Chapter III: System Analysis methods

1. System Analysis overview

-It is a set of activities to identify the requirements of the system.
-Business systems usually complex.
-Without doing analysis, you may create a software solution that solves wrong problem and the result is the loss of money and time.
-System analysis plays the most important role in success or failure. of an information system.
-The job of system analyst is to understand the specific requirements that must be achieved to high information system.
-The system analysis must define what the system does and what the system needs.

Chapter III: System Analysis methods

1.1 System Analysis Methods

-To analyze the system effectively, methods are required. There are a a lot of system analysis methods that differ in steps and processes to to analyze the system.

-The popular methods used by system analysts are structured analysis method, prototype, and object-oriented analysis.

-Good methods must include a way to determine system requirements clearly.

Chapter III: System Analysis methods

1.2 System Development Life Cycle (SDLC)

-A model defines the ordered stages in building an information System.

-The popular system development methods developed to guide the processes of building information systems include Waterfall model (Original SDLC), Rapid Application Development (RAD), and Joint Application Development (JAD) .

Chapter III: System Analysis methods

1.3 Structured Analysis Method

-Structured analysis method defines clear system requirements that both users and system developers can understand.

-It breaks down a big project in to small and well-defined activities.

-Sequences and interactions between the activities are also defined.

-Various graphical models are used to gain better understanding of system and its requirements.

Chapter III: System Analysis methods

1.4 System Prototype Method

-Prototypes are created to present only a few aspects of the software.

-Prototypes are incomplete version of the software being developed.

-Software designers and programmers can receive feedback from their customer as soon as possible at the early state of software development.

-Software developers can estimate the accuracy of initial project and determine deadline and milestones can be met.

-Modifying the prototype is not very expensive as to final product.

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

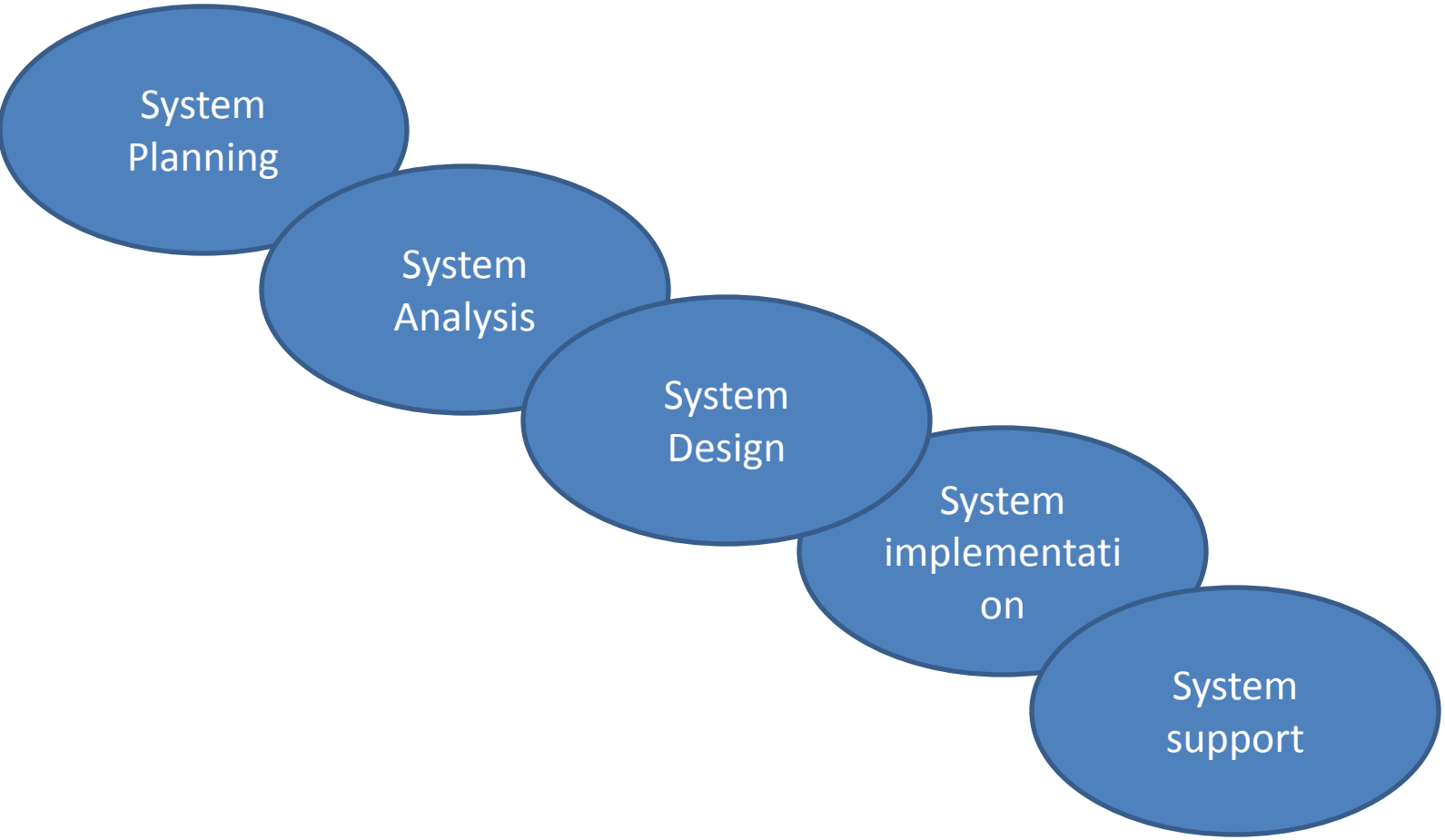
-The software project planning involves the estimate how much money, time, and other resources required to complete to build a specific software.

-Software project managers will use the historical data (software metrics) and experience to guide the estimates.

-The estimates can be affected mostly by the two factors: software complexity and project size.

-The planning objective is achieved through a process of

System Analysis & Design



System Development Lice Cycle (SDLC)

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

information discovery. The activities to perform during the the system planning are 1. determination of software scope 2. Resources.

-The determination of software scope produces a scope statements about software data, functions, performance, and constraints.

-For example, in planning an ATM software project, we define

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

the software scope as the following:

-Data:

+input data: user authentication information (user name PIN code), and transaction types (balance inquiry, withdrawal, and deposit)

+output data: transaction type, date, and time, terminal, card number, account number, and current balance

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

the software scope as the following:

-Functions:

+Read user information

+Authenticate user information

+Perform transactions (inquiry, withdrawal, deposit)

++inquiry: get available balance, get total balance

++withdrawal: display menu of withdrawal amounts,

cancel option

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

++withdrawal: display menu of withdrawal amounts, cancel option, get available balance, test sufficient case, display error message, interact with database to update account, etc.

++deposit: prompt user to enter deposit amount, receive input, instruct user to insert envelope, display error message, interact with database to credit account, etc.

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.1. System planning

-Performance: time to complete a transaction, a process, respond time, time to switch between one transaction to Another, etc.

-Constraints: available memory, a must to build software with C++ and Mysql database, etc.

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.2. System Analysis

-Understand the specific requirements that must be achieved to build a high-quality software.

-System analysis will describe data, functions, performance, and constraints of the system in detail.

-Requirements are refined, and analyzed to ensure their accuracy, clarity, and completeness.

-Various models are produced in stages including data model, and process model.

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.2. System design

-The goal of system design is to generate a model or presentation of the system to be built.

-Hardware requirements of the system , programming language , database structure are also stated.

-In this phase, database , program modules, and interfaces are designed.

-The models produced in system analysis are basis to the system design phase.

Chapter III: System Analysis methods

2. System Development Life Cycle (SDLC)

2.3. System implementation

-The design is translated into a machine-readable form.

-Code generation step performs this task.

-Testing the program is also conducted in this phase.

2.3. System support

-The changes will made to the system.

-Changes will occur because of errors found, because of adapting to external environment or requirement of customer

to enhance software performance.

Chapter III: System Analysis methods

3. Structured System Development Life Cycle (SSDC)

-The Waterfall model or called Classical Life Cycle is a structured system analysis and design that contains the five stages process as mentioned above.

-Each stage produces its own result or end that can flow to the next stage.

-They might see the disadvantages of the SSDC as followings:

+Real project rarely follow the sequential flow that the model processes.

+It is almost impossible for customer to describe all requirements.

Chapter III: System Analysis methods

3. Structured System Development Life Cycle (SSDC)

+Too late to get feedback from customer. The working

production is not available until late in the project time span.

The major problems are not discovered until the working

product is reviewed.

Chapter III: System Analysis methods

4. Prototyping

In prototyping a software, various tools are used.

The prototyping tools should have the following characteristics:

+Support component-base development (e.g. re-usable objects),

+Allow prototype changes to be made quickly and easily,

+Support for debugging and testing,

Chapter III: System Analysis methods

4. Prototyping

+Provide visual ant interactive programming environment (e.g. Syntax for functions, drop-down list for classes, properties, and methods)

+Support database design and database access, and

+Support form layouts and widgets.

+Easy to use

Chapter III: System Analysis methods

5. Types of Prototyping Tools

5.1. Application Generators

-Application Generator tools are used to generate user interfaces, and define functions or components of system.

-Examples of Application Generators are Lumzy, Pencil, Prototype Composer, DICE, SCT, etc.

5.2. Query and Retrieval Languages

-DDL (Data Definition Language) and DML (Data Manipulation are suitable for prototyping tools.

-DDL is used for data definition (create, delete, modify database, table, query/view) while DML is used for data manipulation (insert, delete, update, retrieve data).

Chapter III: System Analysis methods

5. Types of Prototyping Tools

5.3. Computer Assisted Software Engineering (CASE) tools

-CASE refers to the methods for the development of software with the help of automatic tools that can be used in system development processes.

-CASE tools are software used to automate many activities in software development.

-Designers use CASE tools to generate design documents from prototype d functional requirements.

-Programmers used CASE tools to transform design documents to code.

-CASE tools can increase productivity of analysts and reduce time span in the project.

Chapter III: System Analysis methods

5. Types of Prototyping Tools

5.3. Computer Assisted Software Engineering (CASE) tools

+SmartDraw: It is for general purpose of drawings include FlowChart, Organization Chart, ER-diagrams, etc.

+IBMS/DFD: This software tool is for drawing Data Flow Diagram (DFD).

+ARM: This tool can help you to write system requirement specifications.

Chapter III: System Analysis methods

5. Types of Prototyping Tools

5.3. Computer Assisted Software Engineering (CASE) tools

-Examples of CASE tools:

+VP-UML: this tool is used to draw all kinds of diagrams for object-oriented analysis and design, Requirement diagrams, ER-diagrams, etc.

+Visual Basic: It is a tool for interface design.

Chapter III: System Analysis methods

6. Advantages of the Prototyping Approach

-Improves quality of requirements and specifications.

-The clear determination of what the user want results in fast software building.

-By examining the prototype, it increases users involvements and interactions, and prevents misunderstandings between users and development team. So, the quality of the final product is more likely to satisfy the user requirements.

Chapter III: System Analysis methods

5.5. Disadvantages of the Prototyping Approach

-By focusing on the limited prototype, the developers may not properly analyze the complete system. Thus, the better solution may be overlooked. This may lead to poor quality final product that is hard to maintained.

Chapter III: System Analysis methods

7. Disadvantages of the Prototyping Approach

-The prototype is to get working quickly. If the developers lose sight of this and they try to develop a complex prototype, the excessive development time of the prototype will be.

-The developers may attempt to convert the limited prototype in to final product because they spent much effort to build it and they are already familiar with some inefficient algorithms or inappropriate programming language that were used to build prototype.