**LECTURE 2**

**SYSTEMATIC DEVELOPMENT**

Systematic development refers to a structured and organized approach to developing a product, system, or software. It involves following a series of predefined steps and processes to ensure efficient and effective development while meeting the desired objectives.

**SYSTEMS DEVELOPMENT LIFE CYCLE**

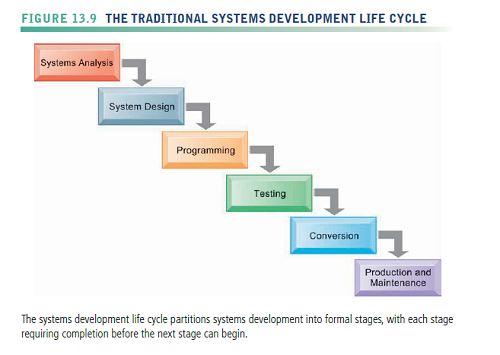
The System Development Life Cycle (SDLC) is a systematic approach to developing and managing information systems. It consists of a series of well-defined phases or stages that guide the development process from initial concept to final implementation and maintenance. The SDLC provides a framework for structuring, planning, and controlling the development of a system while ensuring that it meets user requirements and quality standards.

* For explanation purpose, we will consider that, in essence, there are two approaches to systems development
  + The pre-specified approach
  + The evolutionary approach

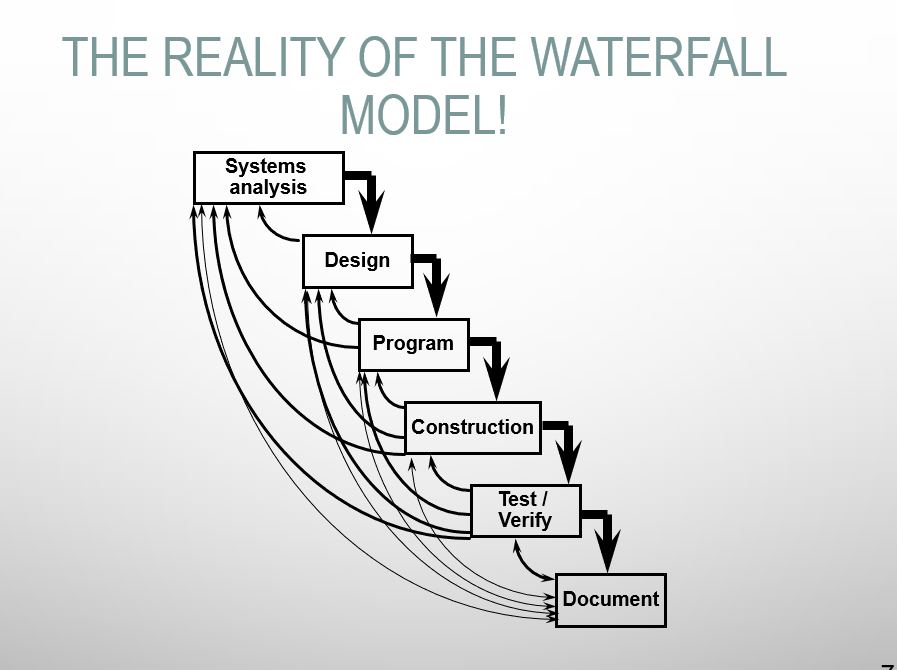
**PRE – SPECIFIED APPROACH**

In systems development, a pre-specified approach refers to the predetermined plan and set of activities established before designing, building, and implementing a software system or application. It involves defining the development methodologies, frameworks, tools, and techniques to be used throughout the project lifecycle.

**PRE-SPECIFIED LIFE CYCLE**

****

The systems life cycle is still used for building large complex systems that require a rigorous and formal requirements analysis, predefined specifications, and tight controls over the system-building process. However, the systems life cycle approach can be costly, time-consuming, and inflexible. Although systems builders can go back and forth among stages in the life cycle, the systems life cycle is predominantly a “waterfall” approach in which tasks in one stage are completed before work for the next stage begins. Activities can be repeated, but volumes of new documents must be generated and steps retraced if requirements and specifications need to be revised. This encourages freezing of specifications relatively early in the development process. The life cycle approach is also not suitable for many small desktop systems, which tend to be less structured and more individualized.



* There is no rule that says there must be six phases. Neither are terms such as ‘design’ or ‘construction’ used consistently.
* For example, a three-phase lifecycle might be:
  + Analysis
  + Design
  + Construction
* But here, construction would have to include programming, testing, integrating all the programs together, documentation, user training and so on.

**EVOLUTIONARY APPROACH**

In the evolutionary approach, the view is taken that it is better to gradually develop an information system over time, thus allowing both users and systems analysts time to come to terms with what the requirements of the information system really are.

The evolutionary approach, also known as the iterative or incremental approach, is a software development methodology that focuses on gradually refining and improving a software system through multiple iterations or increments. It involves the iterative development of the system, where each iteration builds upon the previous one, adding new functionality or improving existing features.

1. Agile Development: Agile methodologies, such as Scrum, Kanban, and Extreme Programming (XP), are popular variants of the evolutionary approach. These methodologies emphasize iterative and incremental development, frequent collaboration with stakeholders, continuous feedback, and adaptability to changing requirements.
2. Lean Development: Lean development, inspired by the principles of Lean manufacturing, focuses on maximizing customer value while minimizing waste. It emphasizes continuous improvement, waste reduction, and the elimination of non-value-adding activities. Lean development promotes iterative and incremental development cycles, with a strong emphasis on customer involvement and quick feedback loops.
3. Spiral Model: The Spiral Model is a risk-driven iterative model that combines elements of both waterfall and iterative approaches. It involves repeatedly iterating through four key activities: determining objectives, identifying and resolving risks, developing and testing the software, and evaluating the results. The Spiral Model allows for iterative refinement of the software based on risk analysis.
4. Rapid Application Development (RAD): RAD is a variant of the evolutionary approach that focuses on rapid prototyping, quick development, and continuous user involvement. RAD emphasizes building prototypes or mock-ups early in the development process to gather user feedback. It typically involves a series of iterations that incrementally refine the software based on user input.
5. Prototyping: Prototyping involves creating a basic working model of the software system to gather user feedback and refine requirements. It allows for rapid exploration of design ideas and validation of concepts. Prototyping can be used as a standalone approach or as part of the evolutionary approach to refine the software system incrementally based on user feedback.
6. DevOps: DevOps is an approach that emphasizes collaboration and integration between development and operations teams. It aims to automate the software delivery process, ensuring faster and more reliable deployment of software updates. DevOps aligns with the evolutionary approach by facilitating continuous integration, deployment, and feedback loops for iterative development and improvement.

**PROTOTYPING**

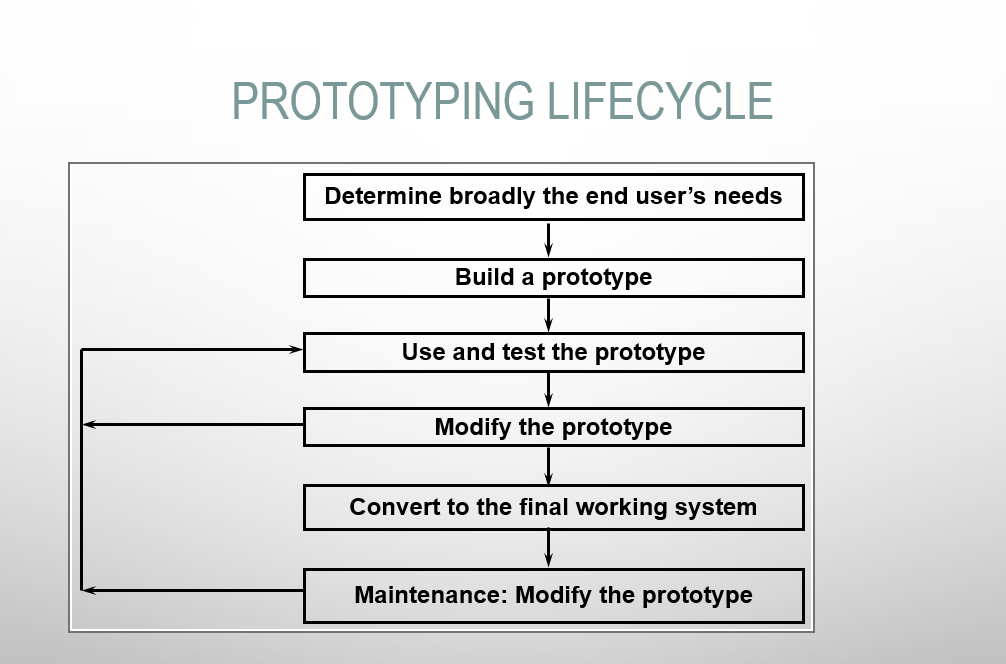
Prototyping consists of building an experimental system rapidly and inexpensively for end users to evaluate. By interacting with the prototype, users can get a better idea of their information requirements. The prototype endorsed by the users can be used as a template to create the final system. The prototype is a working version of an information system or part of the system, but it is meant to be only a preliminary model. Once operational, the prototype will be further refined until it conforms precisely to users’ requirements. Once the design has been finalized, the prototype can be converted to a polished production system.

The Prototype model is one of the software development life cycle models in which a prototype is built with minimal requirements. This prototype is then tested and modified based on the feedback received from the client until a final prototype with desired functionalities gets created. This final prototype also acts as a base for the final product.

**Phases of Prototype Model**

The following are the primary phases involved in the development cycle of any prototype model.

* **Initial Communication** – In this phase, business analysts and other individuals responsible for collecting the requirements and discussing the need for the product, meet the stakeholders or clients.
* **Quick Plan** – Once basic requirements have been discussed, a quick plan of the initial prototype is made.
* **Modeling Quick Design** – User interface part i.e. designing part of the prototype is carried out in this phase.
* **Development of the Prototype** – In this phase, the designed prototype is coded and developed.
* **Deployment, Delivery, and Feedback of the Prototype** – In this phase, the initial prototype is deployed and is accessible to clients for its use. Clients review or evaluate the prototype and they provide their feedback to the requirements gathering and development teams.  
  Above mentioned phases keep repeating until the replica of the final product is deployed.
* **Final Product Design, Implementation, Testing, Deployment, and Maintenance** – Once the client finalizes a prototype, on the basis of the prototype, the final product is designed and developed. This developed product is tested by the testing team and if it is ready to go LIVE, the product is deployed and is available for end-user.

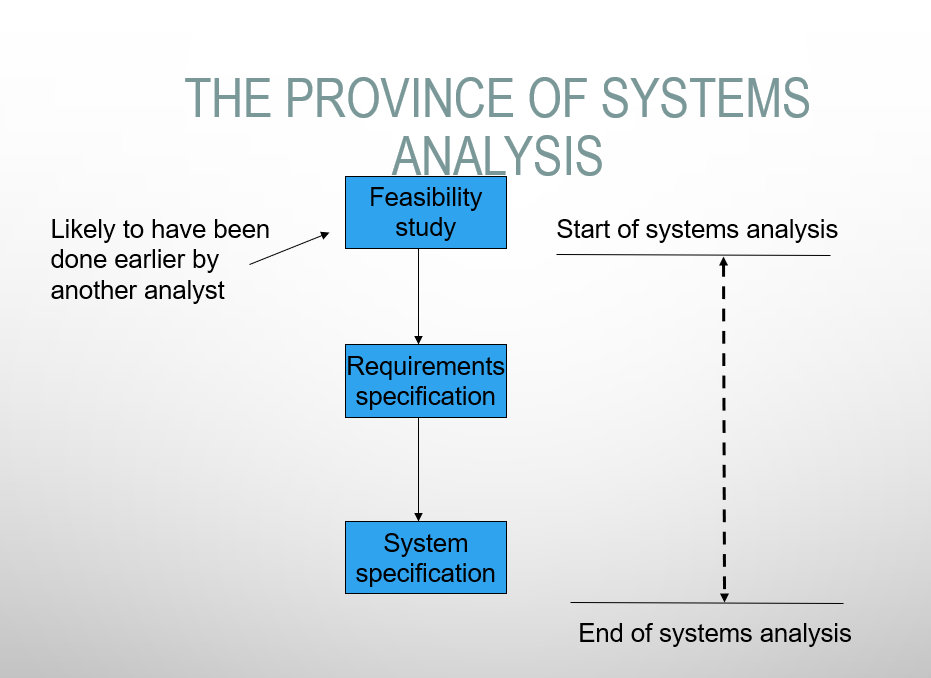


**THE REALITY OF SYSTEM DEVELOPMENT**

* The reality of systems development is often that a blend of the approaches is taken. For example,
  + Prototyping can be used (and often is) within the traditional waterfall approach
  + In the evolutionary approach a degree of pre-specification is necessary to get started
  + Where do you think agile approaches, such as extreme programming, fit here?

**METHODOLOGY**

**BENEFITS OF USING METHODOLOGY**



**THE SPECIFICATION**

If there is an existing system (either manual or computerised) the analyst can examine this, perhaps interview people and establish its problems. By modelling the existing system (using certain diagrams) and examining its problems, possible solutions can be identified and if agreed, implemented in a new computer system.

If there is no existing information system, then the analyst has to create one from scratch. Again, it will probably be necessary to model the likely information system using diagrams or possibly prototype it and get the new users to agree that this is what is required.