Chapter 17: Prototyping and Feasibility Study

Prototyping

Problems with Classical SDLC

- The problems with classical SDLC that prototyping approach considered inexpensive and quickest way in building a system
  - User Requirements Change
  - Incomplete User Requirements
  - Development Targets are Missed

Problems with Classical SDLC – User Requirements Change

- Systems often take years to analyze, design, and implement.
- During development period, the environment and user needs is are constantly changing.
- The finished system might not meet the new applications environment.
- With the traditional SLDC approach, it is relatively difficult to respond to new needs once users have signed off on the requirements analysis and the design.

Problems with Classical SDLC – Incomplete User Requirements

- People often cannot articulate what type of information they really need in their work. Frequently, it is not until people begin working with a system that these needs become apparent to them. With the traditional SDLC approach, users do not really see the new system until it is installed.
- Users have a difficult time visualizing the system by looking at a set of diagrams or specifications. In many projects, requirements analysis had to be repeated because additional requirements were discovered or because users changed their minds.

Problems with Classical SDLC – Development Targets are Missed

- Many business managers are not over-enthusiastic about traditional systems development in their organizations. While the systems that ultimately emerge are most of the time successful, the process of completing them is often fraught with disappointing delays and unpleasant surprise.
- Systems developers are often notoriously optimistic. Deadlines are often set and missed.
- The systems usually require more resources than originally estimated and costs frequently climb over budget.
- Problems of miscommunication, omissions, and human Error complicate matters. Projects are often cancelled as cost begin to exceed benefits and the probability of ever finishing them within a reasonable span of time becomes more and more remote.

Prototyping

- **Prototyping** is the *process of quickly putting together a working in order to test various aspects of the design, illustrate ideas or features and gather early user feedback*.
- This is the process of building an experimental system quickly and inexpensively for demonstration and evaluation so that users can better determine information requirements.
- By interacting with the prototype, users can get a better idea of their information requirements.

The Prototyping Process

- The main steps of a prototyping process are:
  - Phase 1:
    - Define System Specifications.
  - Phase 2:
    - Creating the Prototype System.
  - Phase 3:
    - Refine the Prototype System.
  - Phase 4:
    - Develop Operational System.

Phase 1: Define System Specifications

- One of the rough spots in the pre-specification approach to systems development has been the definition of system specifications.

Phase 2: Creating the Prototype System

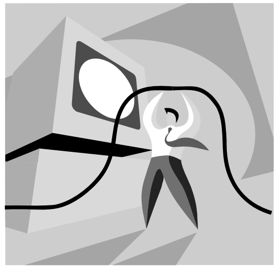
- Project team members rough out the logic of the system and how the elements fit together and then work with the user to define the I/O interfaces.

Phase 3: Refine the Prototype System

- Users sit down at a terminal or PC and evaluate portions and eventually all of the prototype system.
- The system is then expanded and refined to meet the users' total information needs.

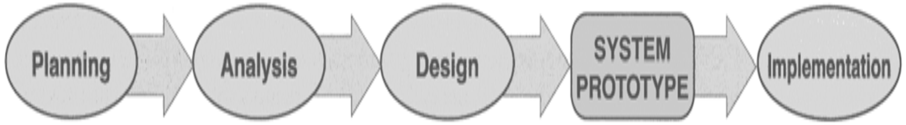
Phase 4: Develop Operational System

- The prototype system is custom coded.



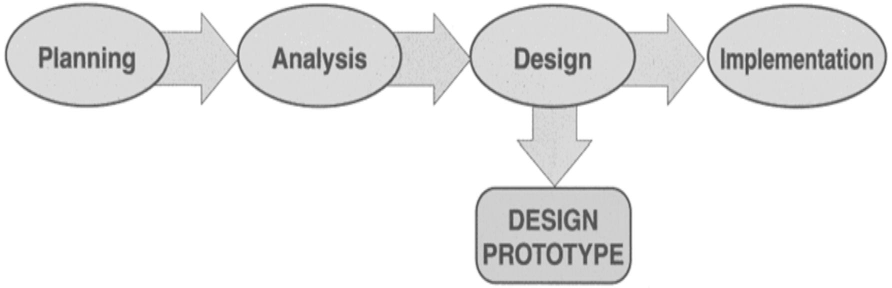
System Prototyping

- Prototyping involves a repetitive sequence of analysis, design, modeling and testing.
- The end product of **System Prototyping** is a full-featured, working model of the information system, ready for implementation.



Design Prototyping

- Design Prototyping is used to verify user requirements.
- The end product of **Design Prototyping** is a user-approved design prototype that documents and benchmarks the features of the finished system.



Advantages of Prototyping

- Sound Development in the Final Product
  - Most useful when there is uncertainty about requirements or design solutions.
- Development experience
  - Valuable for the design of End-user interface of an Information System.
  - Prototyping encourages intense end-user involvement throughout the systems development lifecycle.
- Greater user satisfaction,
  - It is more likely to produce systems that fulfill user requirements.

Disadvantages of Prototyping

- Lengthens the application development process
  - The rapid pace of development can create quality problems, which are not discovered until the finished system is operational.
  - Other system requirements, such as reliability and maintainability, cannot adequately be tested using a prototype.
- Inability to relinquish the use of the prototype
  - In very complex systems, the prototype becomes unwieldy and difficult to manage

Conditions for Prototyping Realization

- The size of the project team must be as small as possible.
- Users must be committed towards providing constructive feedback.
- The system should be designed and built in a way that makes it easy to modify.

Deployment of Prototyping in Application Development Process

- It can be used in either all or part of the development of a given system.
- As users work with the initial prototype, they refine it and create and second version. This process is repeated until a satisfactory version is created.
- The knowledge learned in creating the prototype is then used in the construction of the real system.

Prototyping and Feasibility Study

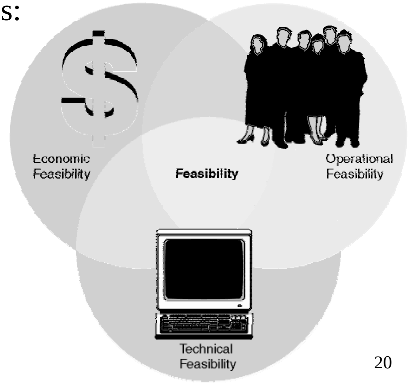
Feasibility Study

What is a Feasibility Study?

- The **Feasibility Study** is one of the steps within the **Preliminary Investigation Phase** of the Systems Development Life Cycle (SLDC).
- The analyst learns from the user what is expected of the proposed information system.
- The analyst uses technical skills and knowledge to decide on various alternatives to be considered for implementation.
- The feasibility study has to contain enough detail so that an alternative can be selected for development.

Types of Feasibility

- Different types of feasibility that must be evaluated in order to determine if the project is worth doing.
- A feasibility study uses three major yardsticks to measure, or predict a system's success:
  - Economic Feasibility
  - Technical Feasibility
  - Operational Feasibility



Operational Feasibility

- A system that has operational feasibility is one that will be used effectively after it has been developed.

Technical Feasibility

- A systems request has technical feasibility if the organization has the resources to develop or purchase, install, and operate the system.
- Project risk can be assessed based upon:
  - Project size
  - Project structure
  - Development group’s experience with the application
  - User group’s experience with development projects and the application area

Economic Feasibility

- A systems request has economic feasibility if the projected benefits of the proposed system outweigh the estimated costs involved in acquiring, installing, and operating it.
- Before management will approve any large computer system, an appraisal of the benefits and costs of the system must be done.
- Essentially what happens is that all costs associated with the proposed systems project is calculated, as well as the benefits to be derived from this system.
- Then we use certain financial methods to rank projects.

What is Costs?

- Among the cost items that are taken into consideration are hardware, software, personnel, and establishment of new procedures and representations of data.
- Cost is divided into several types:
  - Tangible Costs (can easily be measured in dollars)
    - Example: Hardware
  - Intangible Costs (cannot be easily measured in dollars)
    - Examples: loss of customer goodwill, loss of employee morale
  - One-Time Costs (project startup, initiation and development)
    - Example: System Development, Data Conversion
  - Recurring Costs (ongoing use of the system)
    - Example: Software Maintenance

What is Benefits?

- The benefits are divided into **Tangible Benefits** and **Intangible Benefits**.
  - Tangible Benefits are those that can be measured easily (most readily translated into dollars).
    - Examples: Cost reduction, Opening new markets and increasing sales opportunities
  - Intangible Benefits are those that cannot be measured easily (difficult to quantify)
    - Examples: Increased employee morale, Promotion of organizational learning and understanding

What is a Requirement?

- A **Requirement** is a feature that users identify as being necessary or useful in a system.
- How are requirements determined?
  - The business must first be understood.
  - The problems are studied and opportunities for improvement are identified.
  - It is these opportunities that are eventually turned into statements of requirements.

Importance of Requirements

- The requirements stage is one of the early stages in SDLC.
- By studying both the existing and proposed systems in detail, various problems can be anticipated.
- It is important to consider how these problems can be solved or at least, how to minimize their impact.
- It is important that requirements are clear and complete.
- Subtle misunderstandings may result in defects which could be hard to detect.

Methods of Data Collection

- **Primary Data Collection** comprises direct interaction with the source of data
  - Interview
  - Observation
  - Questionnaire
- **Secondary Data Collection** are collected from previous year’s data or alternative source (but not direct from source)
  - Documents Review

Interview

- The systems analyst spends a great deal of time interacting with others, particular in interviews settings.
- Interviews have varying degrees of structure; for a first meeting there may be no structure at all.
- As the project progresses, more structured interviews are conducted.
- The analyst may wish to prepare in advance an interview schedule containing the questions to be asked and the pints to be covered.
- This will ensure that no pint is omitted or that the interview drifts from its original objectives.

Observation

- One technique for collecting data on a process is to observe that process.
- Frequently in systems analysis and design we will **Walkthrough** system observing crucial information flows and decision points.

Questionnaire

- A questionnaire allows us to collect data from a relatively large number of people at a reasonable cost.
- A questionnaire can be thought of as a structured interview form with questions designed so that they can be answered without face-to-face encounter.
- Designing a good questionnaire is a difficult task.

Documents Review

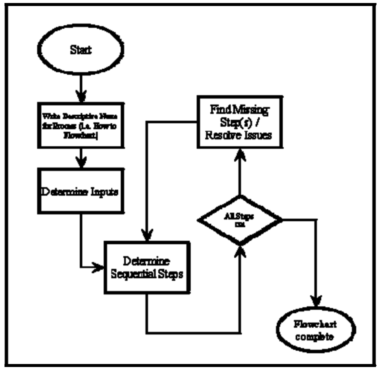
- Some types of information may be obtained from reviewing existing company records or documents.
- The existing work procedures or company policy may be reviewed to obtain information about the current business processes.
- An examination may be made of company records concerning the volume of invoices or shipment to determine problem areas.

Automatic Data Collection

- For automatic data collection accept any automated system such as bar code reading, OCR, systems that automatically count entities (turnstiles for counting people, machines for counting banknotes and coins etc)

Problems of Conventional Flowcharts

- Lack of discipline or structure.
- Very few guidelines on how to develop flowcharts
- Flowcharts for complex systems are often unreadable.


Start
Write Descriptive Name for Process (i.e. How to Flowchart)
Determine Inputs
Determine Sequential Steps
All Steps OK
Find Missing Step(s) / Resolve Issues
Flowchart complete

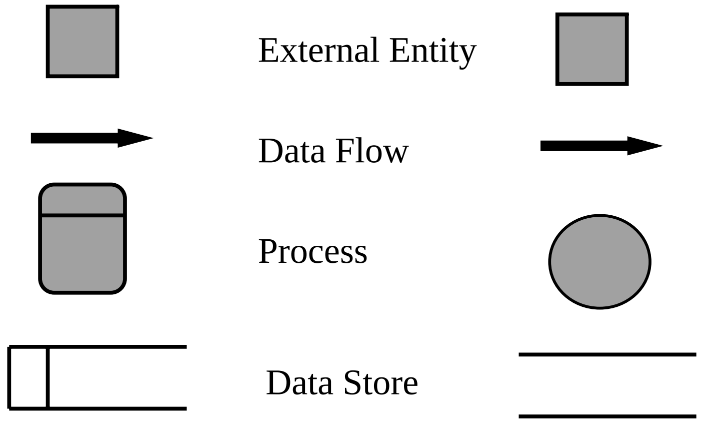
Data Flow Diagrams (DFD)

- A Data Flow Diagram (DFD) is a graphical tool to depict the flow of data through a system and the work or processing performed by that system.
  - ◆ It shows how data moves through an information system but does not show program logic or processing steps.
  - ◆ It only represent a logical model that shows what the system does, not how it does it.

Data Flow Diagrams

- Data Flow Diagrams (DFD) require only four symbols, making them very easy to create.
  - ◆ Open-ended rectangles represent store of data, such as a file of items in inventory, their quantity, and the Economic Recorder Quantity (EOQ).
  - ◆ Circles stand for some type of processing, and arrows represent the flow of data.
  - ◆ A square is a source or destination of data.
  - ◆ Dataflow diagrams should also be prepared in a top-down manner.
- The context diagram is a summary diagram of the first level DFD and serves to put the system under study in its context within the business environment.

DFD Symbols/Elements



External Entity

Data Flow

Process

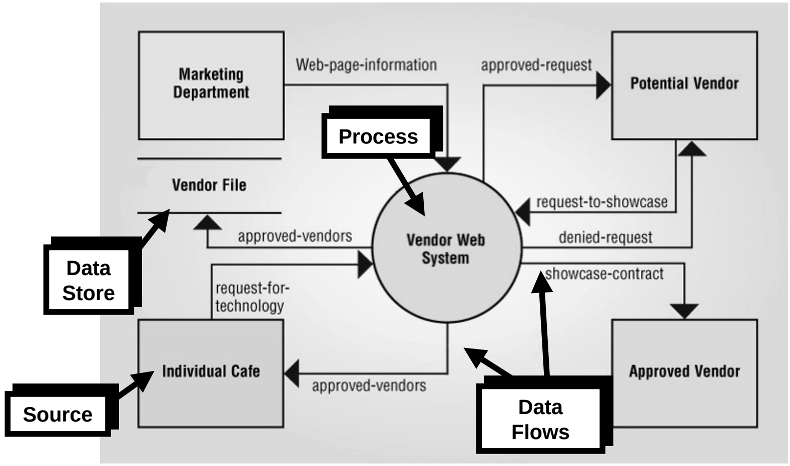
Data Store

Gane & Sarson Notations

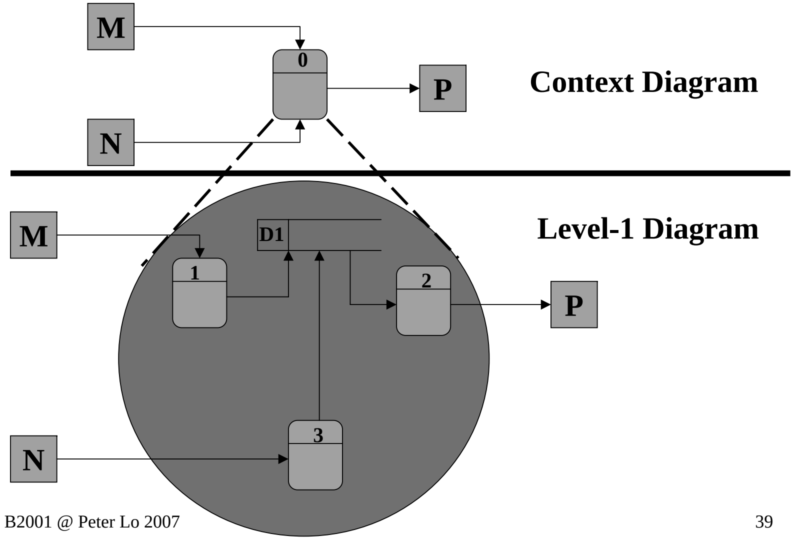
De Marco & Yourdon Notations

Data Flow Diagram (DFD)

- Tool that graphically shows flow of data in system



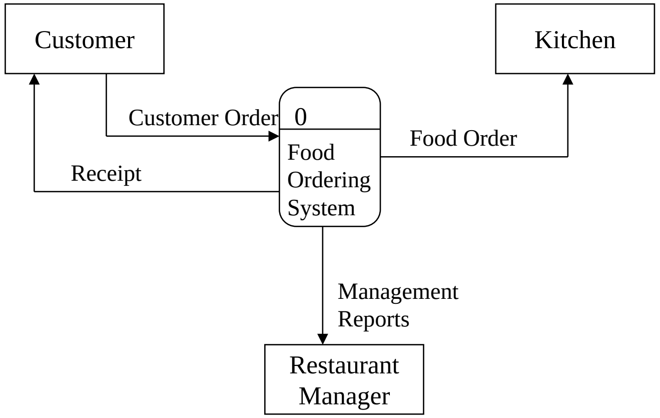
Decomposition of Context Diagram



Context Diagram

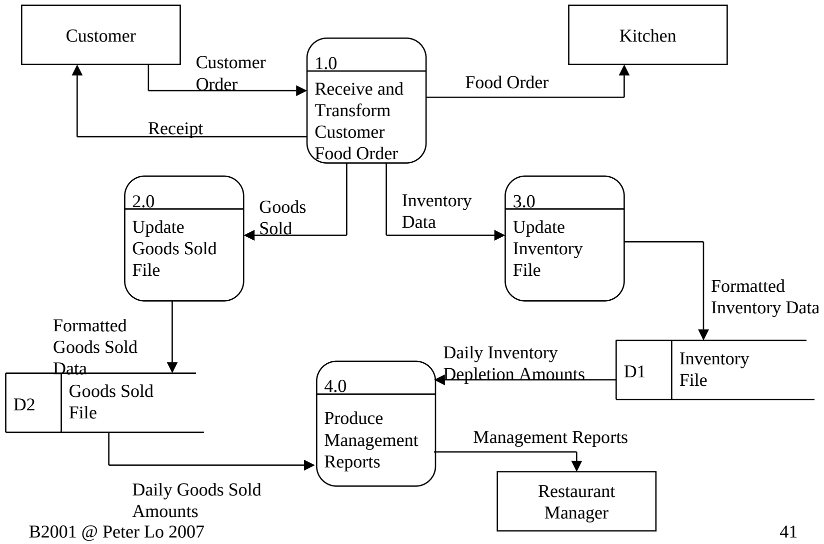
Level-1 Diagram

Context Diagram



Context DFD of Hoosier Burger’s food ordering system

Level 1 Diagram

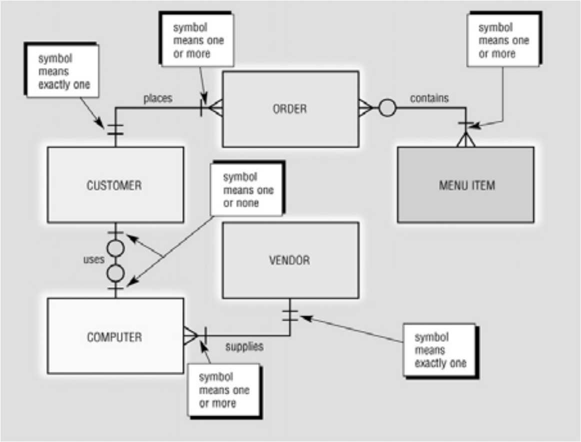


Entity-Relationship Diagram (ERD)

- Tool that graphically shows connections between entities in system

Entity

Object in system that has data



Entity Diagrams

- An entity is an item of interest to the organization like a department, staff, machines, customers, etc.
- An entity diagram shows the relationships between such items of interest.
- Take the example of a department and its staff, both of which are entities.
- The relationship between these entities can be shown by drawing a diagram linking them together. We could also show a particular kind of relationship like one to many.
- The entity diagram can be used to understand and communicate the relationships between entities.
- The data fields (attributes) associated with each entity may also be documented.