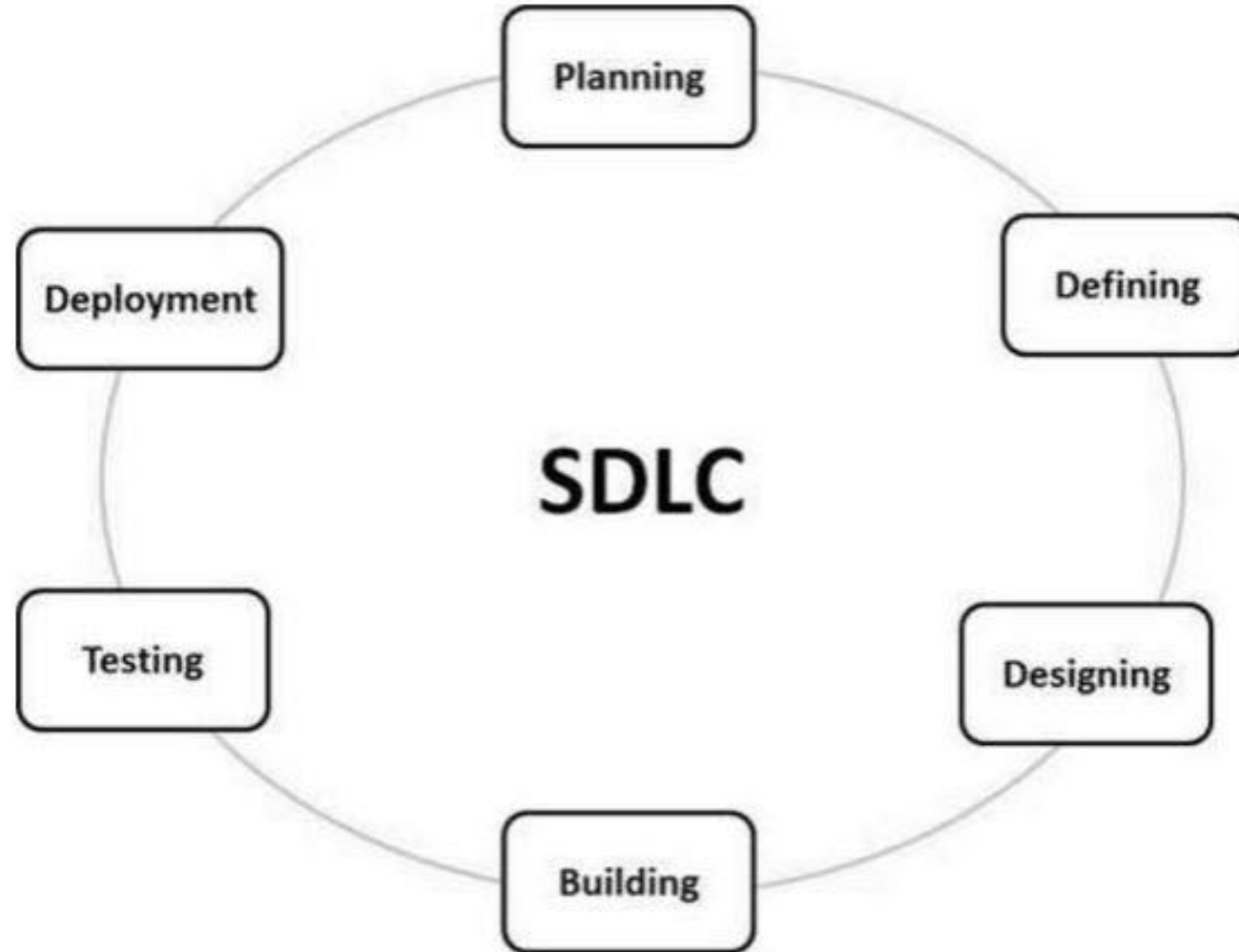
Software Application Domains

- **Web applications**—called “Web Apps,” this network-centric software category spans a wide array of applications. In their simplest form, Web Apps can be little more than a set of linked hypertext files that present information using text and limited graphics. However, as Web 2.0 emerges, Web Apps are evolving into sophisticated computing environments that not only provide stand-alone features, computing functions, and content to the end user, but also are integrated with corporate databases and business applications.
- **Artificial intelligence software**—makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis. Applications within this area include robotics, expert systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing.

What is SDLC?

- Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



- **Stage 1: Planning and Requirement Analysis:** Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.
- **Stage 2: Defining Requirements** Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

- **Stage 3: Designing the Product Architecture:** SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

- **Stage 4: Building or Developing the Product:** In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

- **Stage 5: Testing the Product:** This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.
- **Stage 6: Deployment in the Market and Maintenance:** Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing). Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.