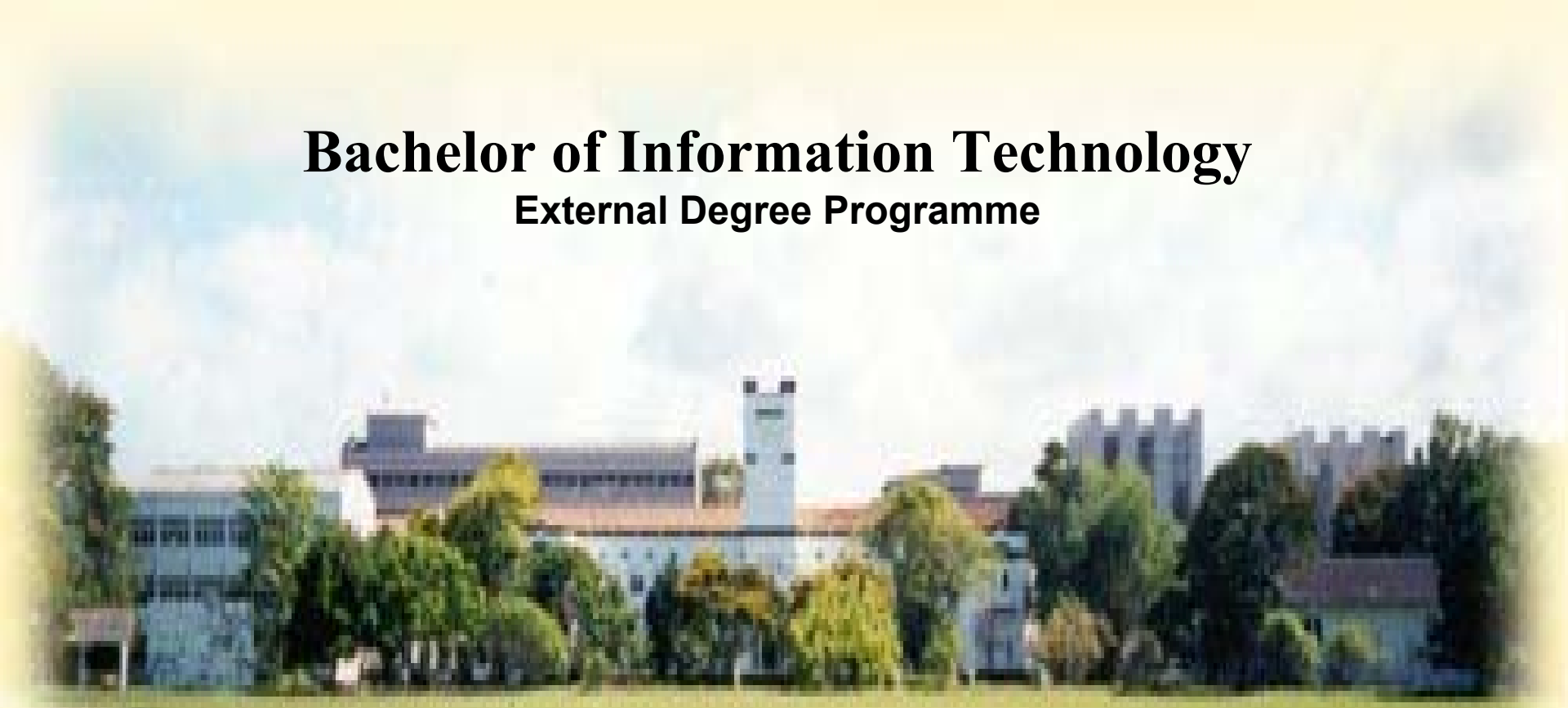
Bachelor of Information Technology

External Degree Programme



Systems Analysis and Design

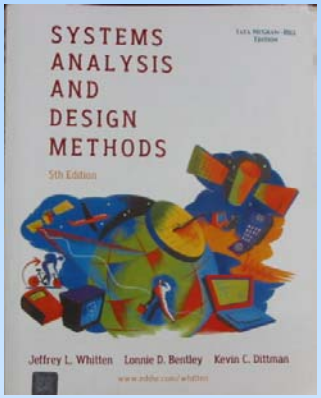
University of Colombo School of Computing

UCSC

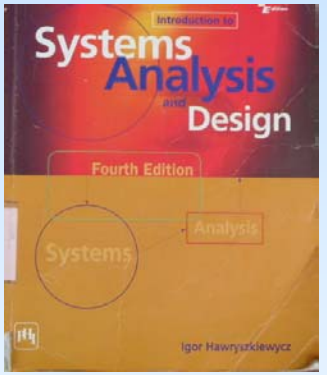
BIT

References

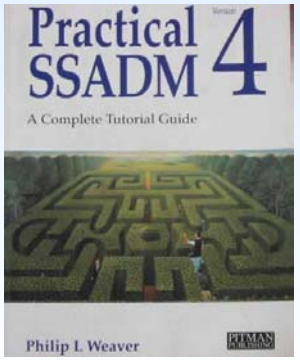
Ref_1 : System Analysis and Design Methods
By Whitten, Bentley, Dittman
ISBN 0-07-044539-7 (5th Edition)



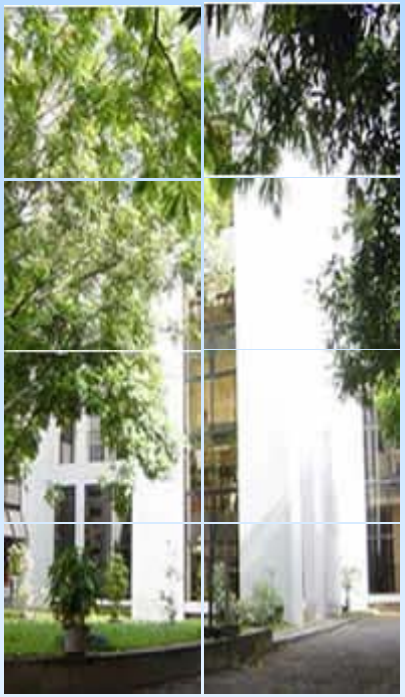
Ref_2 : System Analysis and Design **By** Igor Hawryszkiewycz (4th Edition) ISBN 81-203-1670-3



Ref_3 : Practical SSADM Version 4 A Complete Tutorial Guide **By** Philip L Weaver ISBN 0-273-60095-8



Introduction to System Development Environment



UCSC

BIT

Information Systems

Introduction

- Earlier applications

Airline Reservations

Keeping records of transactions

Keeping records of Stock

Information Systems

Introduction

- Computers are now becoming part of virtually every activity in organization

Production

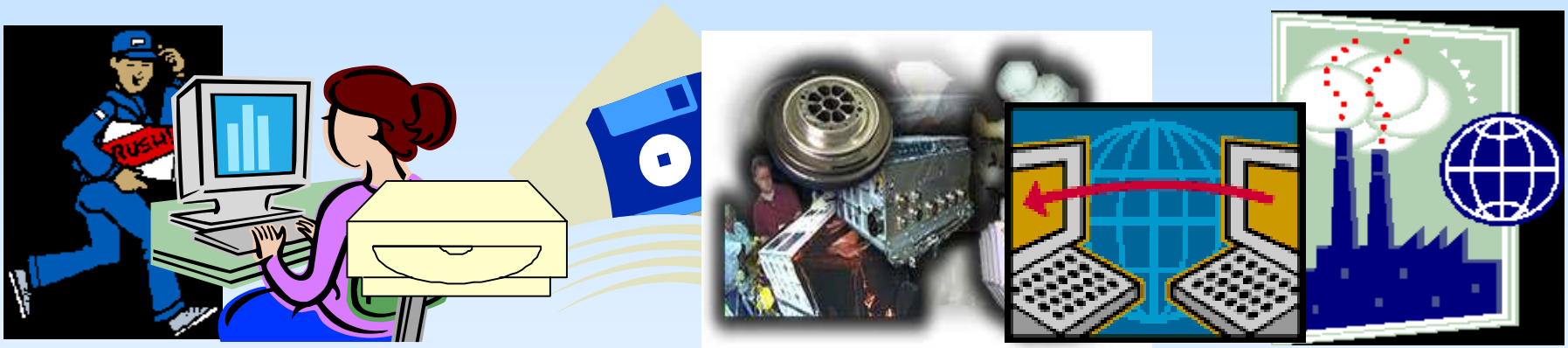
HRM - Training

Telephone Integration

Information System…

Information System

- It is an arrangement of



People Interface Data Technology Network Process

- Improve day to day operations
- problem solving and decision making

Information System…

Information Technology

- It is an arrangement of

+

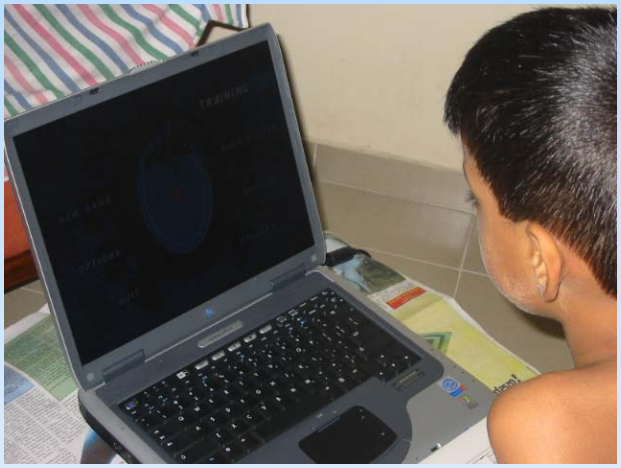
Systems Design Environment…

Stakeholder

Is any person who has an interest in an information system.

Systems Design Environment…

- Stakeholders cont..



Use the system to perform or support the work to be completed.

System Users or Clients

Systems Design Environment…

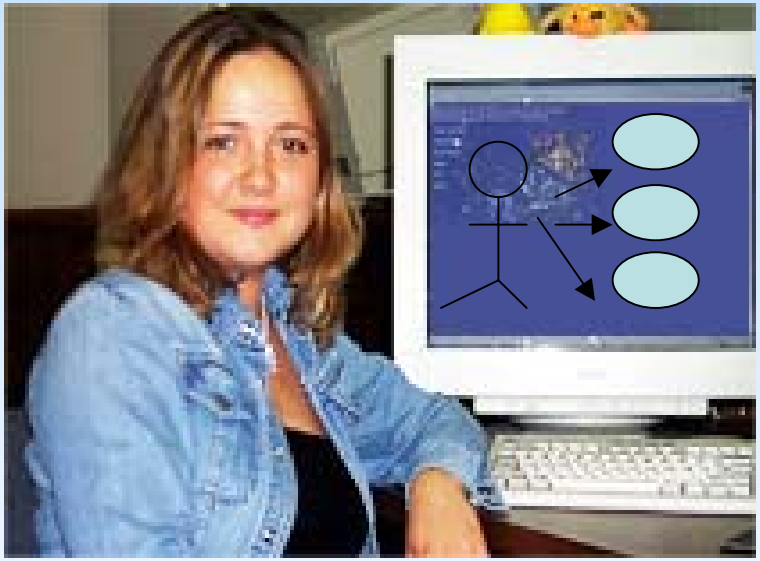
- Stakeholders cont..

System Owner

Owns the system.

Systems Design Environment…

- Stakeholders cont..

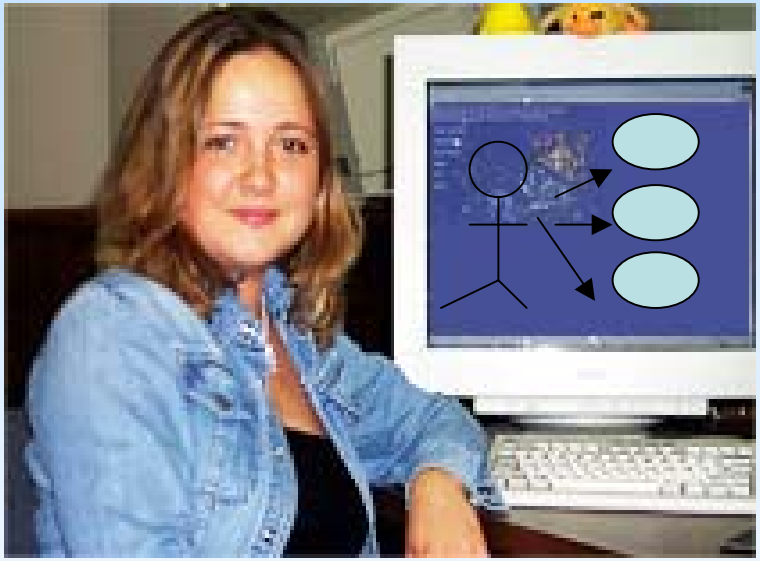


System Designer

Design the system to meet the users requirements

Systems Design Environment…

- Stakeholders cont..

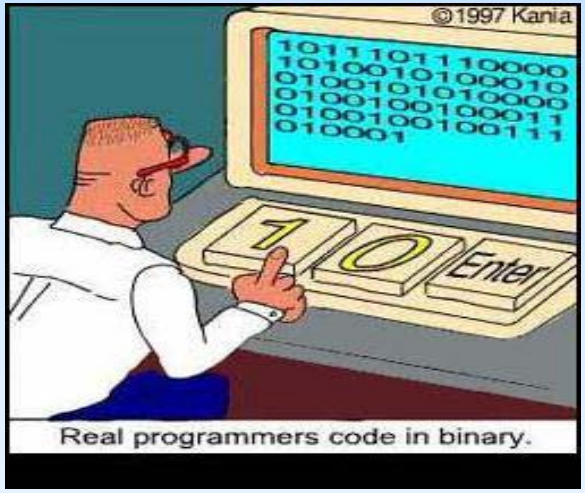


System Designer

Design the computer files, databases, inputs, outputs, screens, networks, and programs that will meet the system users requirements.

Systems Design Environment…

- Stakeholders cont..



System Builders

Construct, test and deliver the Information System

Systems Design Environment…

- Stakeholders cont..

Systems Analysts

People who understand both business and computing

Systems Design Environment…

- Stakeholders cont..

Systems Analysts

Bridge the communication gap that exists between non technical and technical people involved with building systems.

Systems Design Environment…

- Stakeholders cont..

Systems Analysts

What does a systems analyst do?

- Identify the problem
- Analyze and understand the problem
- Identify the solution requirements
- Identify alternative solutions
- Design and implement the best solution
- Evaluate the result

Systems Design Environment…

- Stakeholders cont..

Vendors

Sell Hardware, Software, and services to business

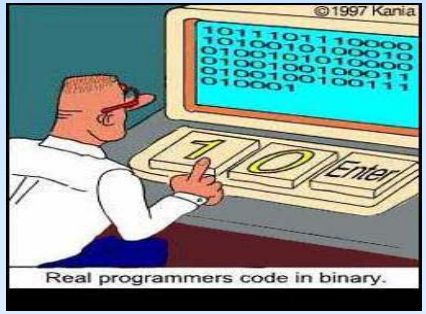
Consultants

Systems Design Environment…

Information System Development Teams usually consists of the these 6 Stakeholders

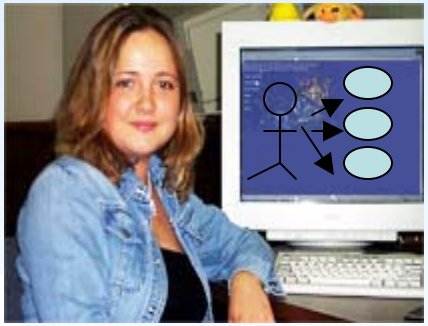
System User

System Owner



System Builders

Vendors



System Designer

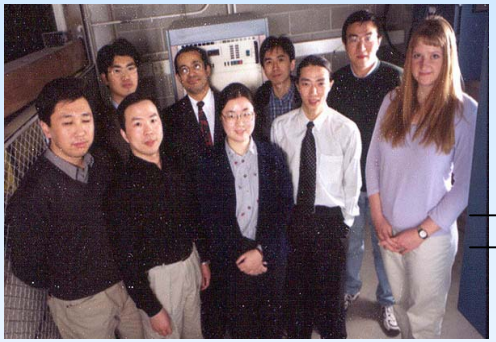
System Analysts

Consultants

Systems Design Environment…

Outsourcing

Contracting with an outside vendor to assume responsibility…



In house developer

Technology Source

Out side vendors

Less costly

Value added

Systems Design Environment…

Legacy systems

- Old Information System Applications
- Use Old or Outdated Technologies that were popular at the time.

Systems Design Environment…

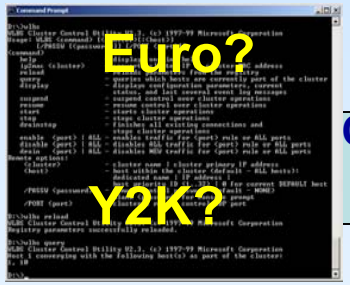
Legacy systems cont.

Support old business requirements

Support new business requirements

Old technology

Old standard



Converted to satisfy new environments

New standard

New technology

New functionality

Old system

New system

Systems Design Environment…

Legacy systems cont.

- Many complex legacy systems yet to be upgraded to new technologies because of
  - Cost,
  - Skills and
  - People required
- Force to change – to reflect new or changing business requirements.
  - Year 2000 problem (Y2K)
  - Euro conversion

Systems Design Environment…

Legacy systems cont.

Y2K problem

- **Many computers and applications stored date with only 2 digits. (e.g. 99 =1999)**
- **Problems : when the millennium changed (e.g. 03=2003)**

Born in 1978
Age? -75, 0, 75

Types of Information Systems

- Transaction Process System (TPS)
- Management Information System (MIS)
- Decision Support System (DSS)
- Expert System (ES)
- Office Automation and Work Group System
- Centralized Systems
- Distributed Systems
  - Client Server Systems
  - Network Computing System

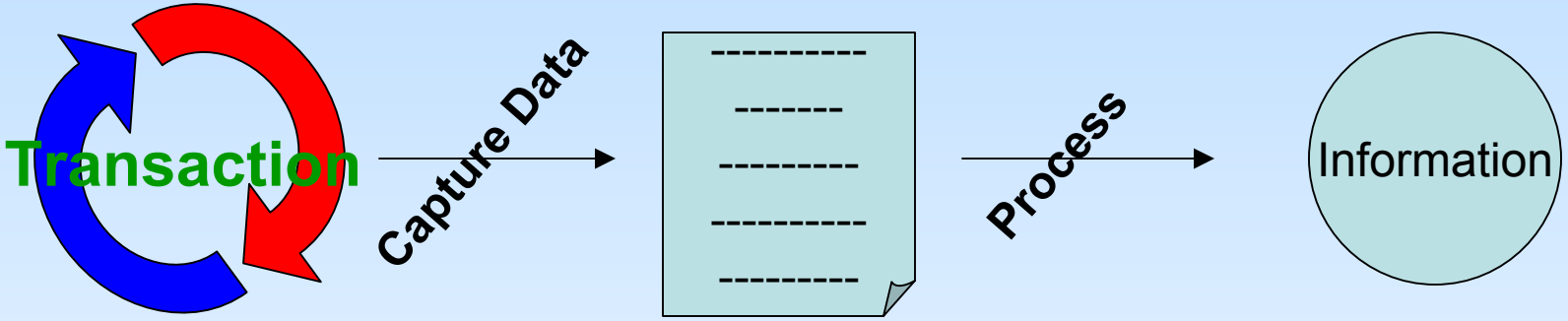
UCSC

BIT

Types of Information System

Transaction Process Systems (TPS)

Information systems that capture and process data about business transactions



Examples

Airline Reservations

Retail point of sale

Bank deposit and withdrawal

BIT @UCSC Course Registration

UCSC

BIT

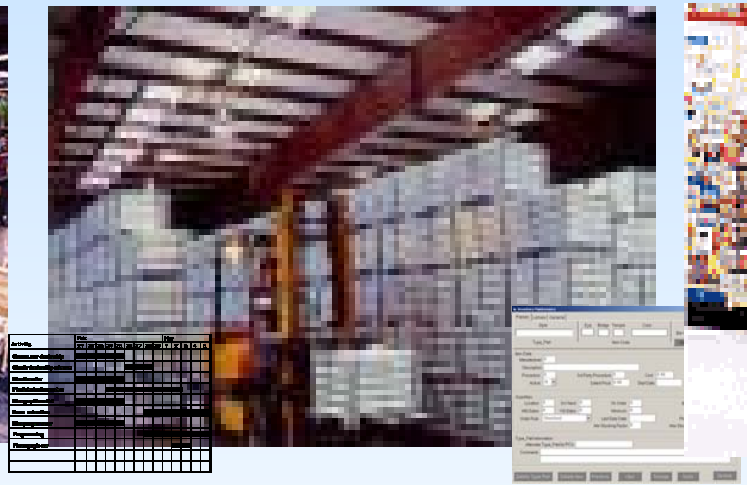
Types of Information System

Management Information System (MIS)

MIS is an information system application that provides for management oriented reporting.

Examples

Production scheduling



Inventory reporting

Sales forecast

Types of Information System

Management Information System (MIS) cont..

Normally produced from a shared database.

Produces

Information

Summery information

Detail information

Exception information

Types of Information System

Decision Support System (DSS)

- Provides its user with decision-oriented information whenever decision making situation arises.
- Sometimes called **executive information system**.

Types of Support

- Analysis of possible decision that will affect the decision
- Simulation of possible solutions and their likely results.

- Identification of problems or decision making opportunities
- Identification of possible solutions and decisions
- access to information needed to solve a problem or make a decision

UCSC

BIT

Types of Information System

Decision Support System (DSS) cont..

- DSS tools include
  - Spreadsheets
  - PC-Database Management Systems
  - Custom reporting tools
  - Statistical Analysis programs

Types of Information System

Expert System (ES)

-- An expert system is a programmed decision making information system.
-- It capture and reproduces the knowledge and expertise of a decision maker and
-- Simulates the “thinking” of the expert.

Implemented with

Artificial Intelligent (AI)
Technology that captures, stores, and provides access to the reasoning of the experts.

Types of Information System

Office Automation Systems

-- Support wide range of business office activities
-- Provide facilities for improved work flow and communication between workers
-- Workers may not be located in the same office.

Work group scheduling

E-Mail

Imagi
ng
facsim
ile

Electronic document

Work group computing

UCSC

BIT

Types of Information System

Office Automation Systems

Designed to support

--- Individuals

--- Workgroups

Personal Information Systems

- for a single user

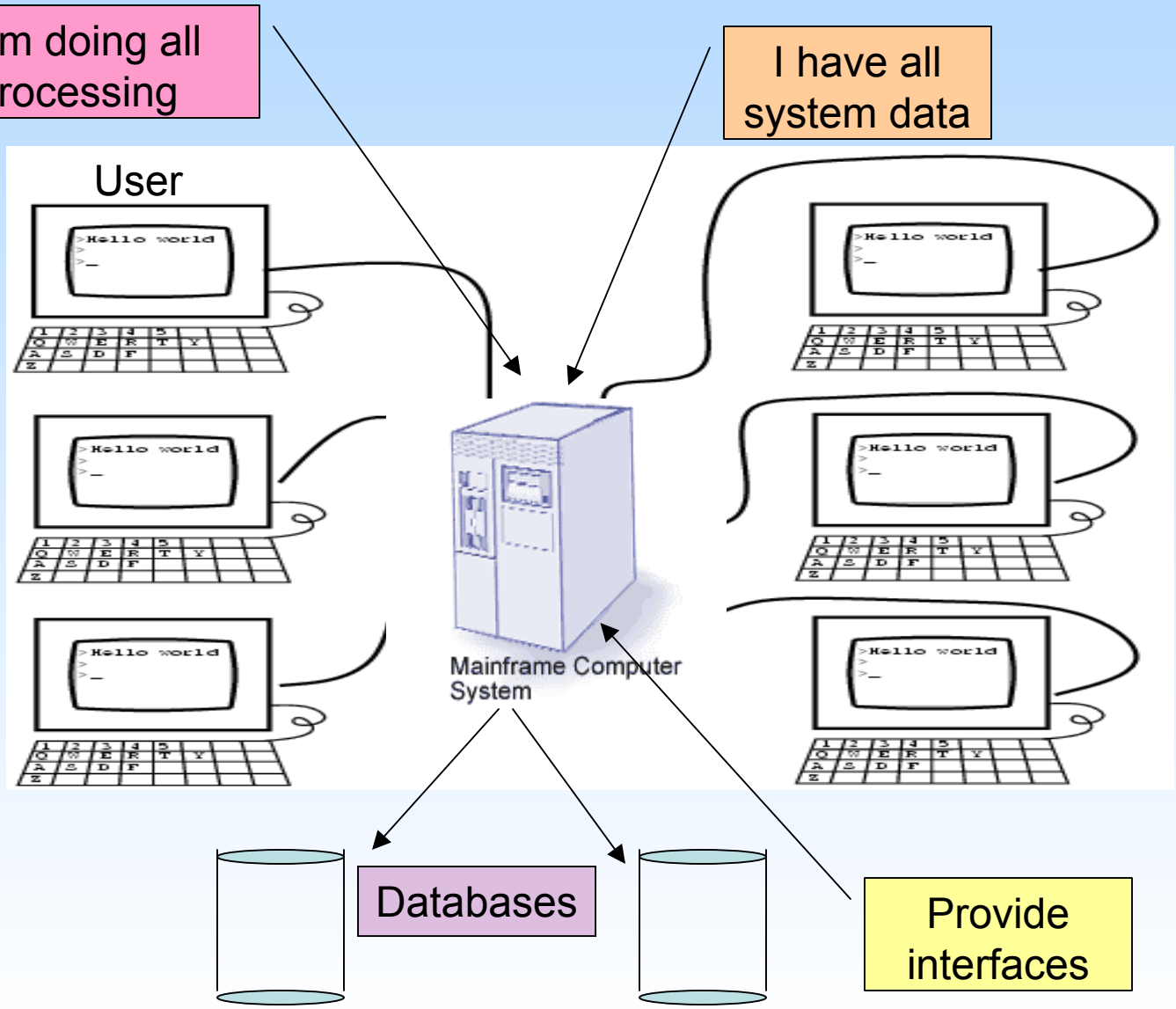
Work group Information Systems

- for a work group

UCSC

BIT

Centralized Systems



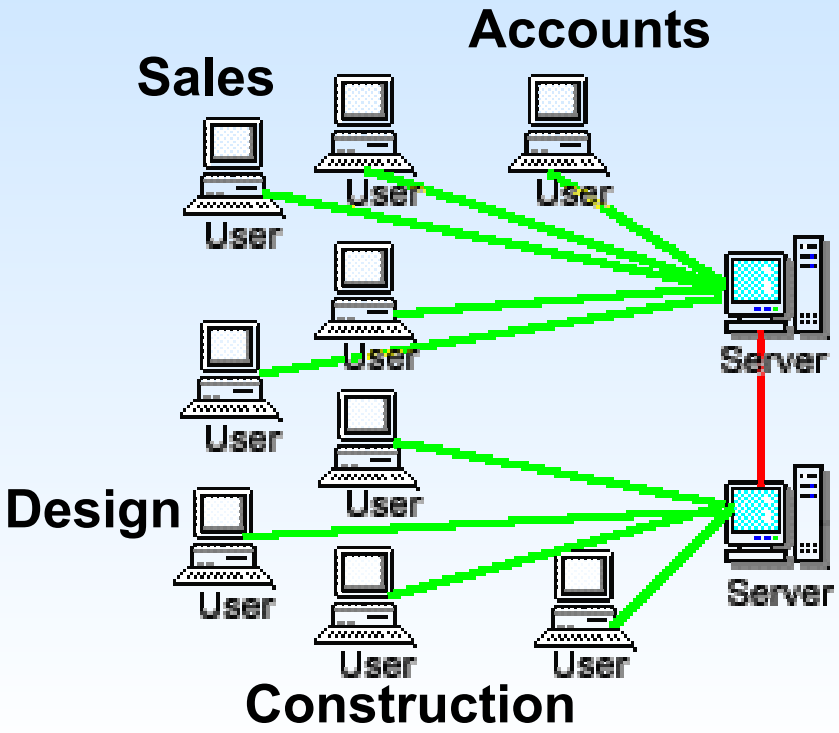
Distributed Systems

-- Components of an information system
-- Processing workload required to support the components

Distributed to multiple locations

1. Client Server Systems

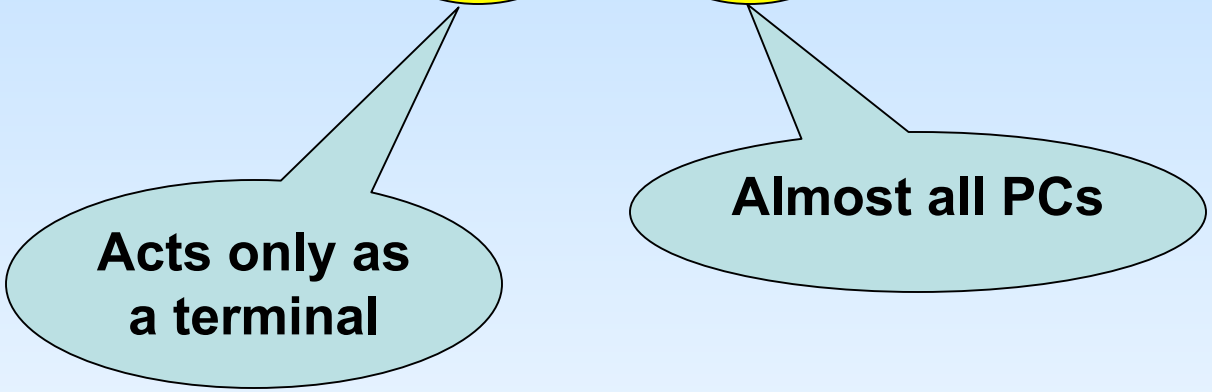
Solutions are distributed between client PC's and one or more servers.



Distributed Systems

Client Server Systems cont..

Clients may be thin or flat



e.g. Windows terminal

Distributed Systems

1. Network Computing Systems

- Presentation and presentation logic layers are implemented in a client side web browser
- The presentation logic layer then connects to the application logic layer that run on an application server,
- Subsequently connects to the database server/s

UCSC

BIT

Processing Types

1. Batch Processing

The data entered is collected into files called batches. Each file is processed as a batch of many transactions.

Super market-Batch processing

Processing Types

2. Online Processing

The captured data is processed immediately.

ATM-Real time processing

Development Process

Methodologies

- have a predefined set of steps, and
- a Collection of tools

used to design a system

It ensures that systems are built in the most effective way.

Tools

Methods

Most Effective Way of Building

Development Process cont.

Methodologies provide the organizers framework for systems development.

- Defines :
  - **The stages into which the development will be broken**
  - **The tasks to be performed.**
  - **The deliverables from these tasks**
  - **Standards to be used**
  - **Quality checks to be applied .**

    Etc.

UCSC

BIT

Development Process cont.

Modeling Methods or Techniques

- Techniques used to implement the Methodology.
- Provides the descriptions of the business system requirements from various view points.

E.g. Data Flow Diagrams, Entity Relationship Diagrams Use Case Diagrams etc.

Development Process cont.

Tools

Software systems that assists analysts and designer to build computer based information system.

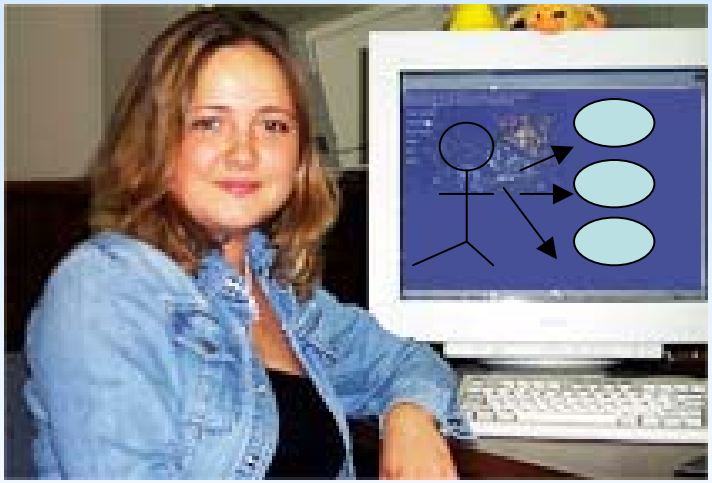
Development Process cont.

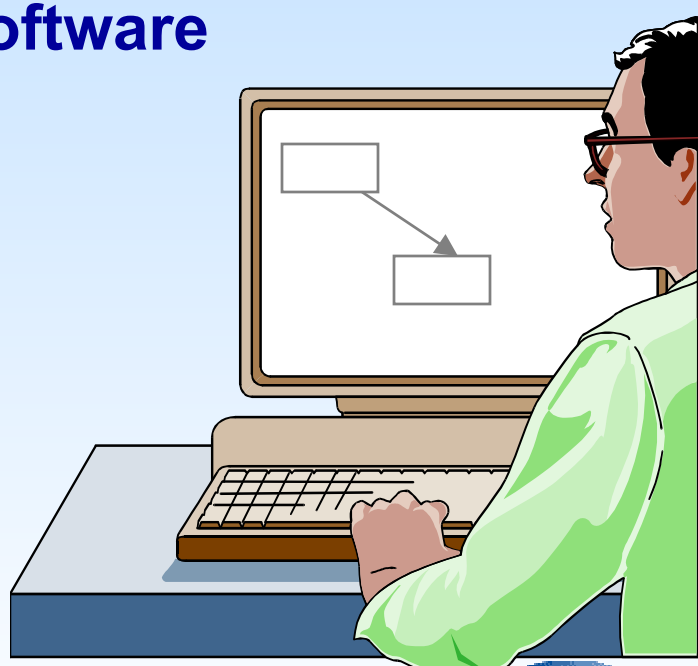
Tools

General Aim :

Decrease the human effort required to develop software.

and/or **increase the quality of software**



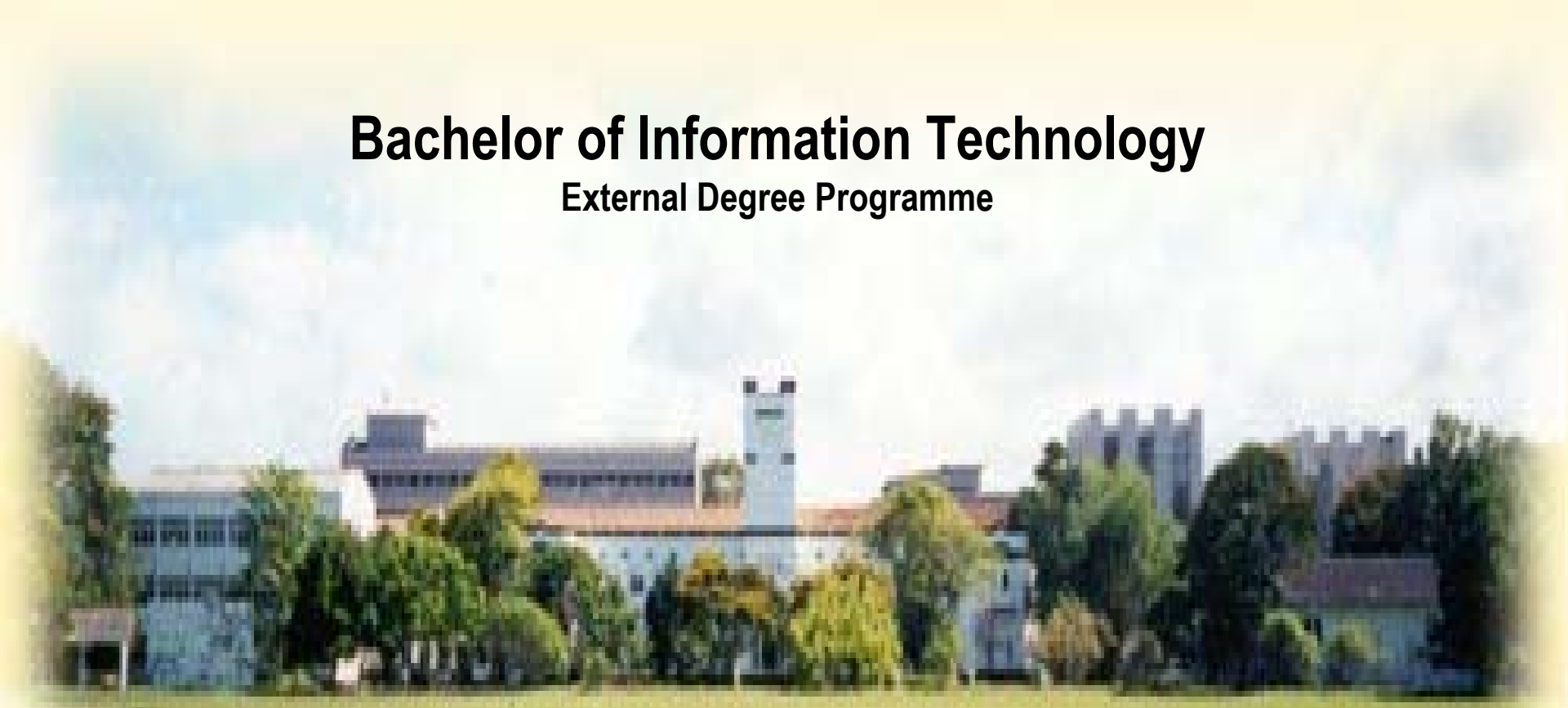


UCSC

BIT

Bachelor of Information Technology

External Degree Programme



Systems Analysis and Design

Ref_1 : System Analysis and Design Methods **By** Whitten, Bentley, Dittman ISBN 0-07-044539-7 (5th Edition)
Ref_2 : System Analysis and Design **By** Igor Hawryszkiewycz (4th Edition) ISBN 81-203-1670-3
Ref_3 : http://www.mhhe.com/whitten
Ref_3 : Practical SSADM Version 4 A Complete Tutorial Guide **By** Philip L Weaver ISBN 0-273-60095-8

University of Colombo School of Computing

UCSC

BIT

System Development Life Cycle

UCSC

BIT

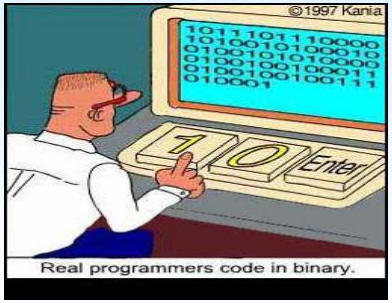
System Development Life Cycle (SDLC)

Problem Definition

Requirement Analysis

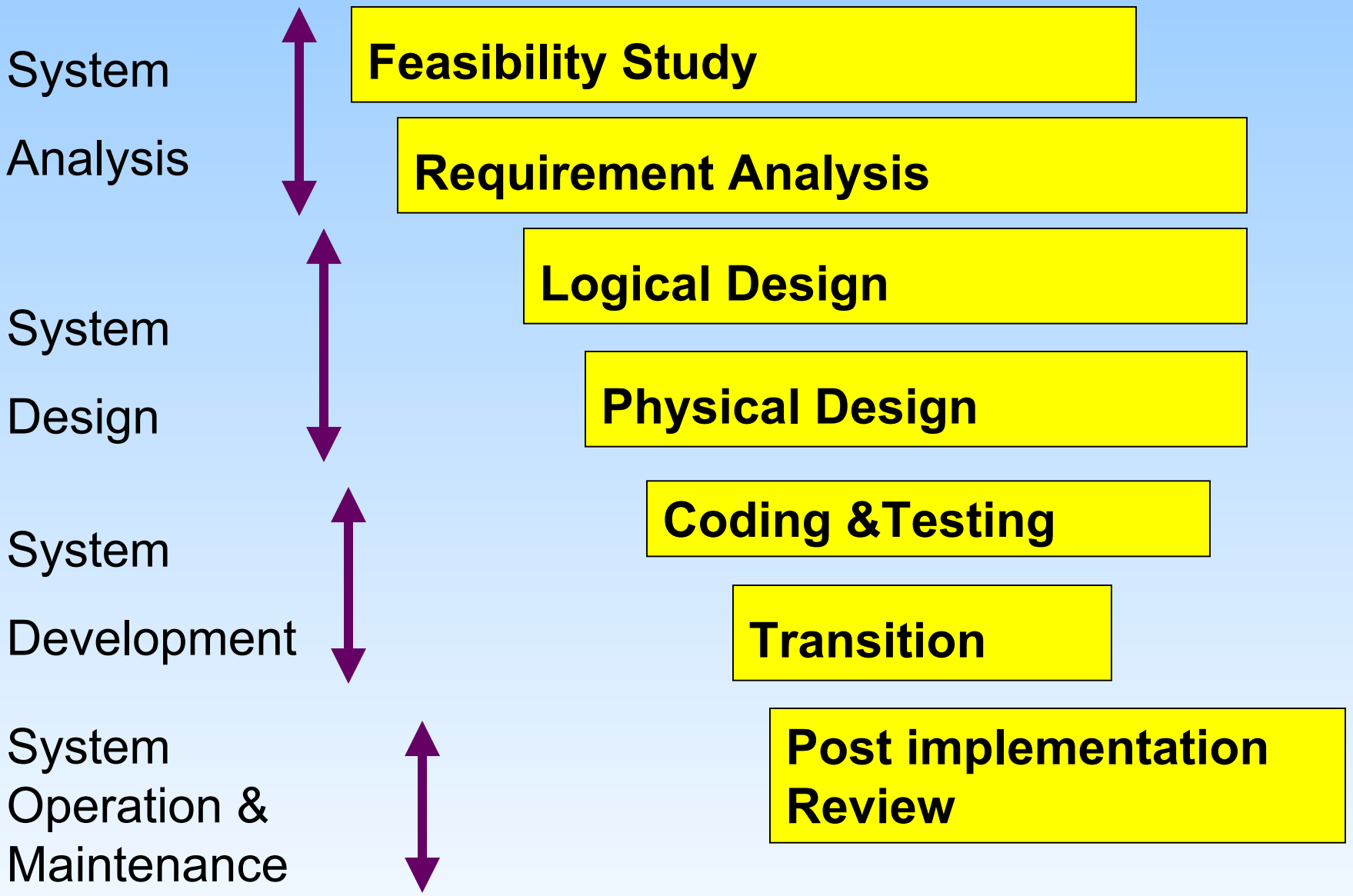
System Design

System Development



System Testing

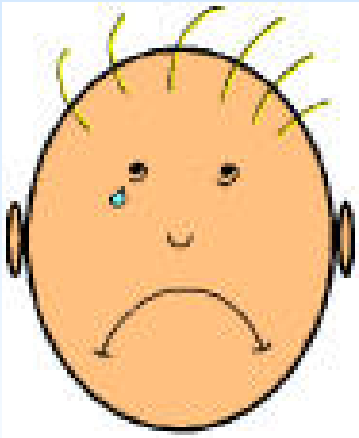
Maintenance


System Analysis
System Design
System Development
System Operation & Maintenance
Feasibility Study
Requirement Analysis
Logical Design
Physical Design
Coding &Testing
Transition
Post implementation Review

Why we need a life cycle in systems development?

- **to ease the process** of building a system.

- **to avoid failures** like unclear objectives, cost overruns, and
- **to avoid systems not meeting user needs**



ucsc

BIT

Facilitate project review

Provides a more efficient / effective system

Guides the systems development

Advantages of life cycle

Improve communications

Improve management and control

Computer-based support

ucsc

BIT

Waterfall Cycle

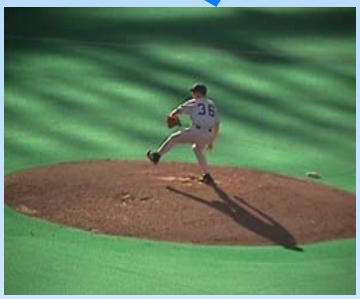
- Problem Definition
- Requirement Analysis
- System Design
- Implementation
- System Testing
- Maintenance

Definition the Problem

Project goals

Provides a broad statement of user requirements in users terms, or what the users expect the system to do

Project bound



***project bounds are set during this phase*. Defines what part of the system can be changed by the project and what parts are to remain same.**

Project limits

Specify the resources to be made available for the project (*resource limits*). *Project goals, project bounds and resource limits* are some times called projects terms of reference. They are set by the organization's management.

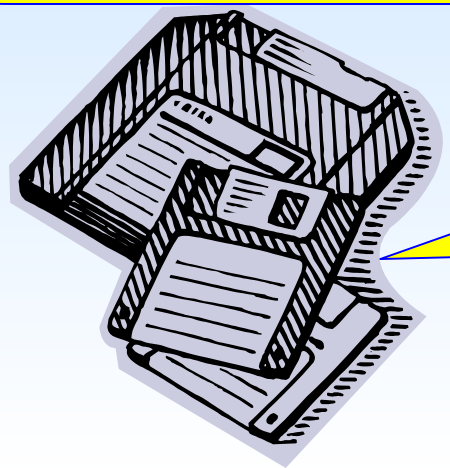
ucsc

BIT

Requirement Analysis

how the current system works and what it does

Producing a detailed model in subject terms of what the new system will do and how it will work.



Producing a high-level description of the system

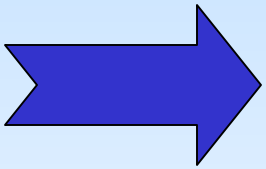
ucsc

BIT

System Design

- Produces a design specification for the new system

Analysts



Design

System Design

- Things to be done:
  - Select equipment
  - Specify new programs or changes to existing programs
  - Specify new database or changes to existing database
  - produce detailed procedures

Design

System Development

- Individual system components are built and tested
- Data and tools are used to build the system
- User interfaces are developed and tried by users
- Database is initialized with data

Analysts

System

System testing

- Test and evaluate results, and
- the system ready to be delivered to the user/client.

Maintenance

- Eliminate errors in the system during its working life.
- Fixing any bugs and problem found by users
- Tune the system to any variations in its working environment

Problems with waterfall cycle

- ✓ It has rigid design and inflexible procedure
- ✓ It is top-down procedure
- ✓ One phase must be completed before the next phase starts, and
- ✓ No phase can be repeated.

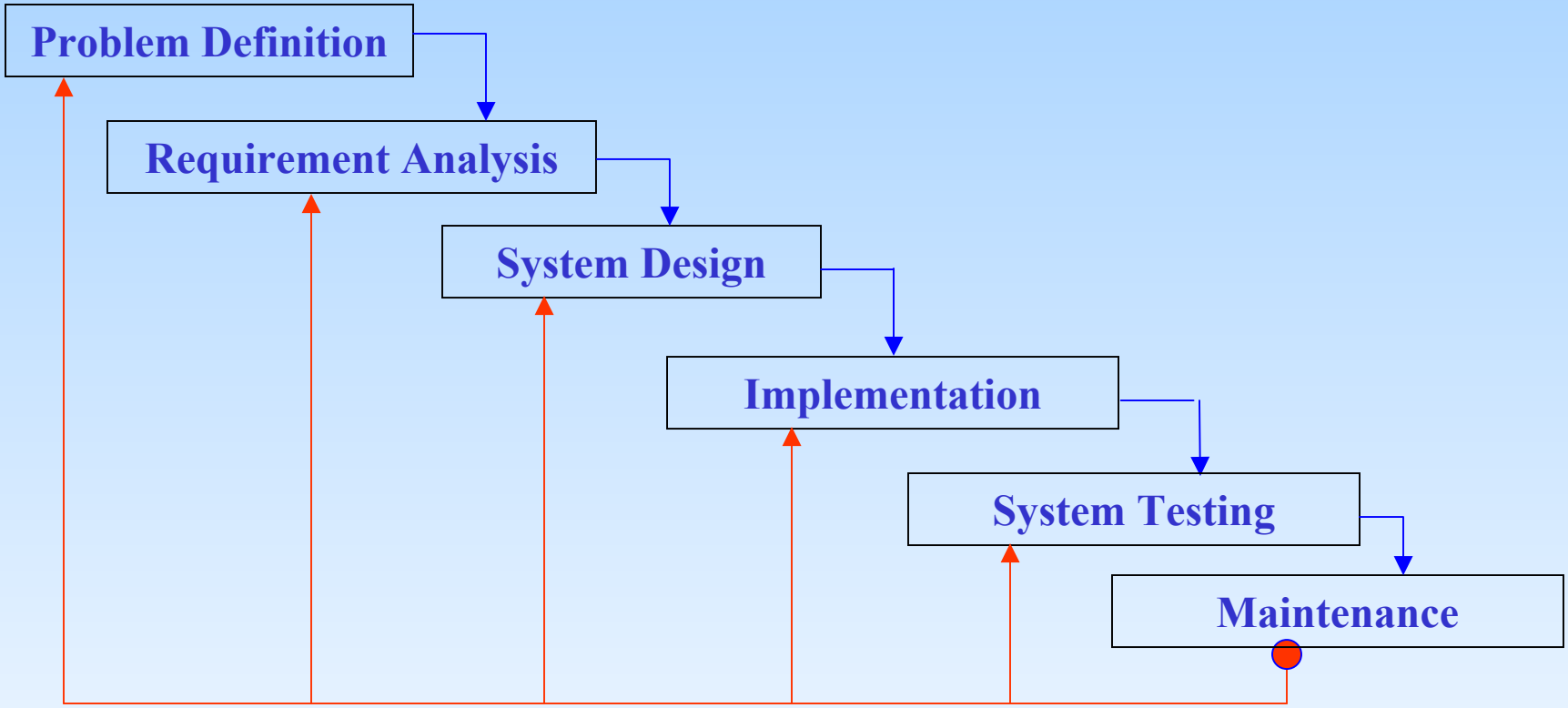
Criticisms fall into the following categories:

- ✓ Real projects rarely follow the sequential flow that the model proposes.
- ✓ At the beginning of most projects there is often a great deal of uncertainty about requirements and goals, and it is therefore difficult for customers to identify these criteria on a detailed level. The model does not accommodate this natural uncertainty very well.

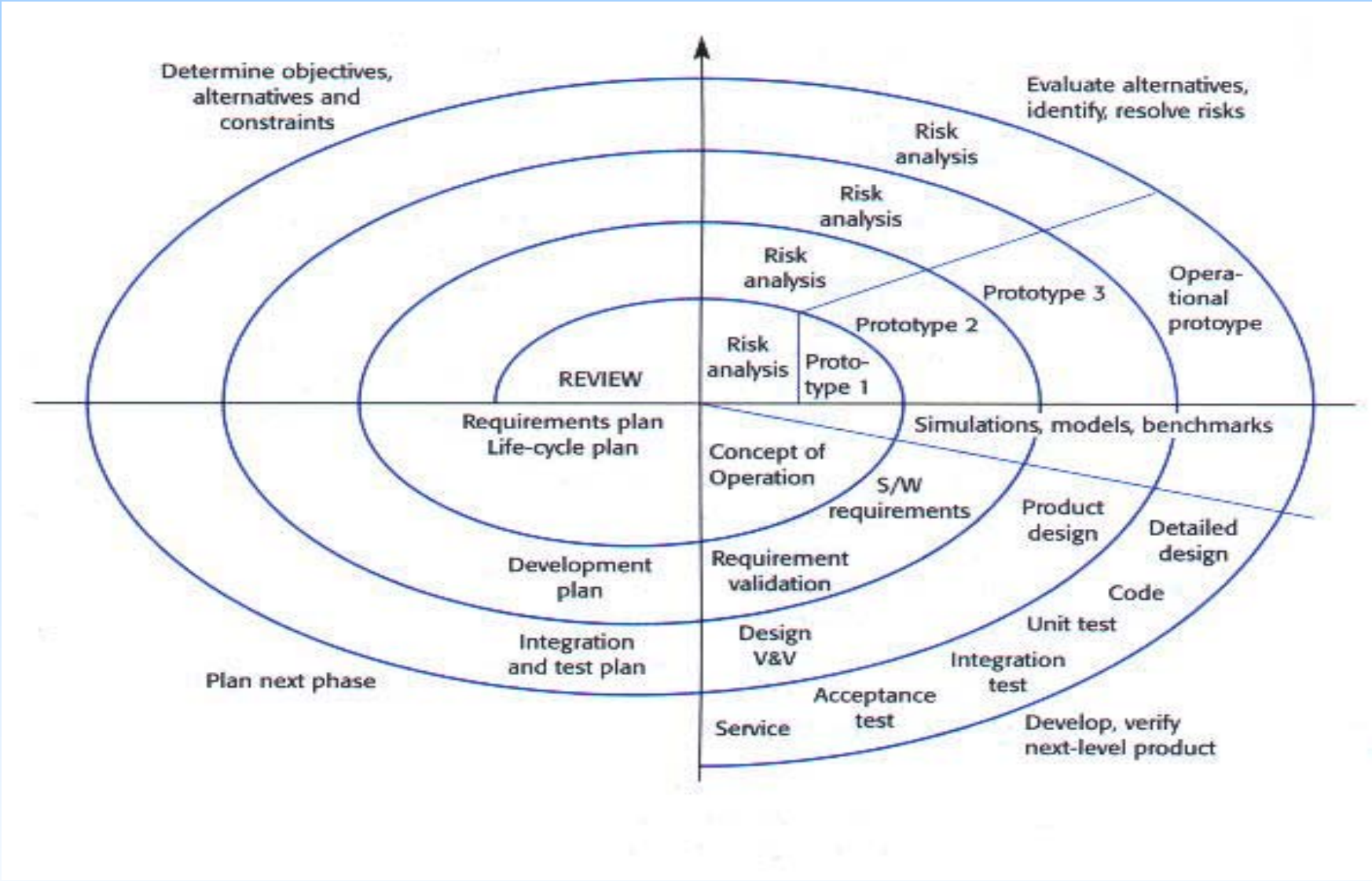
Criticisms fall into the following categories: cont…

- ✓Assumptions made in the early phases no longer hold
- ✓Some of the early work is incomplete
- ✓Something was overlooked or not completely understood.

Modified waterfall model



Spiral Model



Ref2 –Ststems Analysis and Design Igor Hawryszkiewycz,.

Spiral model has four phases

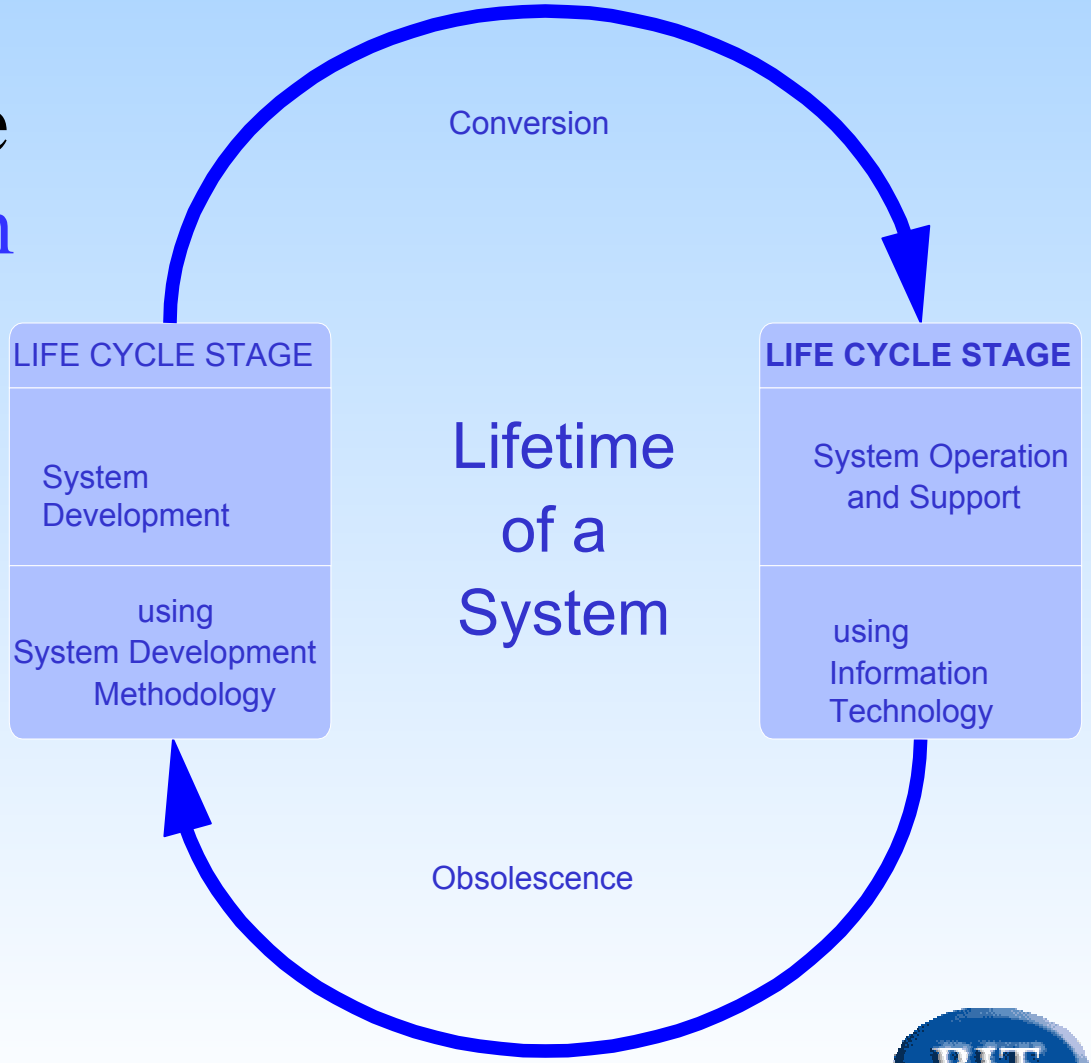
- Requirement analysis
- Design prototype
- Develop and integrate
- Acceptances and testing

Life Cycle vs Methodology

- The software development process consists of several well-defined steps.
- When following a design methodology, a designer can select appropriate modeling method related to each step

Life Cycle vs Methodology

- A **system life cycle** divides the life of an information system into two major stages, **systems development** and **systems operation and support.**

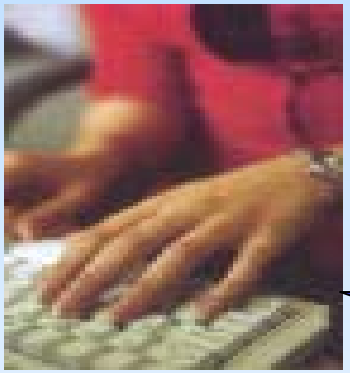


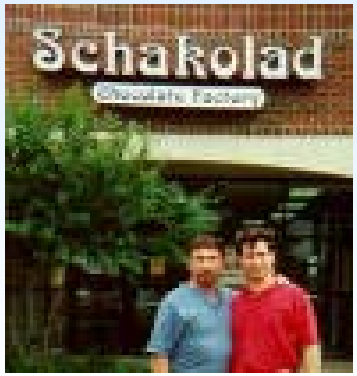
Life Cycle vs Methodology

- A **system development methodology** is a very formal and precise system development process that defines
  - a set of activities,
  - methods,
  - best practices,
  - deliverables,
  - and automated tools

A System Development methodology (Process) has some general principles.

P1: Get the owner and User Involved





Involved

System Development

P2: Use a problem-solving approach.

Study and understand the problem and its context

Define the requirement of a suitable solution.

Identify candidate solutions and select the best solution.

Design and/or implement the solution.

Observe and evaluate the solution's impact, and refine the solution accordingly.

ucsc

BIT

P3:Establish phases and activities.

Preliminary investigation

Problem analysis

Requirement analysis

Decision analysis

Design

Construction

Implementation

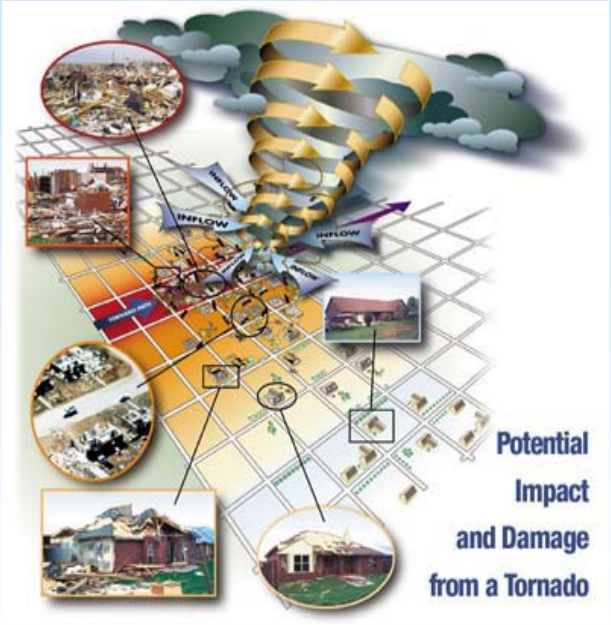
P4: Establish standards.

Documentation

Quality

Automated tools

#Information Technology



ucsc

BIT

P5:Justify systems as capital investments.

Cost-effectiveness

Risk management

P6:Don’t be afraid to cancel or revise scope.

Cancel the project if it is no longer feasible

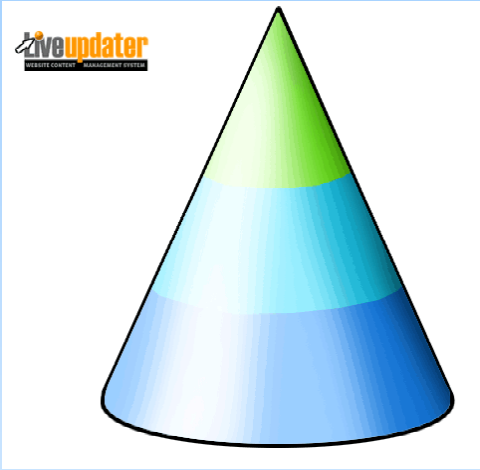
If project scope is to be increased, reevaluate and adjust the cost and schedule

If the project budget and schedule are frozen and not sufficient to cover all project objectives, reduce the scope.

P7:Divide and conquer.

#We divide a system into subsystem and components

--- Easily to conquer the problem
--- Easy to build a large problem



P8: Design systems for growth and change.

Correcting simple mistakes

To redesigning the system to accommodate changing technology

To making modification to support changing user requirements.

Development Process

- Methodology
- Modeling Methods or Techniques
- Tools

Major Components

Methodology

- Provides the framework
- Has a predefined set of steps
- Ensures that systems are built in the most effective way

e.g. SSADM, RUP

Methodology

- Uses tools and modeling methods

Tools

Methods

Most Effective Way of Building

Modeling Methods

A set of techniques used to implement a Methodology

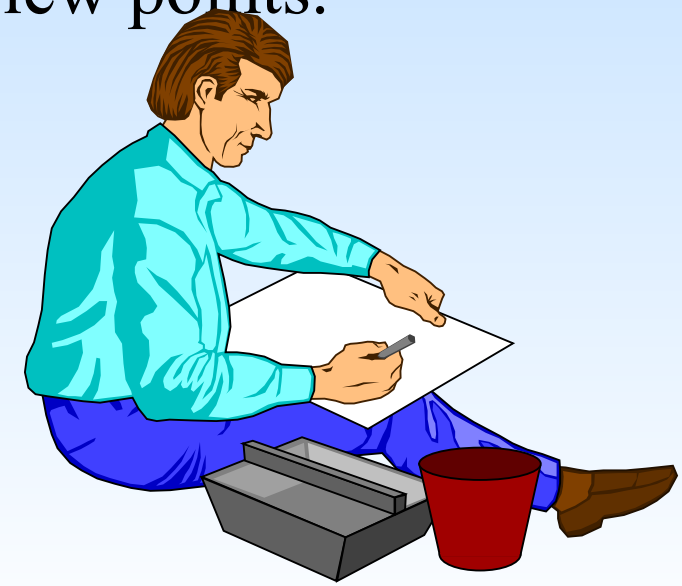
- Data Flow Diagrams
- Entity Relationship Diagrams
- Structure Charts etc.

Different Views of the System

System Development Methodologies

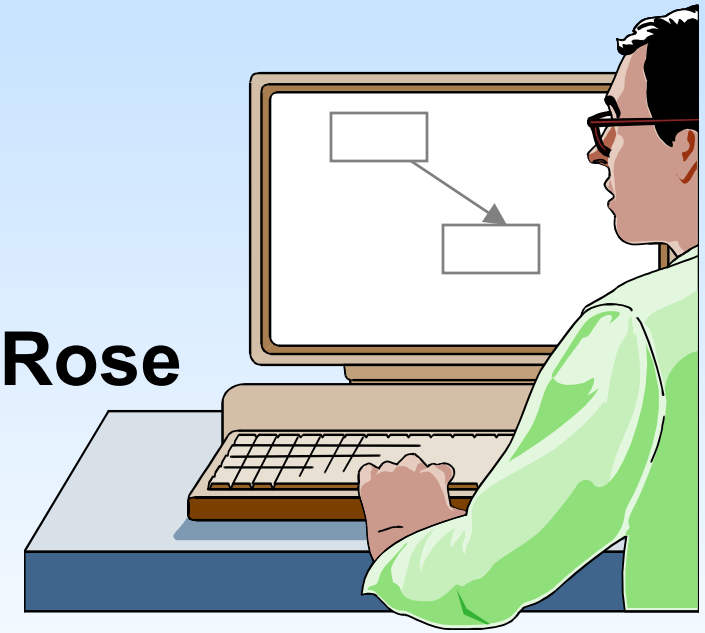
Supported by Modeling Methods or Techniques

- Techniques used to implement the Methodology.
- Provides the descriptions of the business system requirements from various view points.



Tools

- Software systems
- Assists analysts and designer to build information systems.

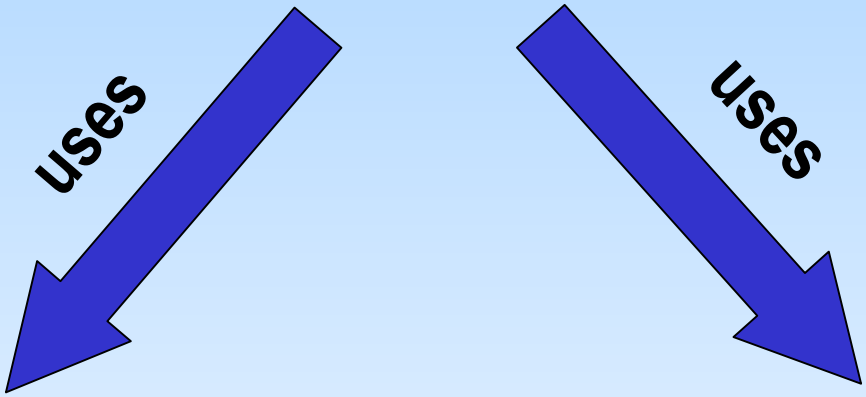


e.g. Easy Case, Rational Rose

System Development Methodologies.

Supported by Tools.
They will not replace Systems Analysts.

Methodology
Eg .Rational Unified Process



Modeling Methods or Techniques

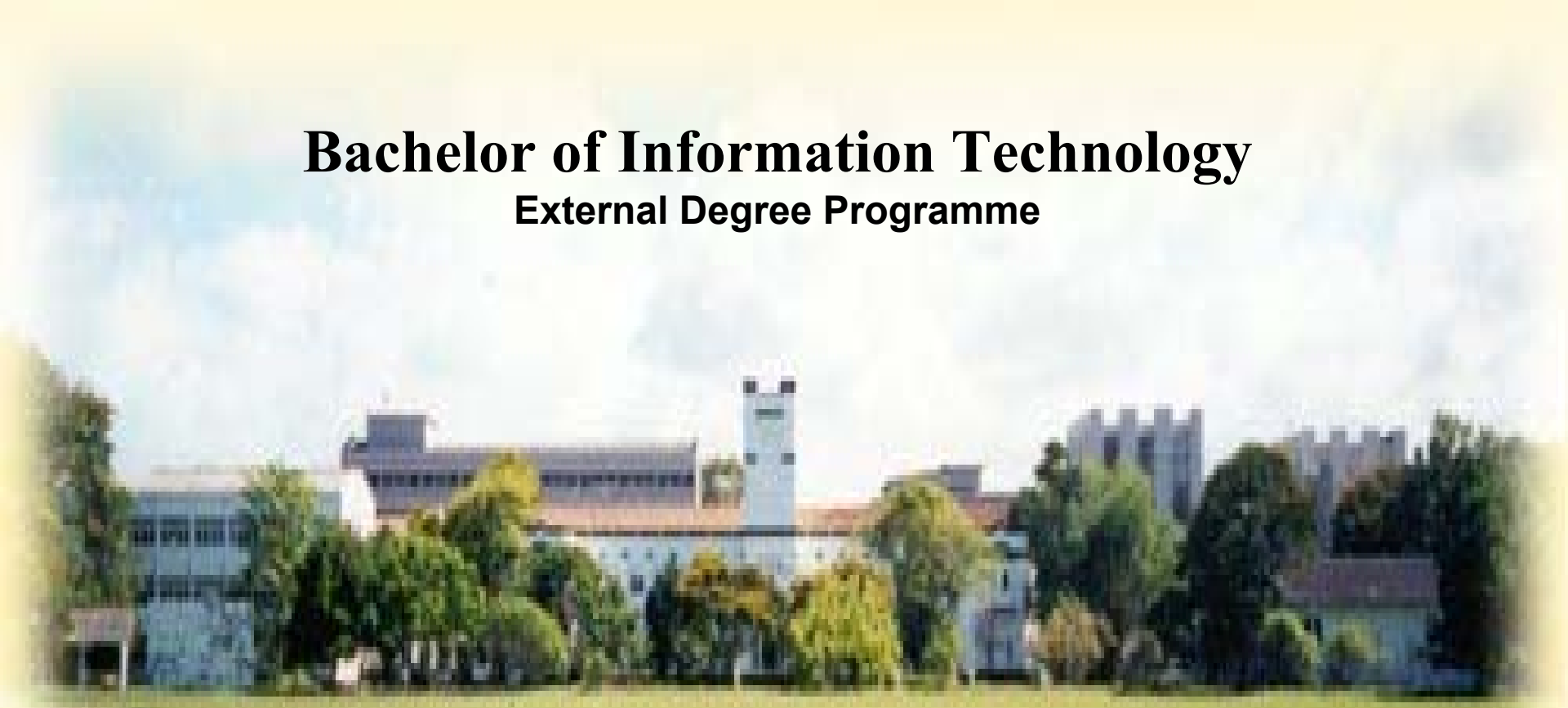
Class Diagram,
Use Case Diagrams etc.

Tools

Rational Rose,
Rational Suit

Bachelor of Information Technology

External Degree Programme



Systems Analysis and Design

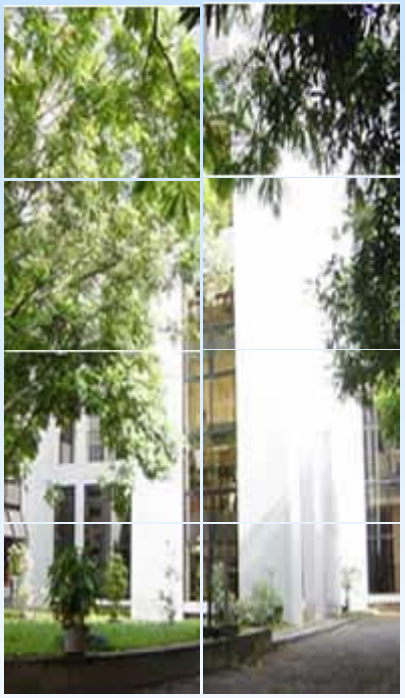
Ref_1 : System Analysis and Design Methods **By** Whitten, Bentley, Dittman ISBN 0-07-044539-7 (5th Edition)
Ref_2 : System Analysis and Design **By** Igor Hawryszkiewycz (4th Edition) ISBN 81-203-1670-3
Ref_3 : Practical SSADM Version 4 A Complete Tutorial Guide **By** Philip L Weaver ISBN 0-273-60095-8

UCSC

University of Colombo School of Computing

BIT

Introduction to Problem Definition



UCSC

BIT

Introduction

Problem Definition

- It is the activity of identifying the problem, understanding the problem (including causes and effects), and
- Understanding any constraints that may limit the solution

There are 3 important factors to be considered when defining a problem.

1. **Project goal**: What the users expect the system to do?
2. **Project bounds**:
   Set the project boundaries

3. **Resource limits:**
   Specify the resources to be available to the project

UCSC

BIT

Introduction….

- Most important and first questions to ask in systems work are

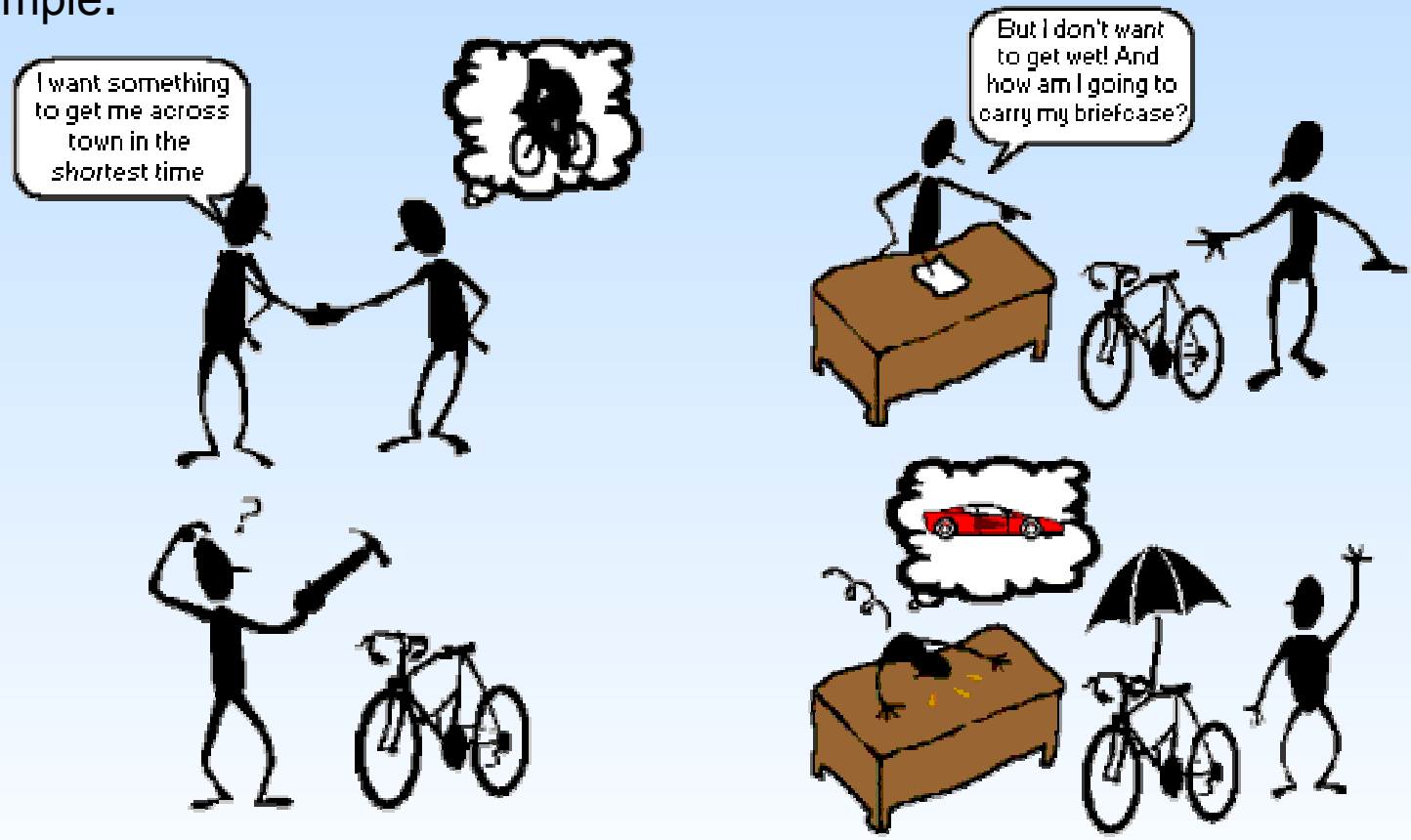
- We must then justify that solving the problem using computers is useful to the business and not just interesting.
- This justification is made to Ourselves and More importantly to users

Introduction….

Why we need to define the problem?

If we solve the wrong problem……

Example:



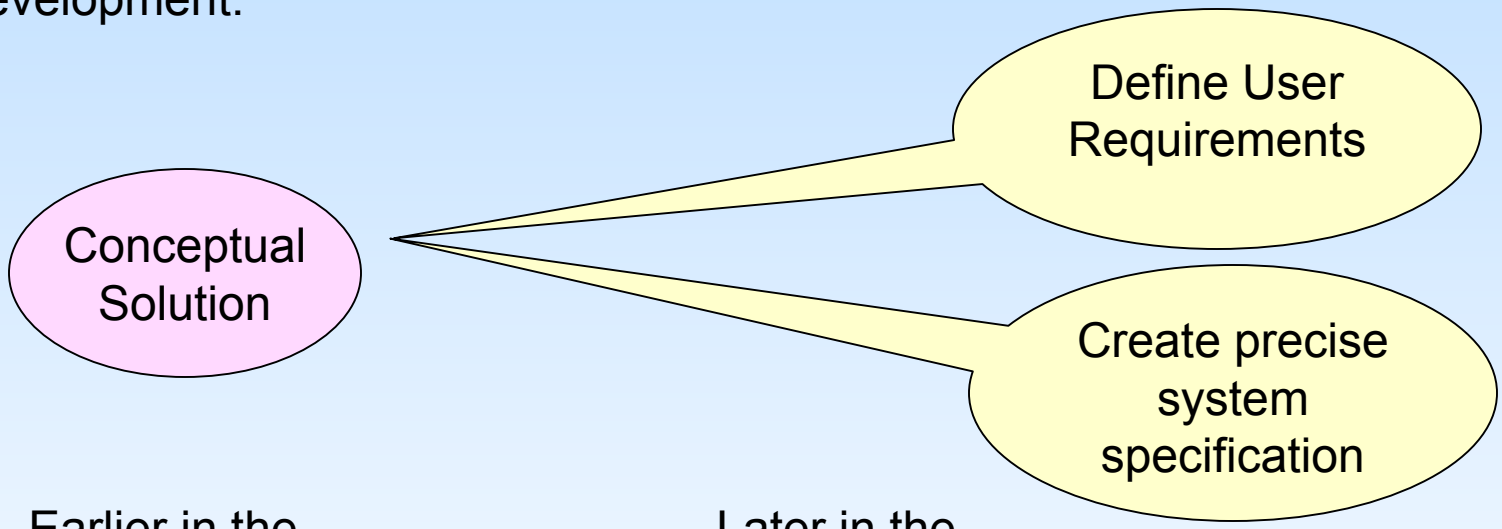
UCSC

BIT

Introduction….

- Solutions in this stage are broad in concept and are sometimes known as the **conceptual solution**

- Agreeing on a conceptual solution is often seen on the first phase of any development.



For more details refer - Ref_2: page 64

How to find problems to Solve?

Finding Problems Using Informal Methods

Eg. listening to what people say
discussion at a conference or meeting
want to do something that some one else is doing

In any case we often compare what is happening now to what we think should be happening.

We get ideas about what should be happening both **internally** and **externally**.

UCSC

BIT

How to find problems to Solve?....

Finding problems using External considerations

Comparing our activities with a competitor's activities

Analyzing community attitudes and changes to government policy

Using normative models

Using historical models

Etc.

How to find problems to Solve?....

Finding problems using Internal considerations

- Goals must be developed for the organization within the practical bounds
  - Project goal ⟶ sub goals
  - sub goals used to guide detailed analysis and design in later stages
  - Project goals will remove deficiencies in the existing system

✓ Deficiencies are found in ⟶ Examining documents about performance

↓

Interviews

Eg. Deficiencies
- Unsatisfactory performance
- Etc.

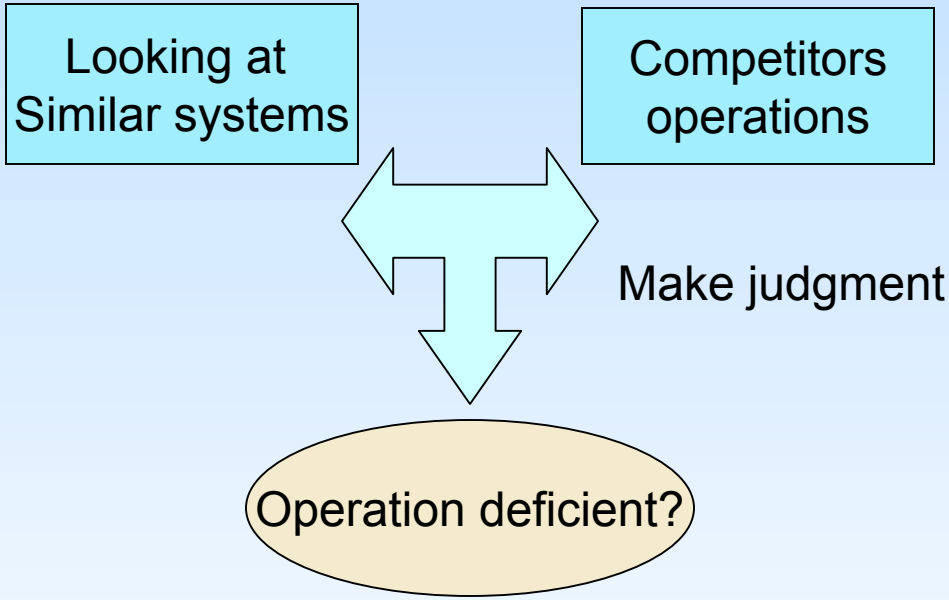
✓ When to judge an operation as deficient?

UCSC

BIT

How to find problems to Solve?....

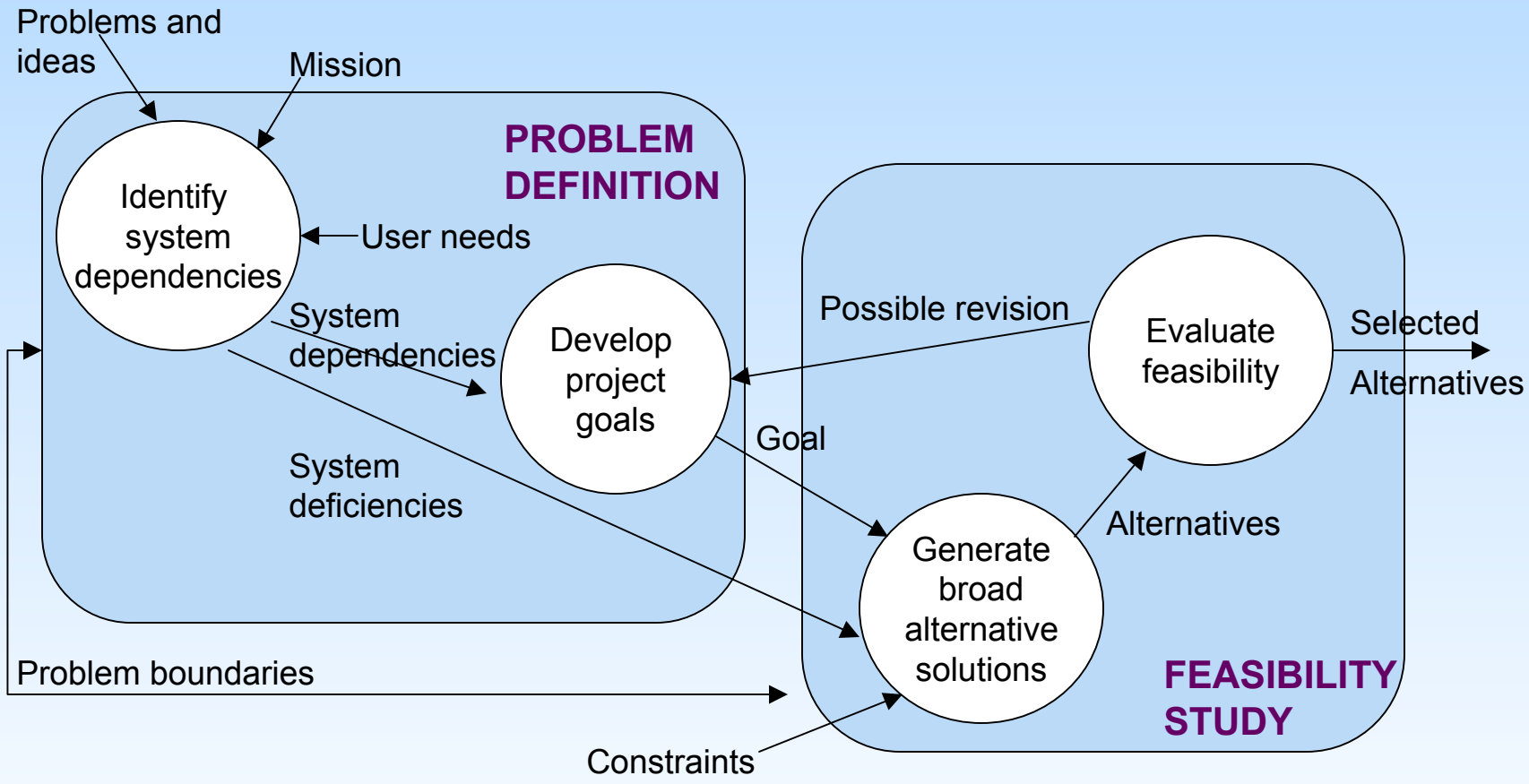
How to solve it?



For more details refer - Ref_2: pages 65-68

Feasibility Study

Introduction



UCSC

BIT

Feasibility Study….

Introduction….

Feasibility

Measure of how beneficial or practical to develop an information system for an organization

Feasibility Analysis

The process to measure feasibility

Feasibility should be measured throughout the life cycle.

Feasibility Study….

Introduction….

Feasibility Checkpoints in the Life Cycle

- **Systems Analysis**

  — The first feasibility analysis is conducted during the preliminary investigation phase. Find the first-cut estimate of development costs.

- **Systems Analysis**

  — The next checkpoint occurs after a more detailed study and problem analysis of the current system

- **Systems Design**

  — Problems and requirements should be known by now. Alternative solutions are defined in terms of Input/output methods, data storage methods, etc. Range of options like leave the current system alone, reengineer the manual system, etc. are evaluated by the analyst. Each option is analyzed for feasibility.

UCSC

BIT

Feasibility Study….

Tests for Feasibility

- **Technical feasibility.**
  Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?

- **Operational feasibility**.
  Will the solution fulfill the users' requirements? To what degree? How will the solution change the users' work environment? How do users feel about such a solution?

- **Economic feasibility**.
  Is the solution cost-effective?

- **Schedule feasibility**.
  Can the solution be designed and implemented within an acceptable time period?

UCSC

BIT

Feasibility Study….

Tests for Feasibility….

Operational Feasibility covers two aspects

- Is the problem worth solving? or will the solution to the problem work?
- How do the end-users and management feel about the problem (solution)?

UCSC

BIT

Feasibility Study….

Tests for Feasibility….

Technical feasibility

- Is the proposed technology or solution practical?
- Do we currently posses the necessary technology?
- Does the organization have the expertise to use it?

Schedule feasibility

Given our technical expertise, are the project deadlines reasonable?

UCSC

BIT

Feasibility Study….

Economic feasibility

Economic feasibility concerns returns from investments in a project

- Will it be recovered?
- Is it worth while to invest?

Necessary to place actual money values against any:

- Purchases needed to implement the system
- Activities created by the project
- Benefits that will acquire from a new system

Such calculations are often described as **cost benefit analysis**

There are popular techniques to Access Economic Feasibility

For more details refer - Ref_1: pages 367-370

UCSC

BIT

Cost Benefit Analysis

Determining whether a project is worthwhile

Economic feasibility has been defined as a cost-benefit analysis. Costs and benefits are used to determine whether the project is economically feasible

Costs

Costs associated with Operating the system

Recur throughout the lifetime of The system

- Fixed costs

Eg: Software License payments

- variable costs

Eg: Supplies

Costs associated with developing the system

Onetime costs that will not recur after project Has been completed

Eg:
- Equipment cost for the new system
- Personnel costs
- Computer usage
- Training costs
- etc.

UCSC

BIT

Cost-Benefit Analysis….

Benefits

Intangible

- Benefits that are difficult to quantify

Eg. Benefit of maintaining a good business image

Tangible

- Those that can be easily quantified

Eg. Reduced processing cost (using computers)

Techniques to Access Economic Feasibility

- Payback Analysis
- Return On Investment
- Net Present Value

For more details refer - Ref_1: pages 370-373

UCSC

BIT

System Proposals

Feasibility Analysis of Candidate Systems

The system analyst identifies candidate systems solutions and then analyze Those solutions for feasibility

For this there are two ways

- **Candidate system matrix**

  Documents similarities and differences between candidate systems; however, it offers no analysis

- **Feasibility Analysis Matrix**

  Complements the candidate systems matrix with an analysis and ranking of the candidate systems

These matrix are useful for presenting candidates and recommendations to management

System Proposals….

Feasibility Analysis of Candidate Systems

Technologically able?

Yes

No

Analyzing candidate solutions

Can

Organization

Can I accommodate?

You can

Can we have enough background?

System Proposals….

Feasibility Analysis of Candidate Systems….

Feasibility Analysis Matrix Template

Cells contain the feasibility assessment notes for each candidate

	Candidate 1 Name	Candidate 2 Name	Candidate 3 Name
Description			
Operational Feasibility			
Technical Feasibility			
Schedule Feasibility			
Economic Feasibility			
Ranking			

System Proposals….

Producing a system proposal

- Deliverable is usually a formal written report or oral presentation
  Intended for system owners and users

Formal written report

Oral presentation

System owners
And
End Users

- System analysts should be able to write a formal business report and make a business presentation without getting into technical issues and alternatives.

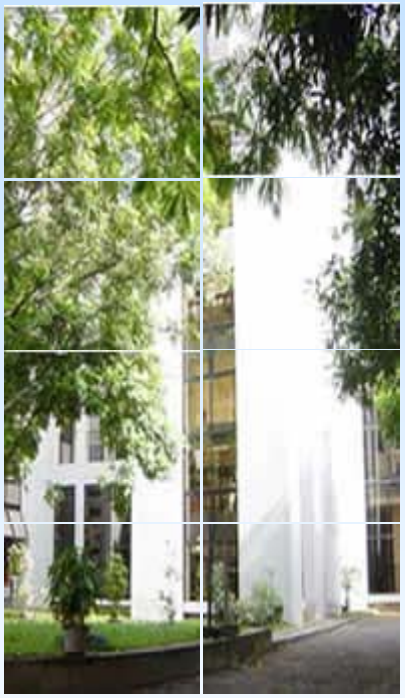
- Size of the report depends on the management level

For more details refer - Ref_1: pages 377-383

UCSC

BIT

Requirements Analysis



UCSC

BIT

Identifying Requirements

- Correct systems can only be built if you know exactly

- Therefore most important factors in building correct systems is to first clearly define what the system must do

- **System Requirements**

Defines the services the system is to provide and prescribe constraints for its operation

- Functional Requirements
- Nonfunctional Requirements

Identifying Requirements….

- System requirements can be developed
  - by discussing with users their requirements
  - building systems that satisfy these requirements

- **Importance of Communication**
  - Analyst must ensure that no ambiguities arise in discussions between various people involved in analysis
  - Different jargon use by different people may cause problems

Example: Ambiguous Requirement Statement

Identify a transportation to transfer a single individual from home to place of work

Management Interpretation

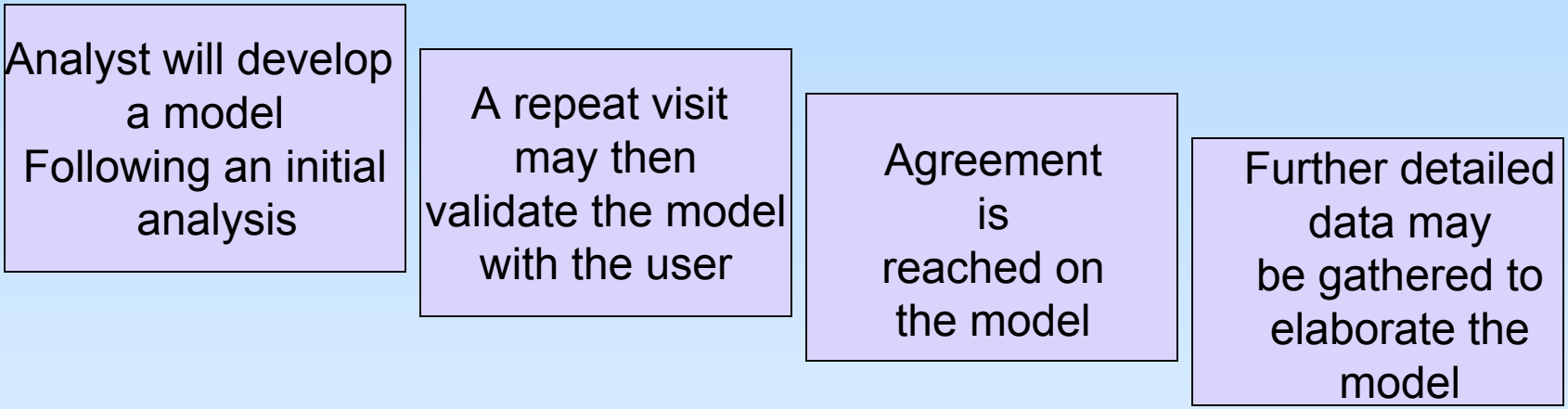
IT Interpretation

User Interpretation

UCSC

BIT

Identifying Requirements….



This iterative approach serves a number of purposes:

- there is always a record of information gathered to date
- ensures correctness of the information as you continually verify the results with the user.
- Analyst does not get too far ahead using wrong assumptions.

Requirements Discovery Methods

Requirements Discovery

Includes techniques to be used by systems analysts to identify or extract system problems and solution requirements from the user community.

Requirement discovery for a system depends on the analysts' ability to first discover and then analyze problems and opportunities that exist in the current system

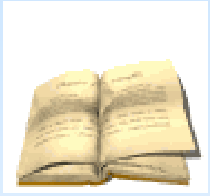
System Analyst

Requirements Discovery Methods..

Fact-Finding

It is the formal process of using techniques to collect information about systems requirements etc. It is also called ***Information gathering***

Fact-Finding Methods



Sampling of Existing documents

Research and site visits

Observations of the work environment

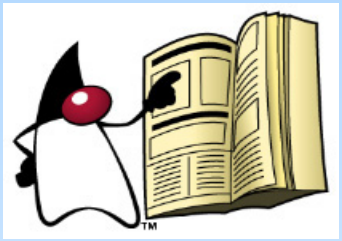
Questionnaires

Interviews

Joint requirements planning

Prototyping

Requirements Discovery Methods..



Sampling of existing Documentation

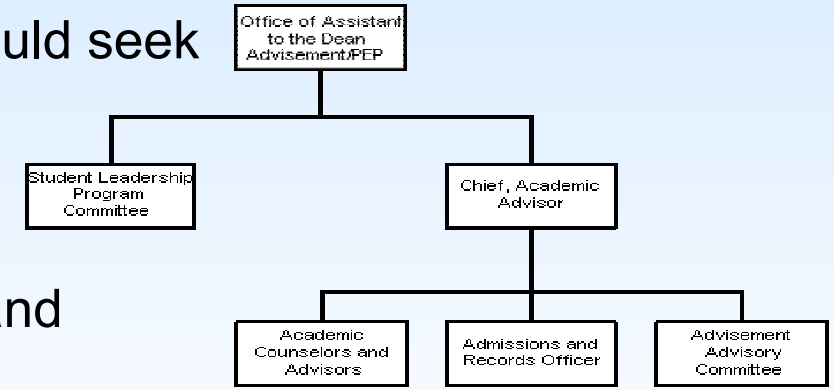
- When you are studying an existing system, you can get a good idea by studying existing

Documentation

Forms

Files

- First document that analyst should seek out is the organizational chart



- Be sure that they are relevant and up-to-date

UCSC

BIT

Requirements Discovery Methods..

Research and Site Visits

- Thoroughly research the application and problem.

Internet

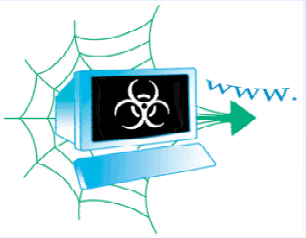
Intranets

Computer trade Journals

Reference books

Good sources of information

World Wide Web



- Identify the material that are relevant and reliable

UCSC

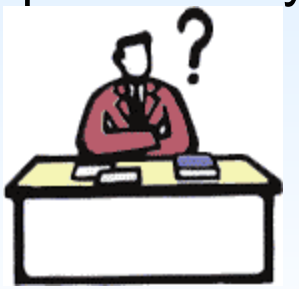
BIT

Requirements Discovery Methods..

Observations of the work environment

- Systems Analyst participates in or watches a person perform activities to learn about the system

- often used when validity of data collected through other methods is in question *or* when the complexity of certain aspects of the system prevents a clear explanation by the end users.



UCSC

BIT

Requirements Discovery Methods..

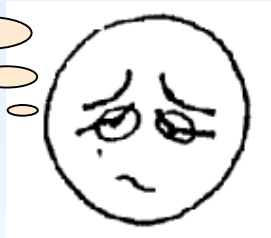
Observations of the work environment…

Advantages

- Data gathered by observation can be highly reliable
- Relatively inexpensive
- Allows system analyst to do work measurements
- Etc.

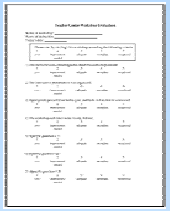
Disadvantages

- People usually feel uncomfortable when being watched.
- Work being observed may not involve the level of difficulty or volume normally experienced during that time
- Etc.



Requirements Discovery Methods..

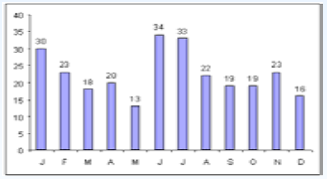
Questionnaires



Questionnaires are special purpose documents that allow the analysts to collect information and opinions from a large audience.

Advantages :

- Most questionnaires can be answered quickly
- Allow individuals to maintain anonymity
- Relatively inexpensive way of gathering data.
- Responses can be tabulated and analyzed quickly *etc.*



UCSC

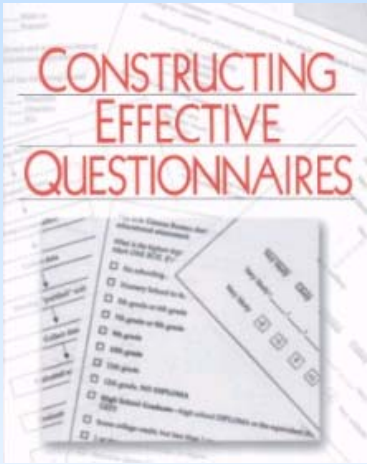
BIT

Requirements Discovery Methods..

Questionnaires…

Disadvantages:

- The number of respondents is often low
- Mostly suited for close questions
- No guarantee that an individual will answer or expand on all the questions
- Good Questionnaires are difficult to prepare
- No immediate opportunity to clarify a vague or incomplete answer to any question.
- Etc.



UCSC

BIT

Requirements Discovery Methods..

Questionnaires…

Types of Questionnaires

- *Free-format* : A question is asked, and the respondent records the answer in the space provided after the question.
- *Fixed-format* : contains questions that require specific responses from individuals

There are 3 types of fixed-format questions

1. *multiple-choice questions*
   -Given several answers to select one. Eg. Yes, No type
2. *rating questions*
   -Given a statement and asked to use supplied responses to state an opinion.
3. *ranking questions*
   -Given a several possible answers to be ranked in order of preference or experience

UCSC

BIT

Requirements Discovery Methods..

Interviews

- Most commonly used technique in analysis
- Systems analyst collects information from individuals face to face.
- Systems Analyst must possess good human relations skills for dealing effectively with different type of people

Can be used to achieve any of the following goals:

find facts, verify facts, clarify facts, generate enthusiasm, get the end-user involved, identify requirements, and solicit ideas and opinions.

UCSC

BIT

Requirements Discovery Methods..

Interviews…

Advantages

- Gives the analyst an opportunity to motivate the interviewee to respond freely and openly to questions.
- Allow the analyst to look for more feedback from the interviewee.
- Permit the analyst to ask questions from each individual etc.
- New ideas may arise

UCSC

BIT

Requirements Discovery Methods..

Interviews…

Disadvantages

- Very time consuming. Therefore costly approach.

- Success of interviews is highly dependent on the systems analyst's human relations skill.

- Interviews may be impractical due to the location of interviewees etc.

UCSC

BIT

Requirements Discovery Methods..

Interviews…

Types of Interviews

- Unstructured interviews..
- Structured interviews..

Types of Interview Questions

- Open-ended questions…
- Closed-ended questions…

Requirements Discovery Methods..

Interviews…

How to conduct an *Interview*?

Select Interviewees

- Interview the end users of the information system you are studying.
- A formal organizational chart will help you identify these individuals and their responsibilities.

- Always make an appointment with the interviewee.
- Higher the management level of the interviewees, less time should be spent.

UCSC

BIT

Requirements Discovery Methods..

Interviews…

How to conduct an *Interview*?...

Prepare for the Interview

Prepare an *interview guide* - checklist of specific questions interviewer will ask the interviewee

Avoid the type of questions such as:

- *Loaded* questions (Do you need to include both of these columns for this report?)
- *Leading* questions (You are not going to use this operator code, are you?)
- *Biased* questions (How many codes do we need for food classification in the inventory file? I think 20 should cover it ?)

UCSC

BIT

Requirements Discovery Methods..

Interviews…

How to conduct an *Interview*?...

Prepare for the Interview…

Interview question guidelines :

- Use clear and concise language
- Don't include your opinion as part of a question
- Avoid long or complex questions
- Avoid threatening questions
- verify before you leave

The purpose of the interview is to investigate, not to evaluate or criticize

UCSC

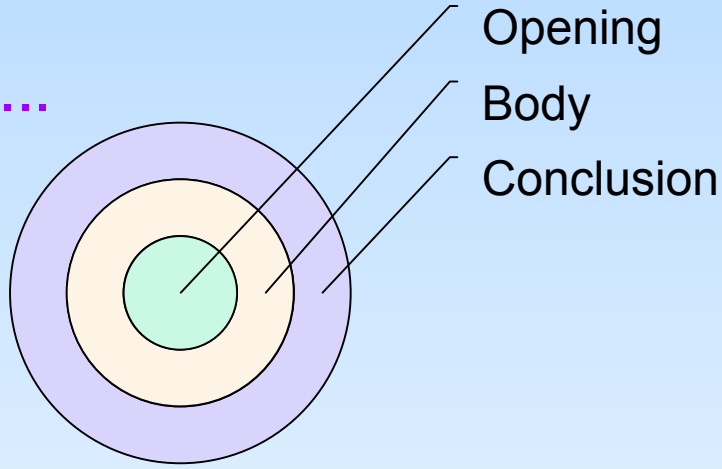
BIT

Requirements Discovery Methods..

Interviews…

How to conduct an *Interview*?...

Conduct the Interview



The actual interview consist of three phases:

Interview Opening : Intended to influence or motivate the interviewee to participate

Interview body : Obtain interviewee's response to your list of questions

Interview conclusion : Express your appreciation. Important for maintaining good relationship and trust.

Requirements Discovery Methods..

Prototyping

- A method used to test or illustrate an idea and build a system in an explorative way.
- Used to discover user requirements
- Allows analyst to quickly create mock forms and tables to simulate the implemented system.

UCSC

BIT

Requirements Discovery Methods..

Prototyping

Advantages

- Allow users and developers to experiment with the software and develop with an understanding
- Helps to determine feasibility and usefulness of the system
- Minimize the time spent for fact-finding and help define more stable requirements. Etc.

Disadvantages

- Developer may need to be trained in the prototyping approach
- Prototype can only simulate system functionality and are incomplete in nature. Etc.

Requirement Discovery Methods…

Joint Requirement Planning (JRP)

Highly structured group meeting are conducted to analyze problems and define requirements. JRP is a subset of a more comprehensive joint application development or JAD technique

JRP Participants

- Sponsor

  Serve as JRP champion. Single person who is in top management.

- Facilitator

  Single individual who plays the role of the leader or facilitator.

UCSC

BIT

Requirement Discovery Methods…

Joint Requirement Planning (JRP)…

JRP Participants...

- User and Manger

  Number of participants from the user and management.

- Scribes

  Those who are keeping responsible for keeping records pertaining to everything discussed in the meeting.

- IT Staff

  IT personal who primarily listen and take notes regarding issues and requirements.

For more details refer - Ref_1: page 213-239

UCSC

BIT

Document Analysis

Documents of a System

Documentation is both a communication tool and a management tool.

- It is a communication tool :
  - because it contains a repository of all work done to date and makes it available to all persons working on related parts of a large project.
  - Such a repository can prevent unnecessary repetitions when someone leaves the project team.
  - Proper documentation ensures that all the information developed about the system is always available to new people joining the project.

UCSC

BIT

Document Analysis…

Documents of a System…

- Documentation is also a management tool.
- It supports management in two ways:
  - gives access to the latest work to all project personnel and thus reduces the chance of work having to be repeated.
  - is the only project deliverable, specially in the early project phases, and thus serves to determine project status and progress.

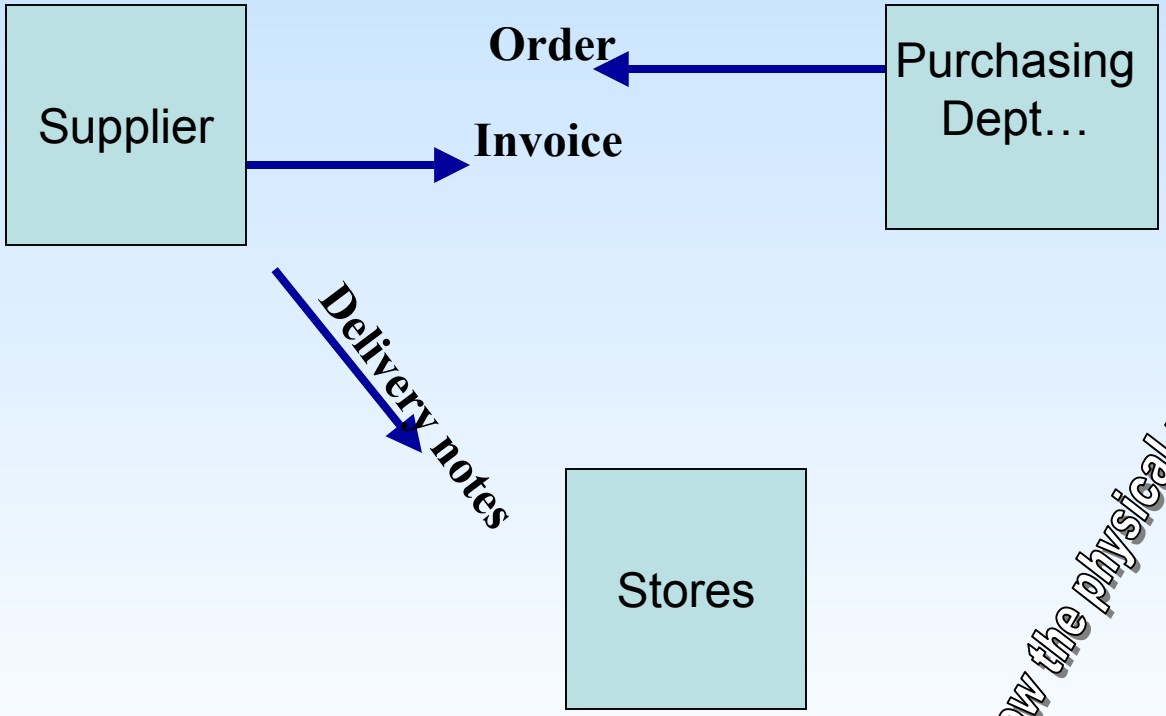
  Is also a part of the phase output.

Document Analysis…

Document Flow Diagrams

Used to examine the flow of documents within the existing system.

Example:



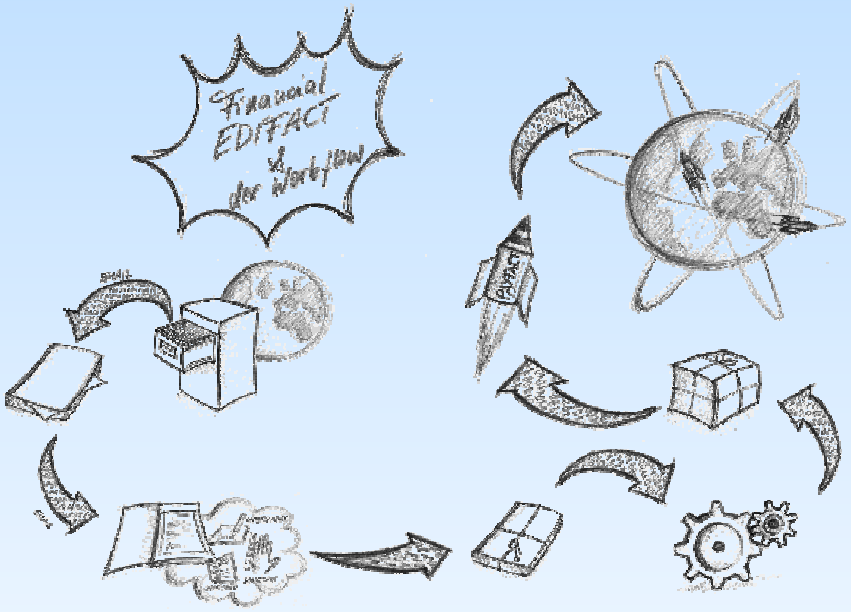
Show the physical movement of the ducuments

Document Analysis…

Document Flow Diagrams…

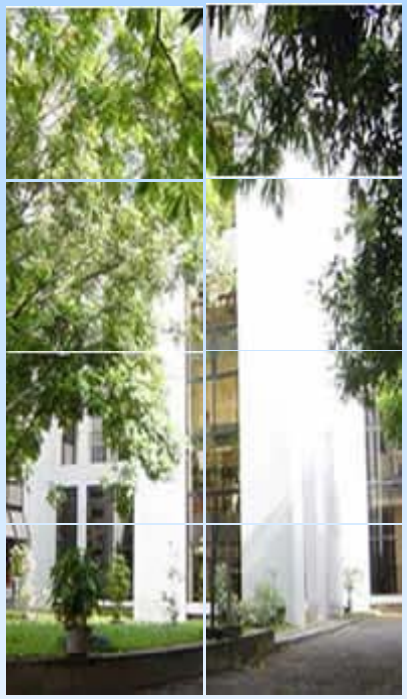
Advantages / Usefulness

- Used to identify the documents in the system
- Identify the flow of document
- To understand the workflow of the existing system
- Used to define the system boundary
- Used to draw Data Flow Diagrams by further analyzing
- Etc..



For more details refer - Ref_4: pages 29-34

Modeling Methods



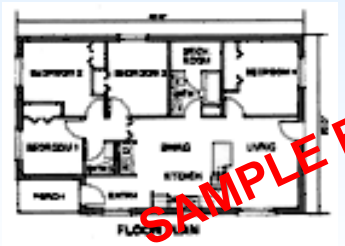
Modeling Methods

How to structure the problem?

The answer is just Simple, use **MODELS**

Model

Is a presentation of reality. Just a picture is worth a thousand of words, most system models are pictorial representations of reality.



SAMPLE FLOOR PLAN

UCSC

BIT

Models

Logical Models

Show what a system is or does?

Other names:
Essential model
Conceptual model
Business model

Physical Models

Show not only what a system is or does? But also how the system is implemented.

Other names:
Implementation Model
Technical model

Process Modeling

Process Modeling

Is a technique for organizing and documenting the structure and flow of data through a system's Processes , policies, and procedures to be implemented by a system's Processes.

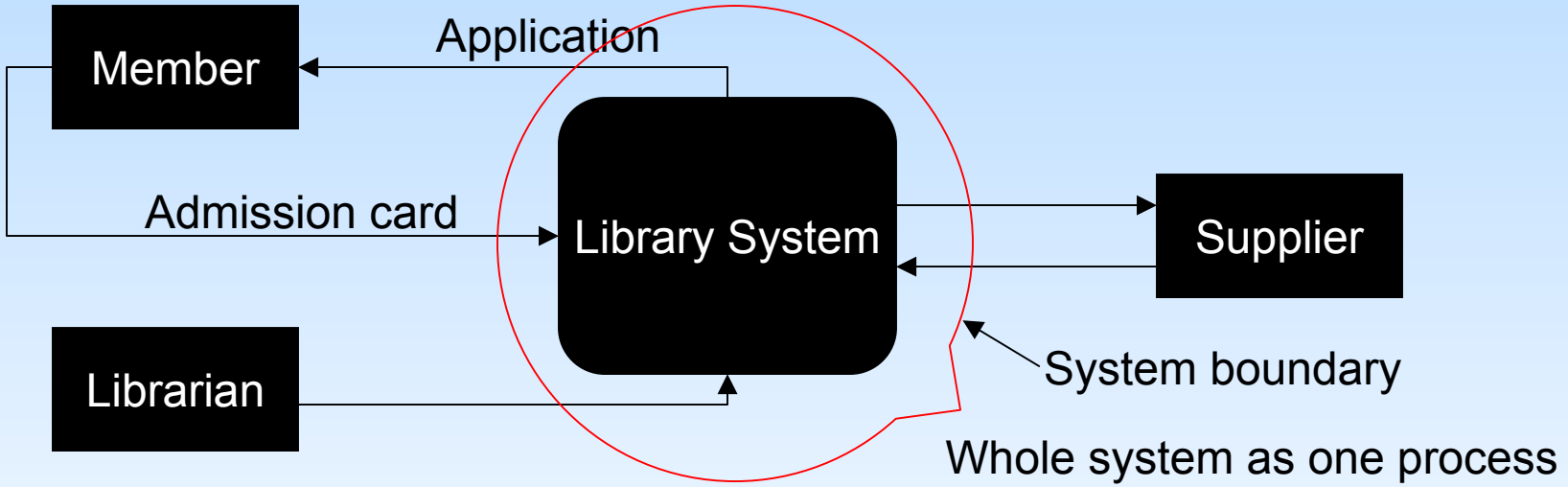
Context Diagram

Illustrate the system's interface to the business and outside world, including other information systems.

Process Modeling…

Context Diagram…

Shows the major data flows into and from the application, and the system boundaries.



E.g. Context Diagram of a Library System

Process Modeling…

Context Diagram…

- Shows the top level function defined to cover the scope of the application. It also shows the major data flows into and from the application, and the system boundaries.

- Top level function is then decomposed to its component functions (5-9 typically).

Process Modeling…

Process Models

DFD (data flow diagram)

Data flow diagram (DFD)

DFD is a tool that depicts the flow of data through a system and the processing performed by that system.

Document Flow Diagram can be the Starting point

Supplier

Order

Invoice

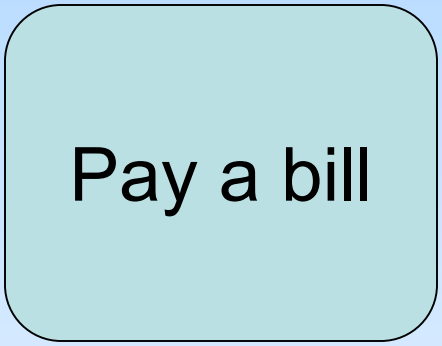
Purchasing Dept.

Delivery notes

Stores

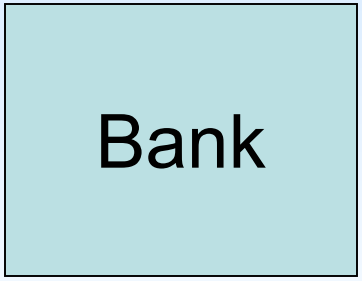
Process Modeling…

Gane and Sarson Symbols



Rounded rectangle

Represent processes or work to be done. E.g.: pay a bill, withdraw money, etc..



Square

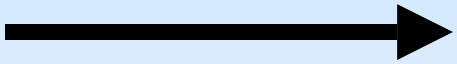
Represent external agent. E.g.: bank, supplier, creditor, etc..

Process Modeling…

Open end box

Represent data store which is some times called files.

E.g.: bank, supplier, creditor, etc..



Arrows

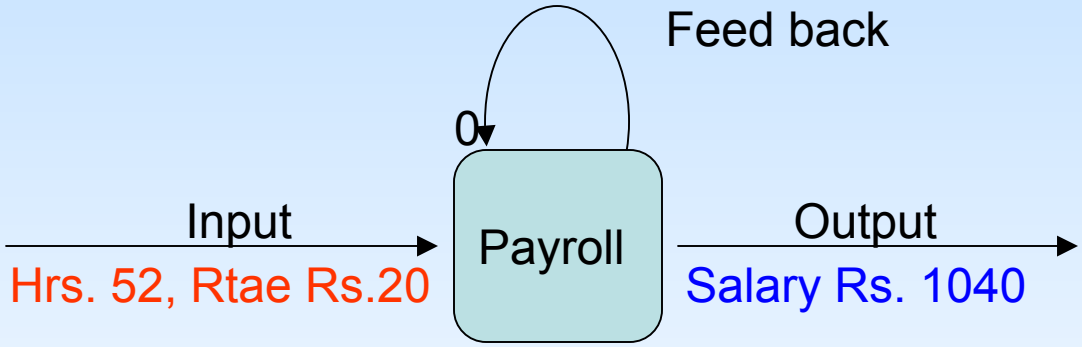
Represent data flows or input and output. E.g.: payment, bill, withdraw, etc

The above notation is called “Gane and Sarson”. There are several competing symbols set for DFDs.

Process Modeling…

process

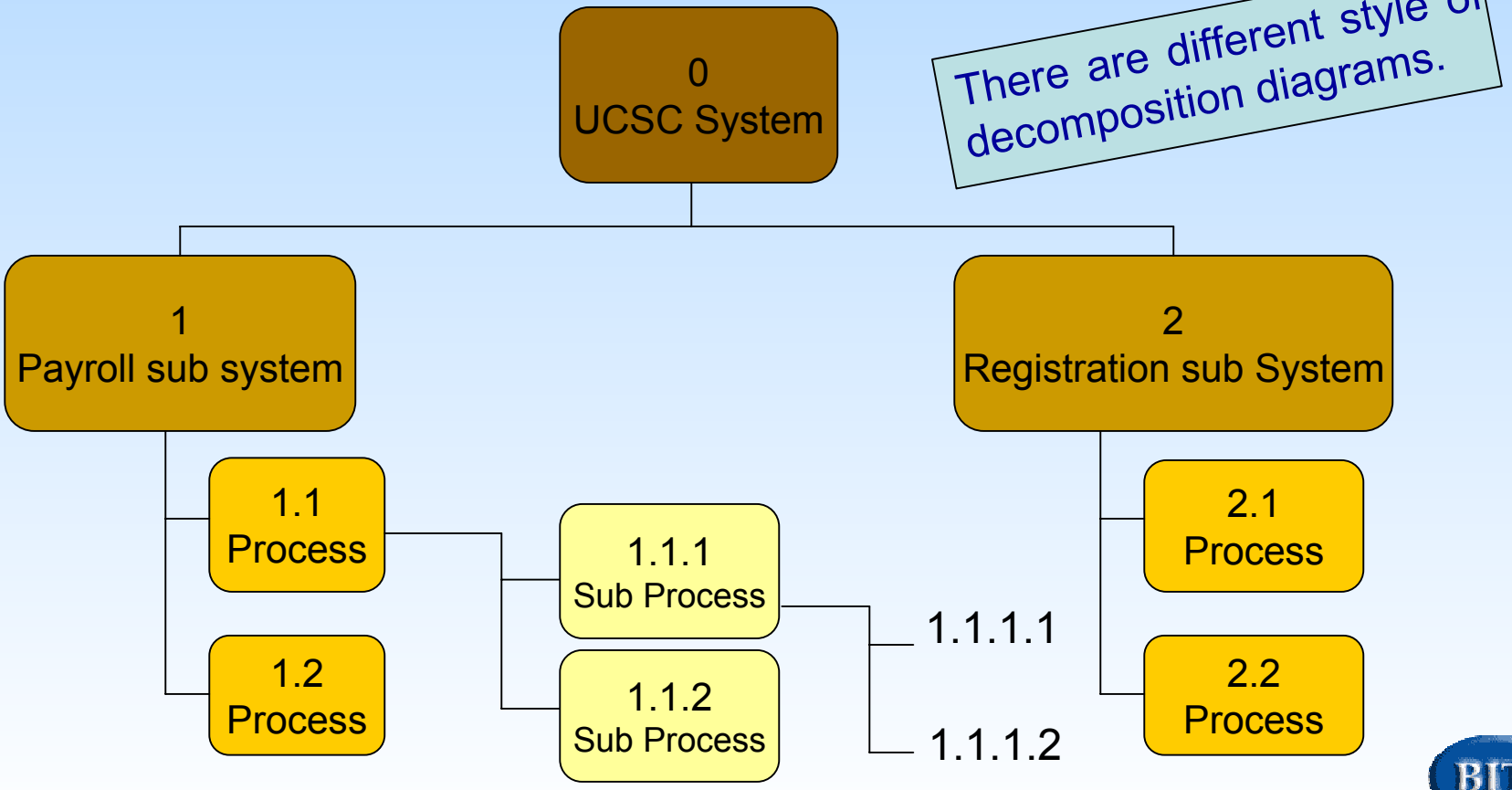
A **process** is work performed on, or in response to, incoming data flows or conditions.



Process Modeling…

Process Decomposition

Is a act of breaking a system into its component subsystems, processes, and sub processes.



Process Modeling…

Process Decomposition…

- Advantages

  Improved…

  - Communication
  - Analysis
  - Design.

Decomposition continues until the functions reach ‘elementary’ level. (cannot be further decomposed)

Process Modeling…

Example:

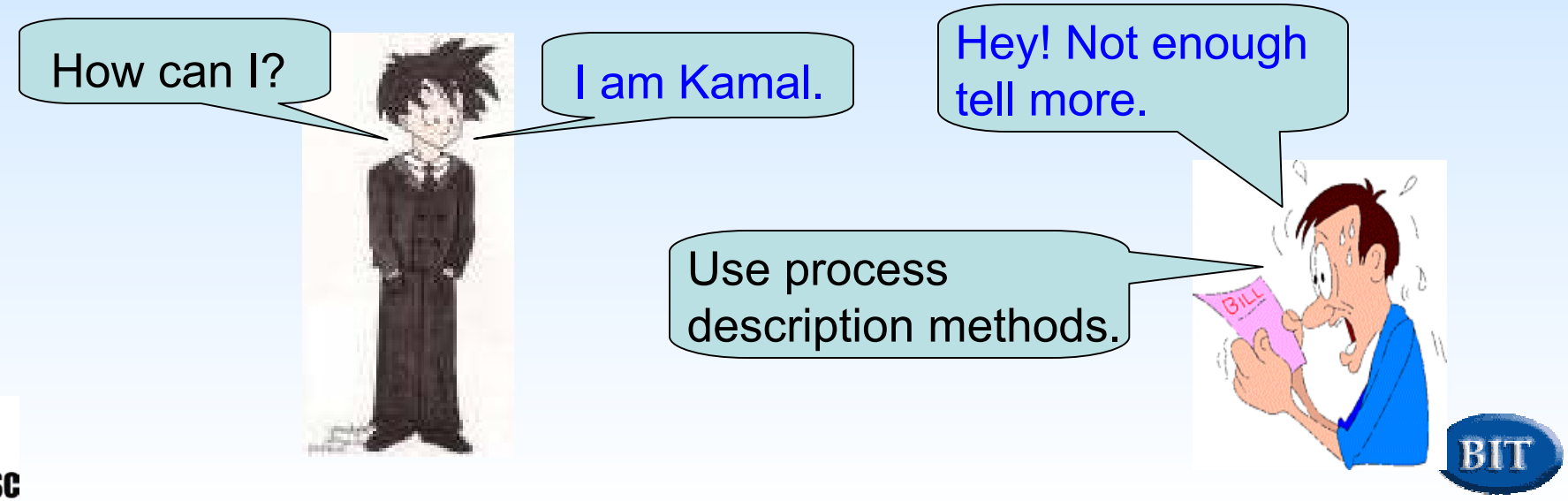
- Consider a system in a bank whereby account holders get their withdrawals affected.
- Whenever an account holder wants to withdraw some cash, he presents a cheque or withdrawal slip.
- The account is checked for appropriate balance.
- If balance exists, the cash is paid and the account is updated.

Process Modeling…

Process Description

The name of the process is insufficient to explain the logic to implement the process.

Processes in the system specification should be specified in such a way that they can be converted to computer program.



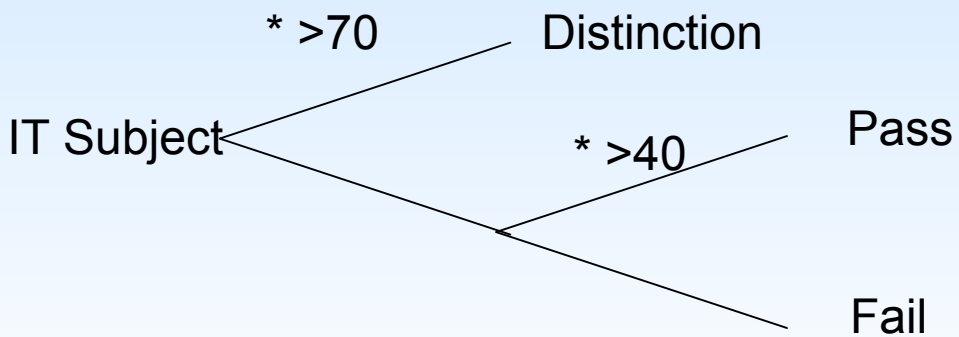
Process Modeling…

Process Description…

Methods of process description

- **Structured English**
- **Decision Tree**
- **Decision Table**

If my IT marks more than 70 then grade is Distinction
Else if mark more than 40 then Pass
Else Fail



Process Modeling…

Process Descriptions…

Structured English

Syntax is very similar to block structured languages.

Example:

IF credit limit exceeded
THEN
IF Customer has bad payment history
THEN refuse credit
ELSE
IF PURCHASE ABOVE Rs.1000/=
THEN refuse credit
ELSE refer to manager
ELSE allow credit

UCSC

BIT

Process Modeling…

Process Descriptions…

Decision Tree

Credit limit exceeded

Good payment History

Purchase above Rs 1000 — Refuse credit

Purchase below Rs.1000/= — Refer to Manager

Bad payment history — Refuse credit

Credit limit not exceeded — Allow credit

UCSC

BIT

Process Modeling…

Process Descriptions…

Decision Table

Condition	**Credit limit exceeded**	Y	Y	Y	Y	N	N	N	N
	Good payment history	Y	Y	N	N	Y	Y	N	N
	Purchase above Rs.1000/=	Y	N	Y	N	Y	N	Y	N
Action	Allow Credit					X	X	X	X
	Refuse	X		X	X				
	Refer Manager		X						

Y-TRUE N-NOT TRUE X-TAKE ACTION

Better way of describing logic involving multiple condition

Process Modeling…

Data flows

Data flows are the communication between processes and the system environment.

Data flow

Represent an input of data to a process or the output of data (or information) from a process

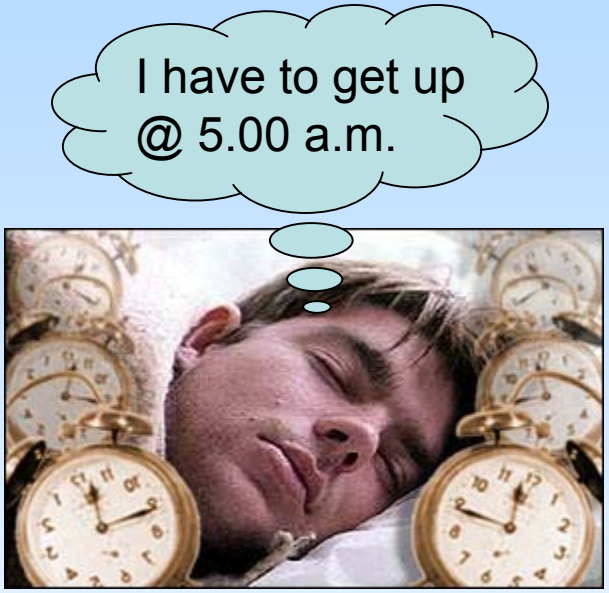
Data flow symbol

Data flow names should be descriptive noun and noun phrases that are singular. E.g. Order not Orders

Process Modeling…

Control flow

Represents a condition or non data event that triggers a process.

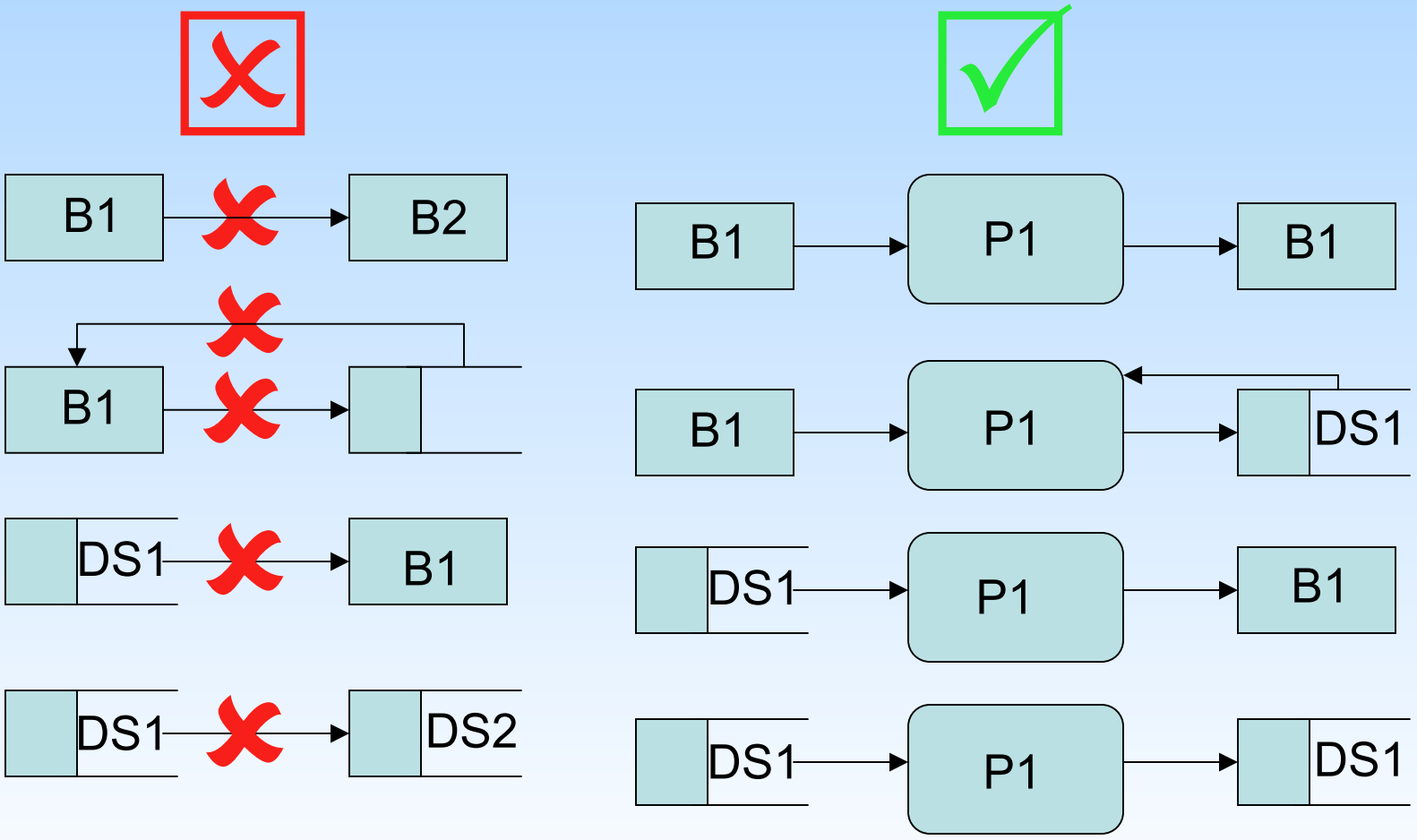


Control flow name

Control flow symbol

Process Modeling…

Illegal vs. legal data flow



Process Modeling…

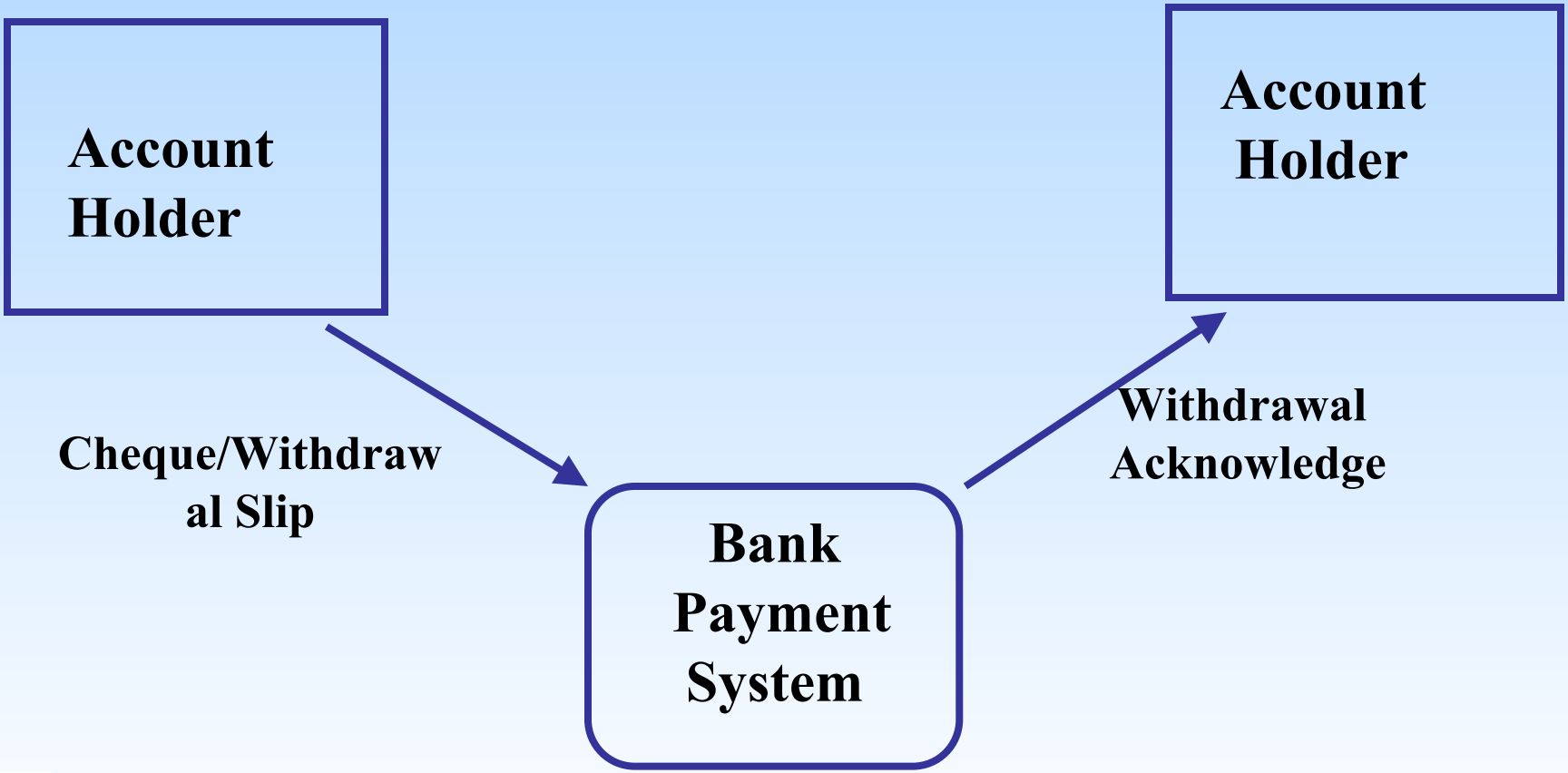
Example : DFD

- Consider a system in a bank whereby account holders get their withdrawals affected.
- Whenever an account holder wants to withdraw some cash, he presents a cheque or withdrawal slip.
- The account is checked for appropriate balance.
- If balance exists, the cash is paid and the account is updated.

Process Modeling…

Example…

Step 1 - Context Diagram

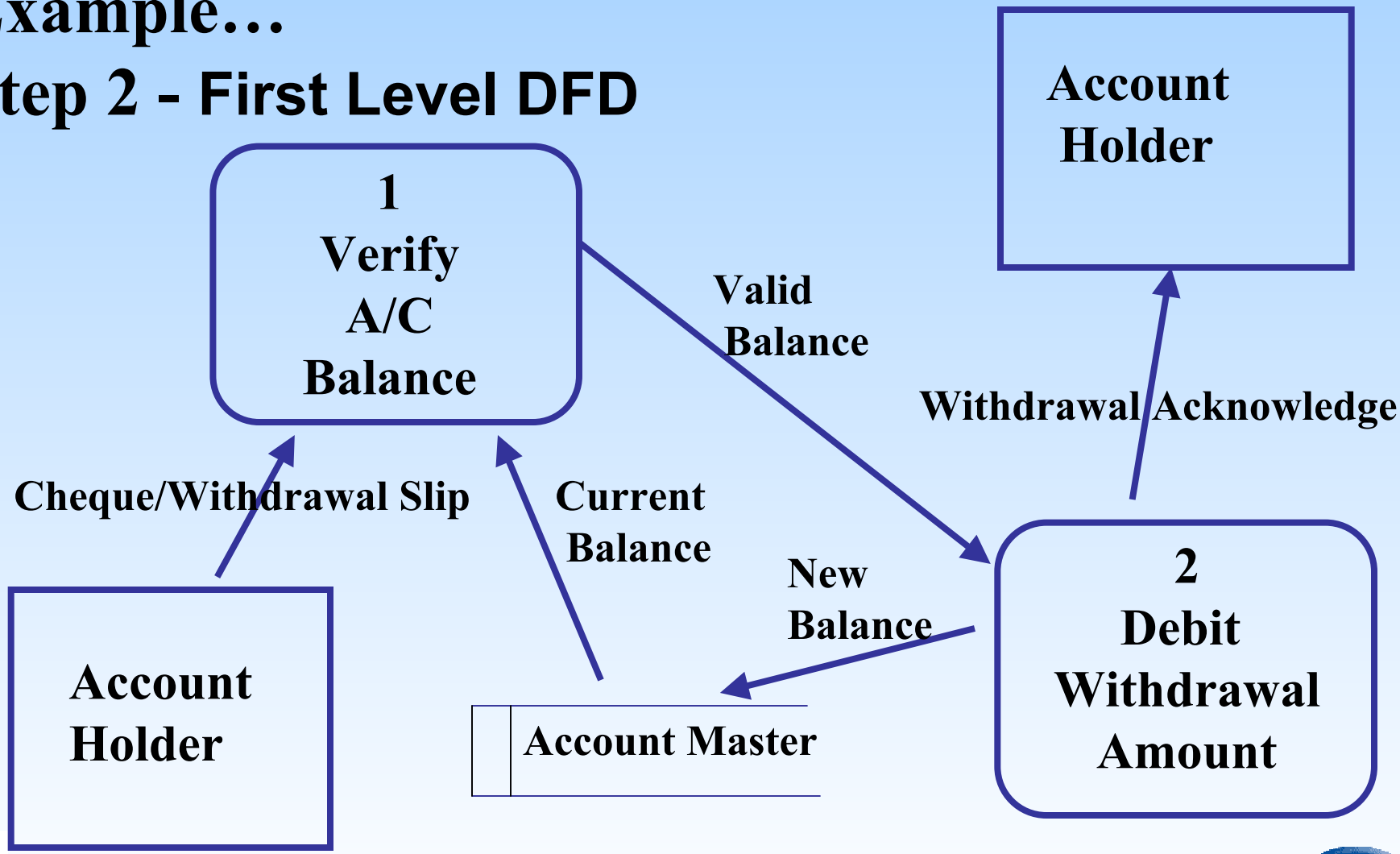


Using Gane and Sarson, notation

Process Modeling…

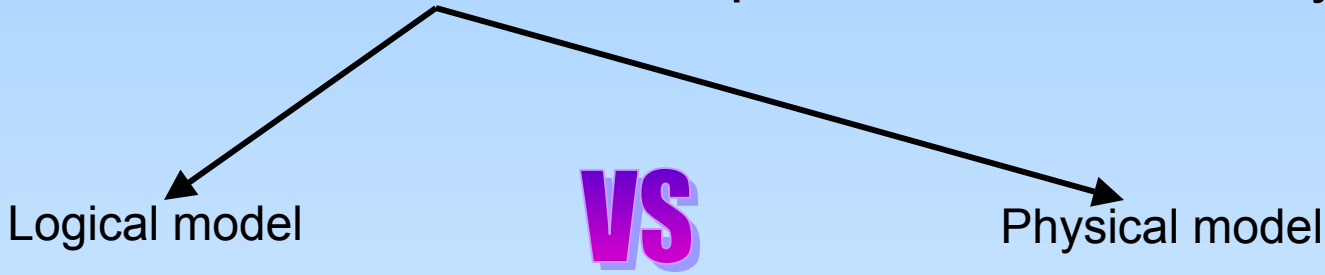
Example…

Step 2 - First Level DFD



Data Modeling

A **model** is a representation of reality.



- Logical functions describe any changes of value made by the processes on logical data.

- Whereas logical functions more closely mirror the subject world.

- Logical DFDs only illustrate what occurs without showing how it occurs.

- Physical-level functions are closer to the usage world (real world).

- Usually a physical device is used to transform data. E.g. Computer, person, etc.

- Physical DFDs show things happen, or the physical components.

Data Modeling…

Entity Relationship Diagrams

Depicts data in terms of the entities and relationships described by data.

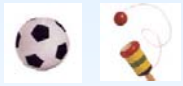
Entities

An entity is something about which the business needs to store data. Synonyms – entity type and entity class

Example:


Employee


Paris


Ball


Payment


Concepts

Data Modeling…

Entity Relationship Diagrams…

Entity:

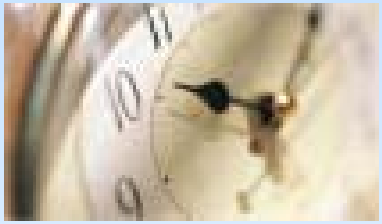
is a class of


Persons


Places


Objects


Events


Concepts

about which we need to capture and store data.

An entity instance is a single occurrence of an entity. Every entity must have an identifier or key to uniquely identify each instance.

UCSC

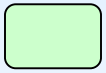
BIT

Data Modeling…

Entity Relationship Diagrams…

Symbol:

Consider Martin notations.

The named rounded rectangle represent the entity. – 

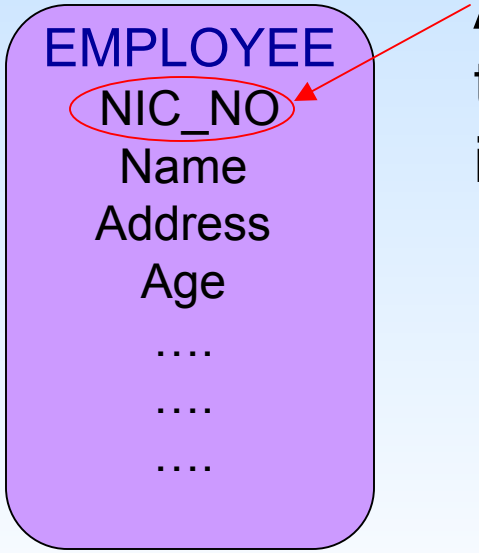
A line represent the relationship. – ————————

Data Modeling…

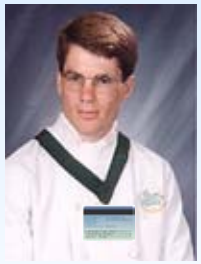
Entity Relationship Diagrams…

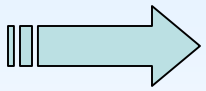
Attribute:

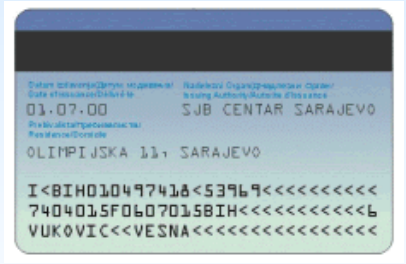
is a descriptive property or characteristics of an entity. Sometimes called as element, property, and field.



A key is an attribute, or group of attributes that assumes a unique value for each entity instance. It is some time called an *identifier*.







This person can be identified using his ID number.

UCSC

BIT

Data Modeling…

Entity Relationship Diagrams…

Compound Attribute is one that actually consist of other attributes.

Synonyms- composite attribute,
concatenated attribute
data structure.

Example : Address

Street Address
City
Country
Postal Code

Data Modeling…

Entity Relationship Diagrams…

The values for each **attribute** are defined in terms of three properties:

1. Data type – What type of data can be stored in that attribute.
2. Domain – What values an attribute can legitimately take on.
3. Default – Is the value that will be recorded if not specified by the user.

Data Modeling…

Person married relation

Entity Relationship Diagrams…

Relationships

Natural business association that exists between one or more entities

May represent an event that links the entities or logical affinity that exits between the entities.

E.g.. an EMPLOYEE assigns to a JOB

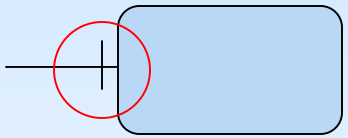
Data Modeling…

Entity Relationship Diagrams…

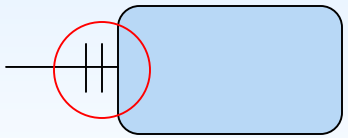
Cardinality

Defines the minimum and maximum number of occurrences of one entity that may be related to a single occurrences of the other entity.

Exactly one



or



Example:

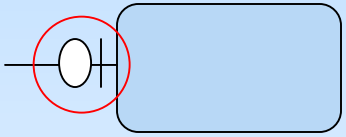
Person married relation

Data Modeling…

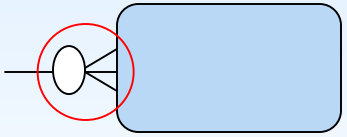
Entity Relationship Diagrams…

Cardinality

Zero or one



Zero, one or more



UCSC

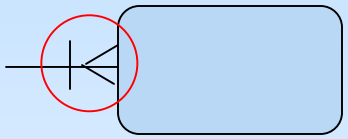
BIT

Data Modeling…

Entity Relationship Diagrams…

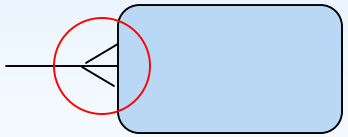
Cardinality…

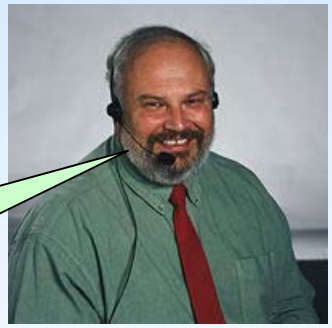
One or more



I have to work at least one, or more projects.

More than one





I am working on many projects.

Data Modeling…

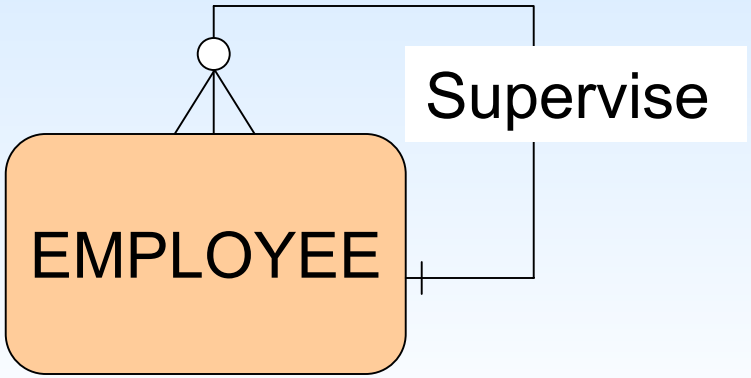
Entity Relationship Diagrams…

Degree

Number of entities that participate in the relationship

Degree =1

Recursive Relationship – Relationship that exists between different instances of the same entity.



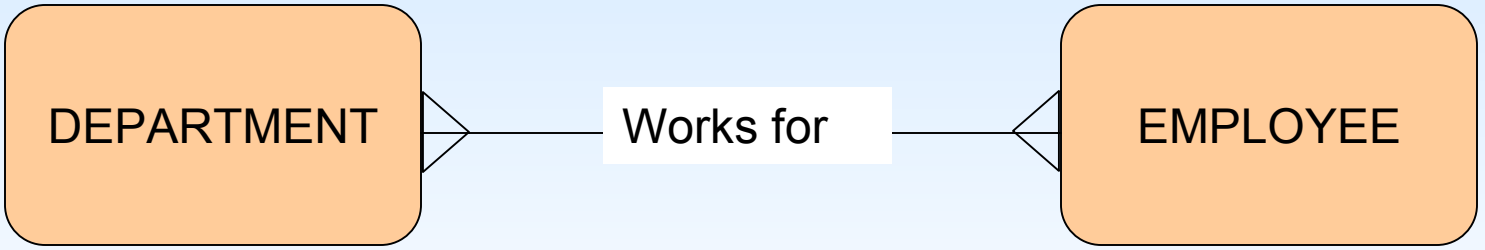
Data Modeling…

Entity Relationship Diagrams…

Degree…

Degree =2

Binary Relationship - When two different entities participates in a relationship



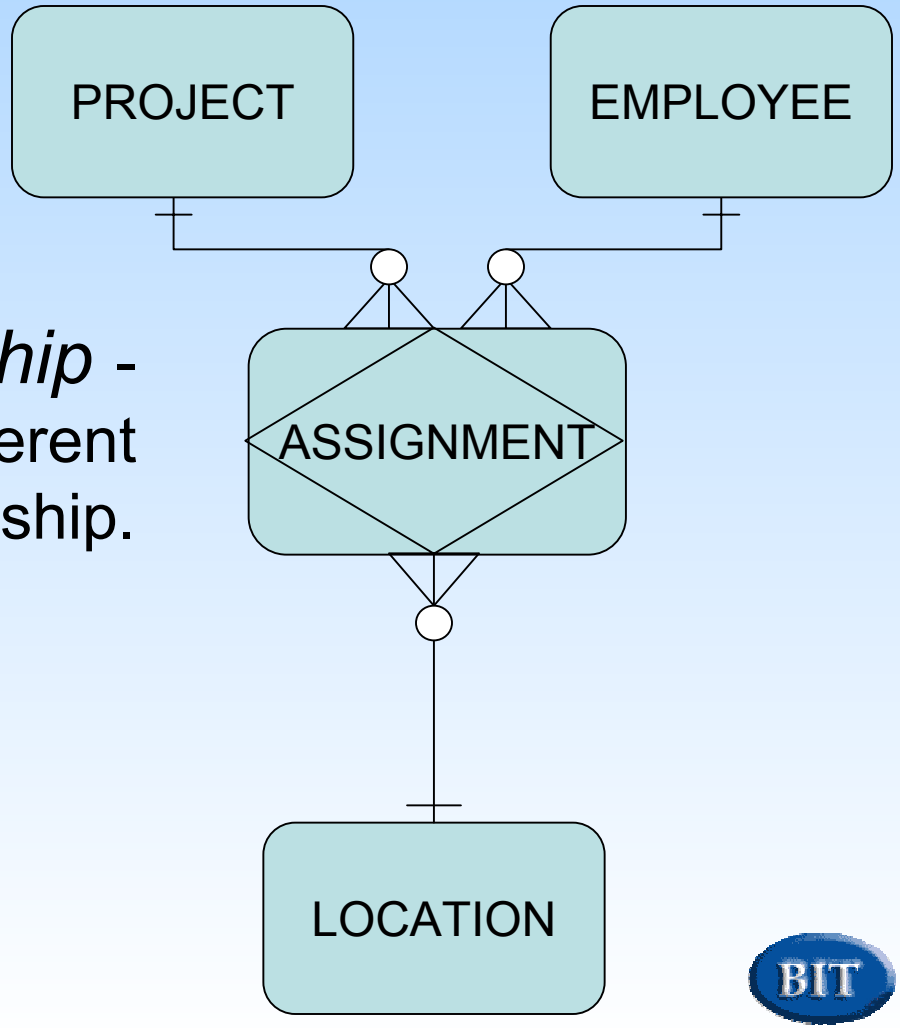
Data Modeling…

Entity Relationship Diagrams…

Degree…

Degree =3

Ternary or 3-ary Relationship - When more than two different entities participates in a relationship.



Object Modeling

Introduction

The object oriented approach differs from structured systems analysis and design.

Thus, rather than modeling systems by data flows, E-R diagrams or process descriptions, all three components are integrated together into objects.

Object Modeling

Introduction…

In Object Analysis you do not need to think in terms of building one large system.

Instead we identify objects as independent entities having their own local goals.

Such objects exchange messages between themselves to achieve a global goal of the large system.

Object Modeling

Introduction…

What is an Object?

Objects are the elements through which we understand the world around us.

Objects have

recognizable identities or properties (what it knows)

and

particular behaviors or methods (What an object does)

These identities and behaviors enables us to recognize them as discrete things

Object Modeling

Human is an object

Behavior

- Smile
- Run
- Walk
- Etc

Properties

- Skin tone
- Height
- Weight
- Etc

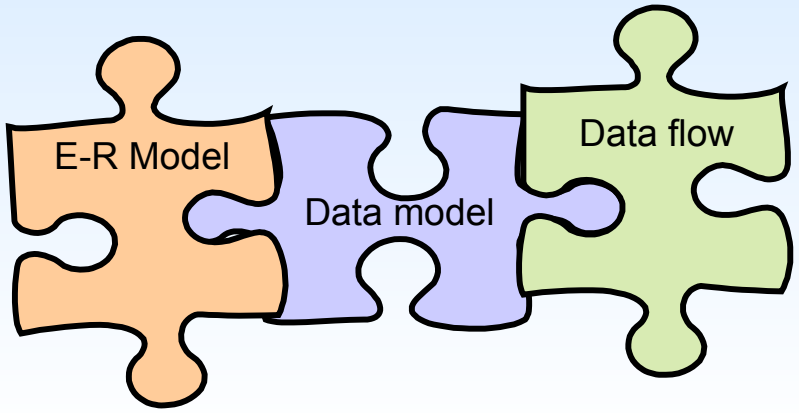
Ball, Stick are other objects

Object Modeling…

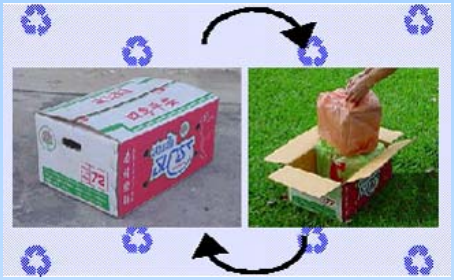
Introduction

The object oriented approach differs from structured systems analysis and design.

Thus, rather than modeling systems by data flows, E-R diagrams or process descriptions, all three components are integrated together into objects.



Object Modeling…



Reuse

Object-oriented analysis (OOA)

OOA is a technique used to:

- Reuse existing components
- Define new or modified objects that will be combined with existing objects

New

Modify

In order to produce useful business computing application.

UCSC

BIT

Object Modeling…

Object modeling (OM)

OM is a technique for identifying objects within the systems environment and the relationships between those objects.

Class room

Identified objects in class room

Teacher

Book

Chair

Student

Etc…

Object Modeling…

Class

A **class** is a set of objects that share common attributes and behavior. A class is sometimes referred to as an *object class*.

Student
Name Age Grade
Smile ( ) Run ( ) Study ( )

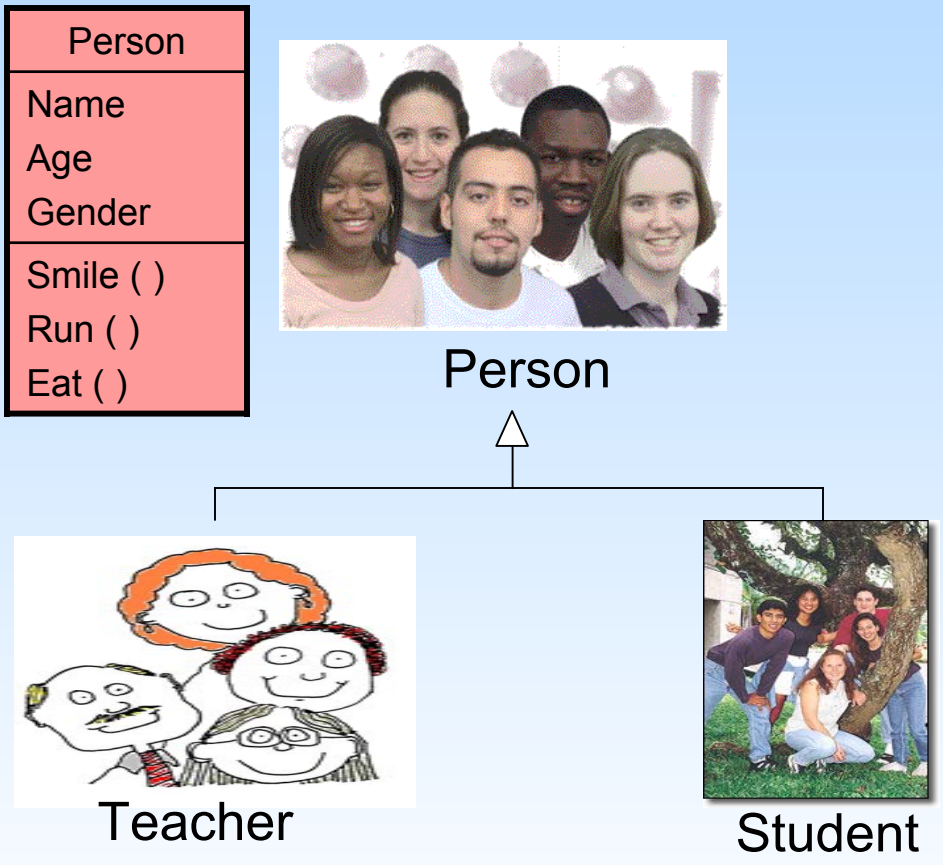
Student Class

Book
ISBN author type title copyright
Open() Close()

Book Class

Object Modeling…

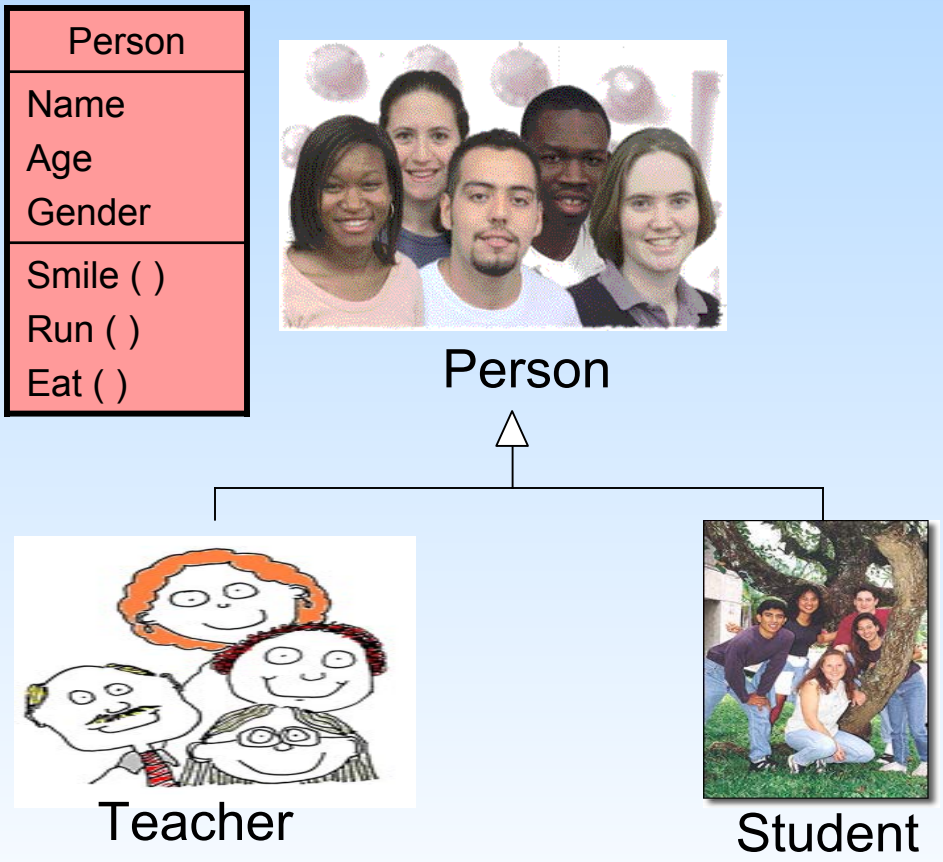
Generalization/Specialization



Generalization/specialization is a technique wherein the attributes and behaviors that are common to several types of object classes are grouped into their own class, called a *supertype*. The attributes and methods of the supertype object class are then inherited by those object classes.

Object Modeling…

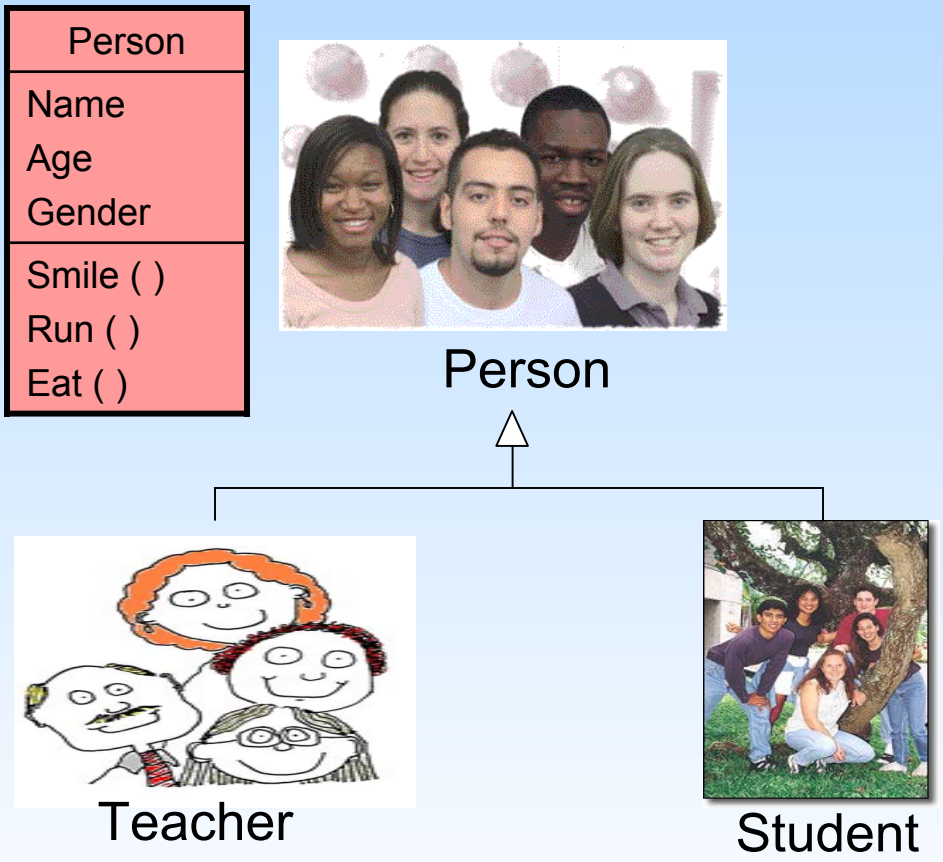
Generalization/Specialization…



A class **supertype** is an object class whose instances store attributes that are common to one or more class subtypes of the object.

Object Modeling…

Generalization/Specialization…

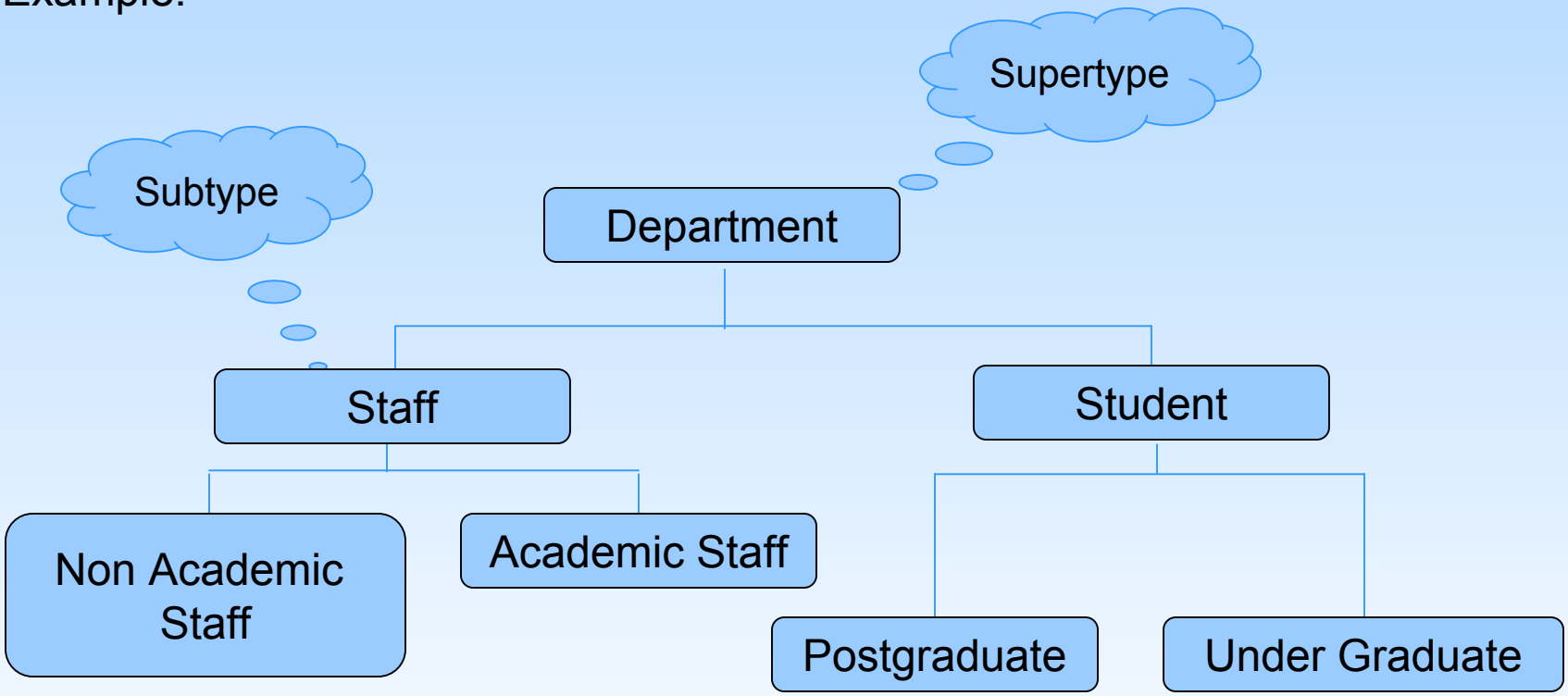


A class **subtype** is an object class whose instances inherit some common attributes from a class supertype, and then add other attributes that are unique to an instance of the subtype.

Object Modeling…

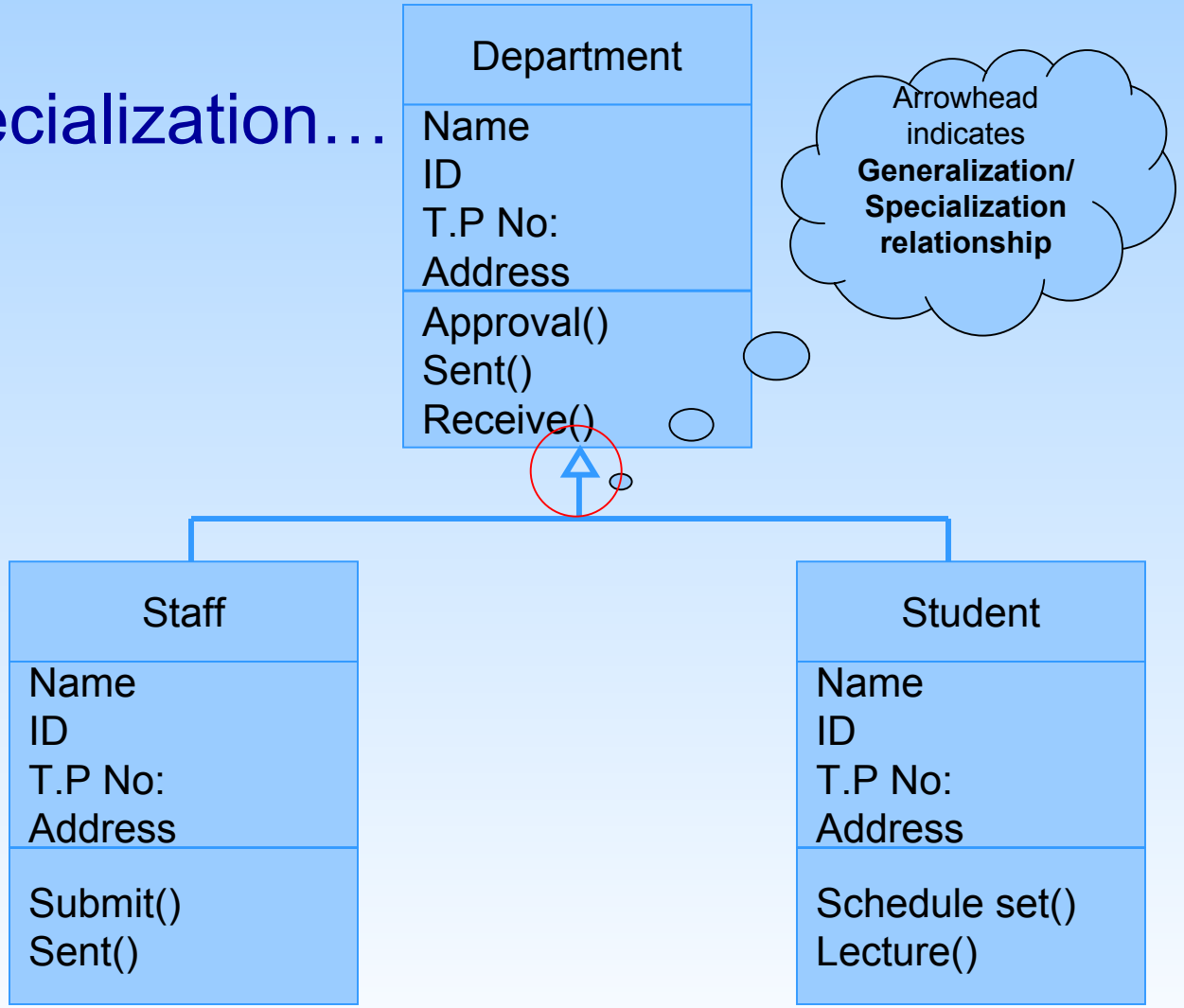
Generalization/Specialization…

Example:



Object Modeling…

Generalization/Specialization…



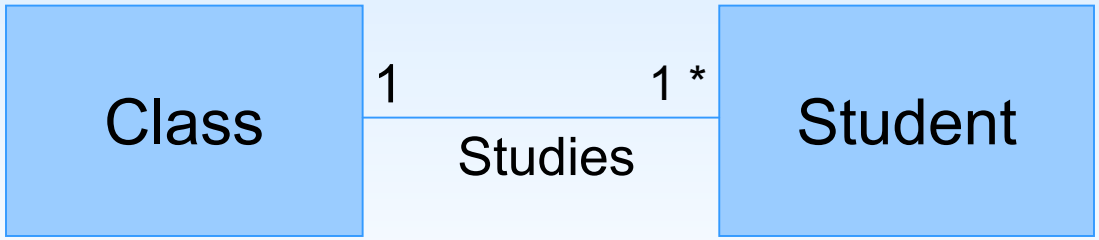
UML Representation of Generalization/Specialization

Object Modeling…

Object/Class relationship

is a natural business association that exists between one or more objects/classes.

- Although objects can be developed independently of other objects, they also have a relationship to other objects.
- One such relationship is to use a link directly from one object to another : **Association**

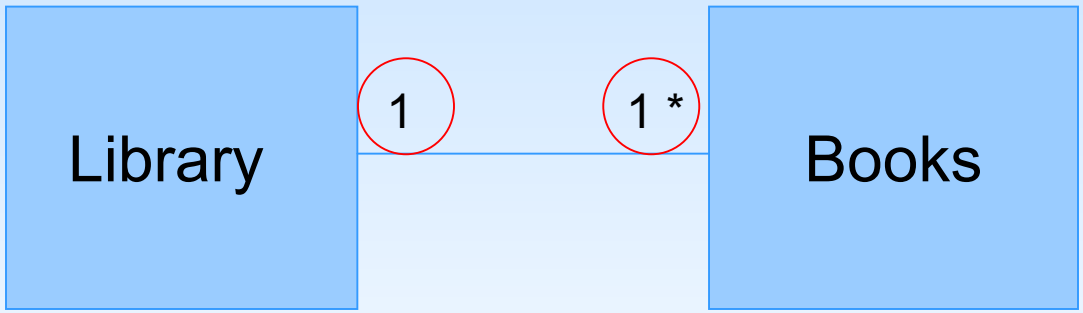


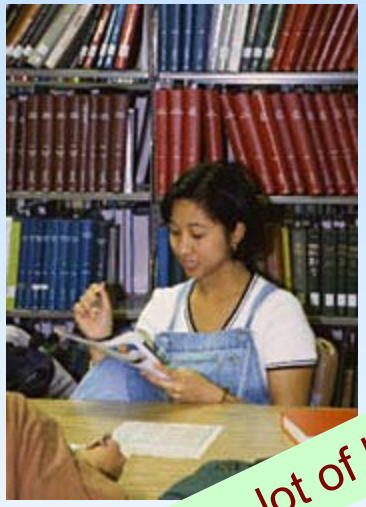
Example: There are many student in the each class

Object Modeling…

Multiplicity

Multiplicity defines how many instances of one object/class can be associated with one instance of another object/class.



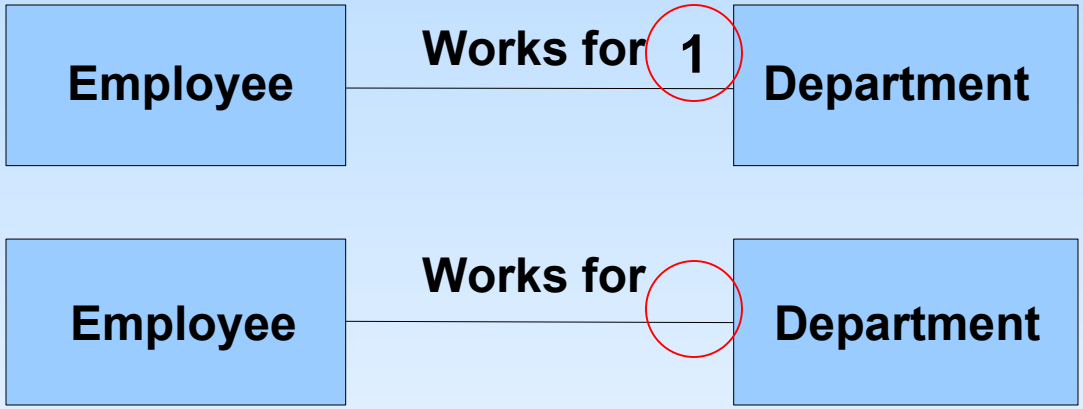


All the library having a lot of books

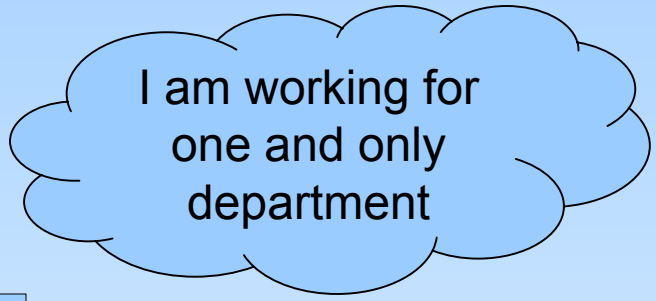
Object Modeling…

Multiplicity…

Exactly one - 1



1 or leave blank



Object Modeling…

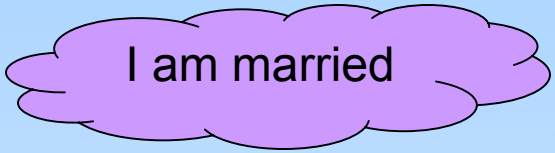
Multiplicity…

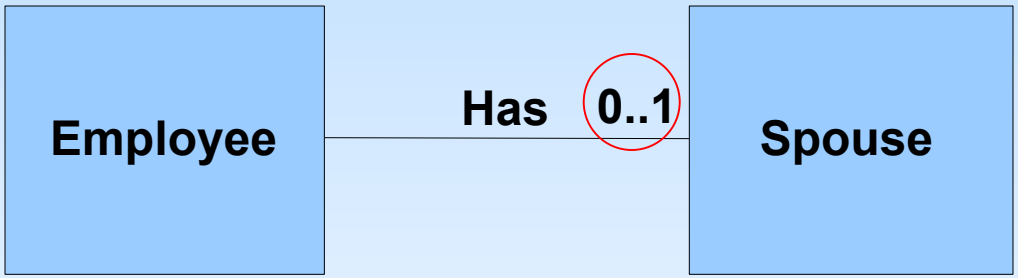
Zero or more – 0..*

Object Modeling…

Multiplicity…

Zero or one – 0..1





UCSC

BIT

Object Modeling…

Multiplicity…

One or more – 1..*

We offer one or more courses

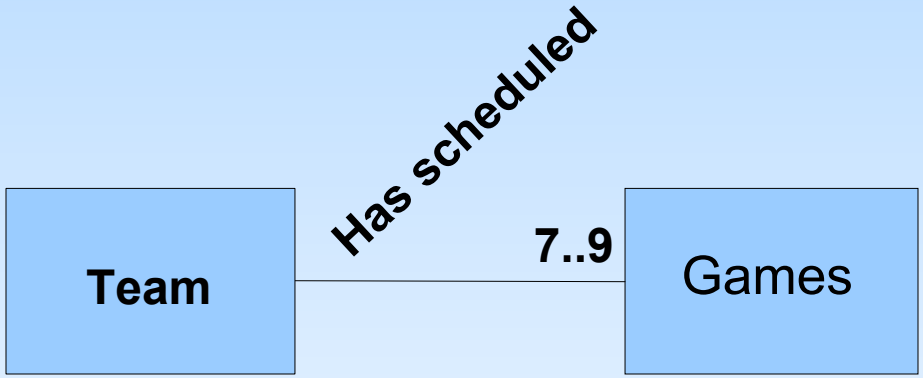
UCSC

BIT

Object Modeling…

Multiplicity…

Specific range – 7..9



In year 2004 BRAZIL have to play 7, 8, or 9 games

UCSC

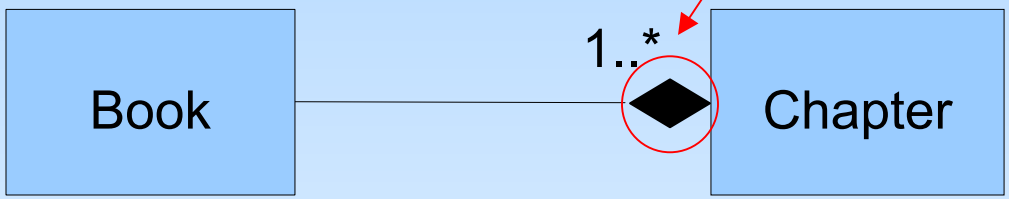
BIT

Object Modeling…

Aggregation Relationship

Black filled diamond

Composition Aggregation Relationship



◆ - Indicates Composition Aggregation Relationship

There are no books without chapter

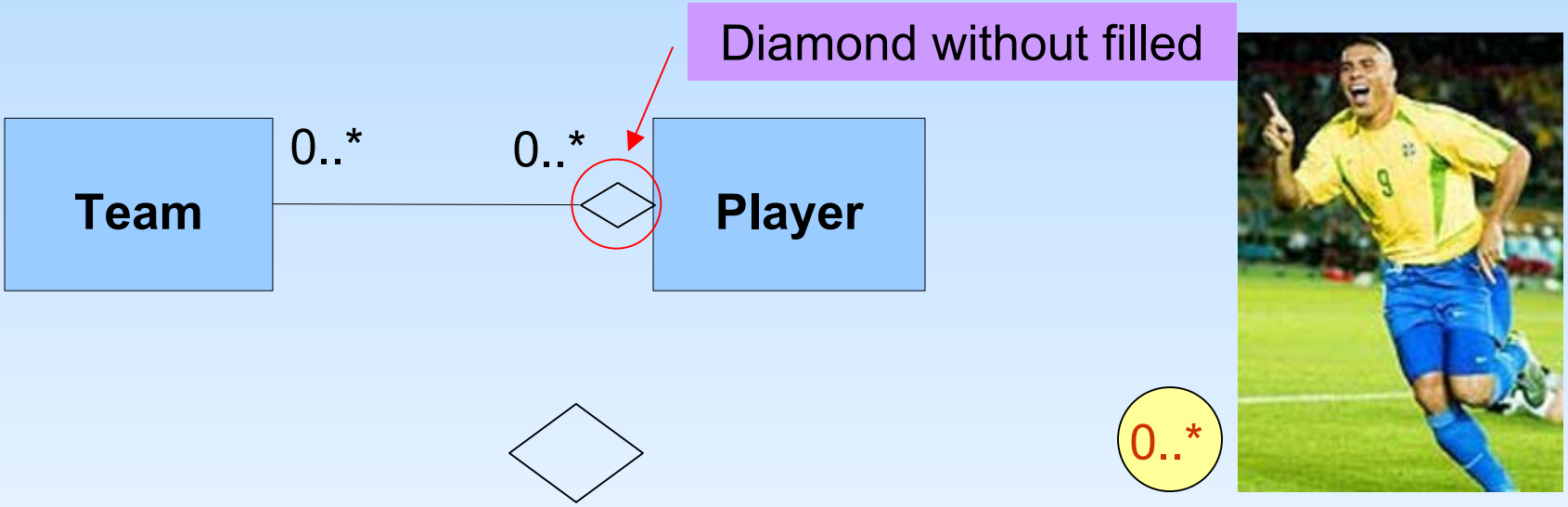
UCSC

BIT

Object Modeling…

Aggregation Relationship…

Shared Aggregation Relationship



Team may have Ronaldo or not

Indicates shared Aggregation Relationship

Object Modeling…

Message

A **Message** is passed when one object invokes one or more of another object's methods to request information or some action.

Catch the cat

The catch message call catch() method of dog then the catch method automatically call the run method.

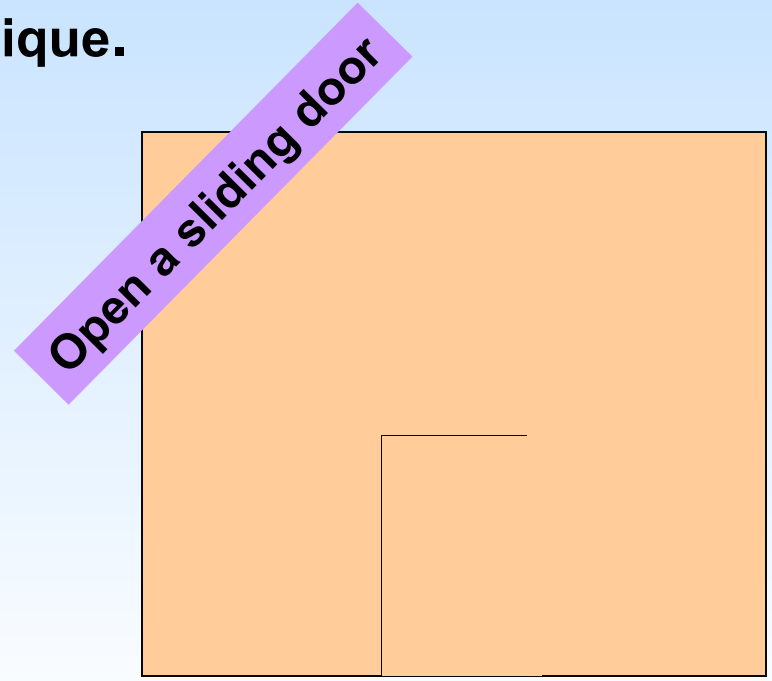
Dog can run & catch

Object Modeling…

Polymorphism

It mean that the same named behavior may be completed differently for different objects.

Polymorphism means "many forms" applied to object oriented technique.



Object Modeling…

Unified Modeling Language (UML)

Only a notation that is now widely accepted as a standard for object modeling.

UML – Diagrams

Gives five different groups of diagrams to model a system. These deferent diagrams provides the development team a deferent perspective of the information system.

Example: Well known Housing industry where they use lot of diagrams.

FLOOR PLAN

ARCHITECTURAL PLAN

ELEVATION

UCSC

BIT

Object Modeling…

UML – Diagrams…

Use Case Diagrams

Graphically describe who will use the systems and in what ways the user expects to interact with the system.

Object Modeling…

UML – Diagrams…

Class Diagrams

Shows object classes that the system is composed of as well as the relationship between those objects.

Object Modeling…

UML – Diagrams…

Object Diagrams

Similar to class diagrams, but instead of depicting object classes, they model actual object instances.

Provide snap shot of the system's objects at one point in time

Object Modeling…

UML – Diagrams…

Sequence Diagram

Graphically depict how object interact with each other via messages in the execution of a use case or operation.

Object Modeling…

UML – Diagrams…

Collaboration Diagrams

Similar to sequence diagrams but do not focus on the timing or sequence of messages. Instead they represent the interaction between the objects in a network format.

Object Modeling…

UML – Diagrams…

State Diagrams

Use to model the dynamic behavior of particular model. They illustrate an objects life cycle. During the life cycle the object can change to various states.

Object Modeling…

UML – Diagrams…

Activity Diagrams

Graphically depict the sequential flow of activities of either a business process or use case.

Object Modeling…

UML – Diagrams…

Components Diagrams

Used to graphically depict the physical architecture of the system.

Object Modeling…

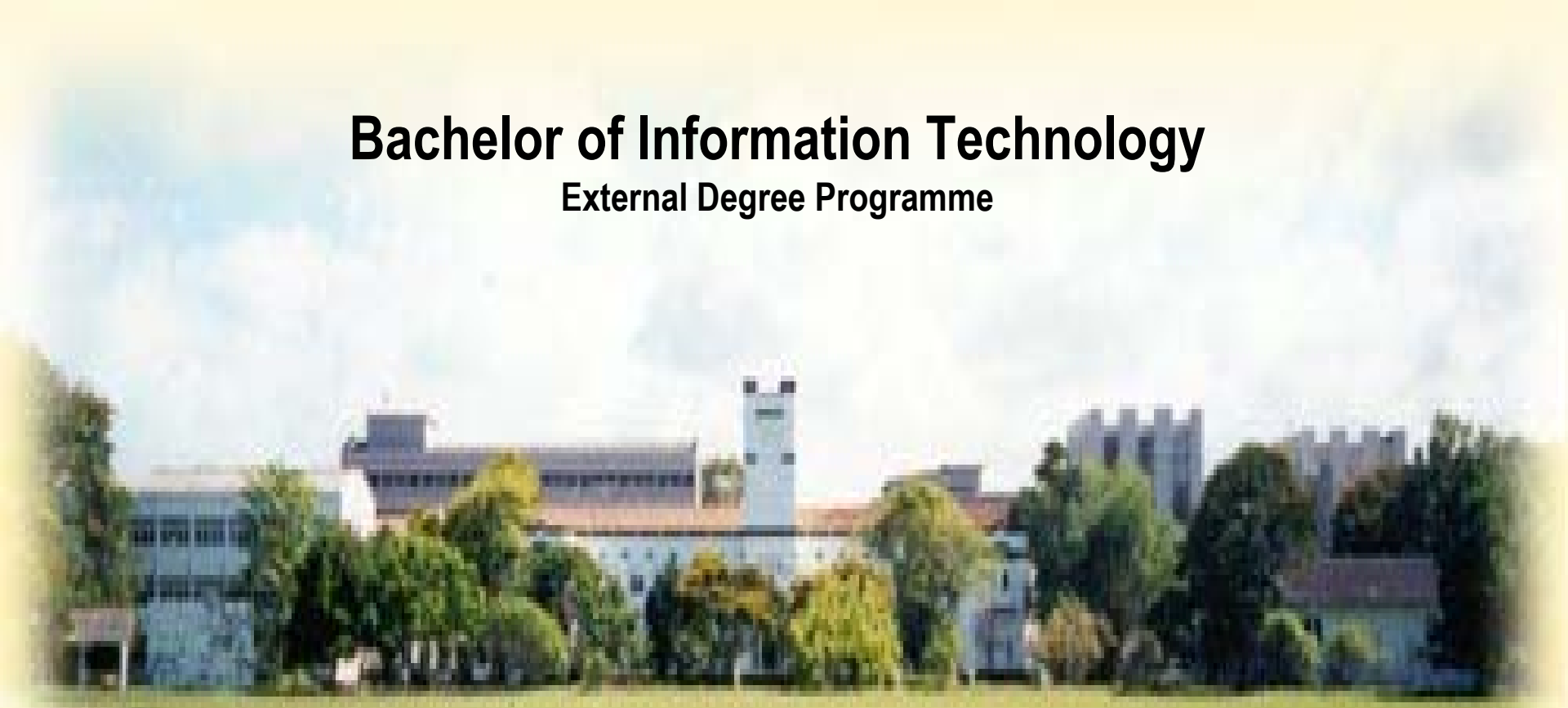
UML – Diagrams…

Deployment Diagrams

Describe the physical architecture of the hardware and software in the system.

Bachelor of Information Technology

External Degree Programme



Systems Analysis and Design

Ref_1 : System Analysis and Design Methods **By** Whitten, Bentley, Dittman ISBN 0-07-044539-7 (5th Edition)
Ref_2 : System Analysis and Design **By** Igor Hawryszkiewycz (4th Edition) ISBN 81-203-1670-3
Ref_3 : http://www.mhhe.com/whitten
Ref_4 : Practical SSADM Version 4 A Complete Tutorial Guide **By** Philip L Weaver ISBN 0-273-60095-8

UCSC

University of Colombo School of Computing

BIT

System Design

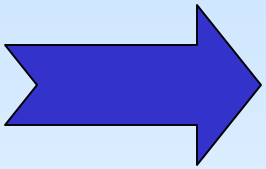
UCSC

BIT

System Design

- Produces a design specification for the new system

Analysis

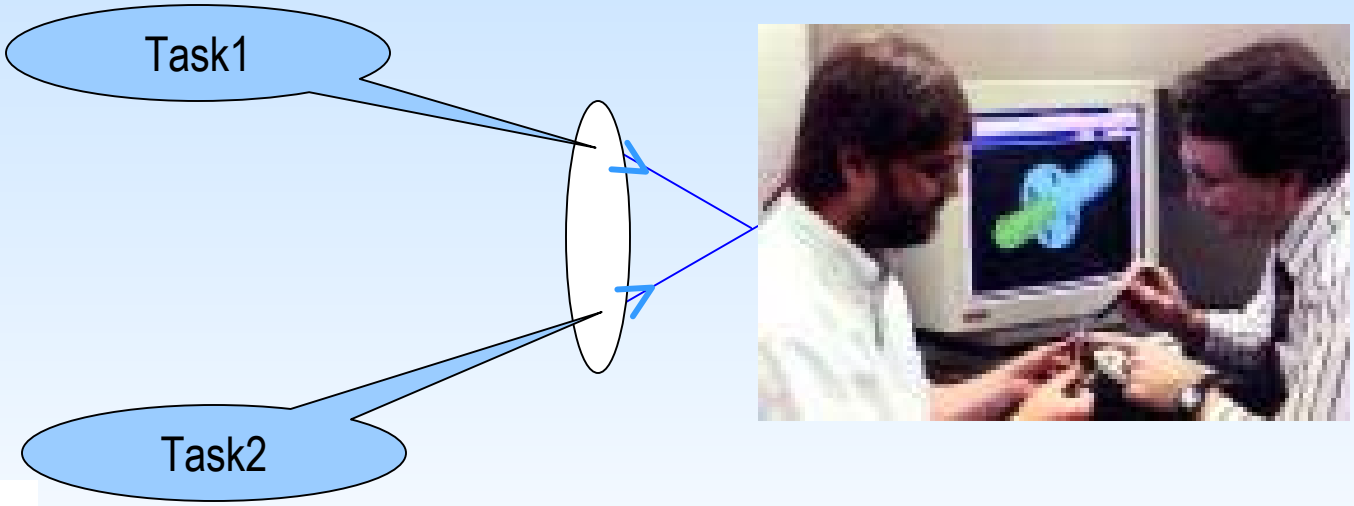


Design

System Design

Information **systems design** is defined as those tasks that focus on the specification of a detailed computer-based solution. It is also called **physical design**.

Thus, whereas systems analysis placed emphasis on the business problem, systems design places emphasis on the technical or implementation concerns of the system.



UCSC

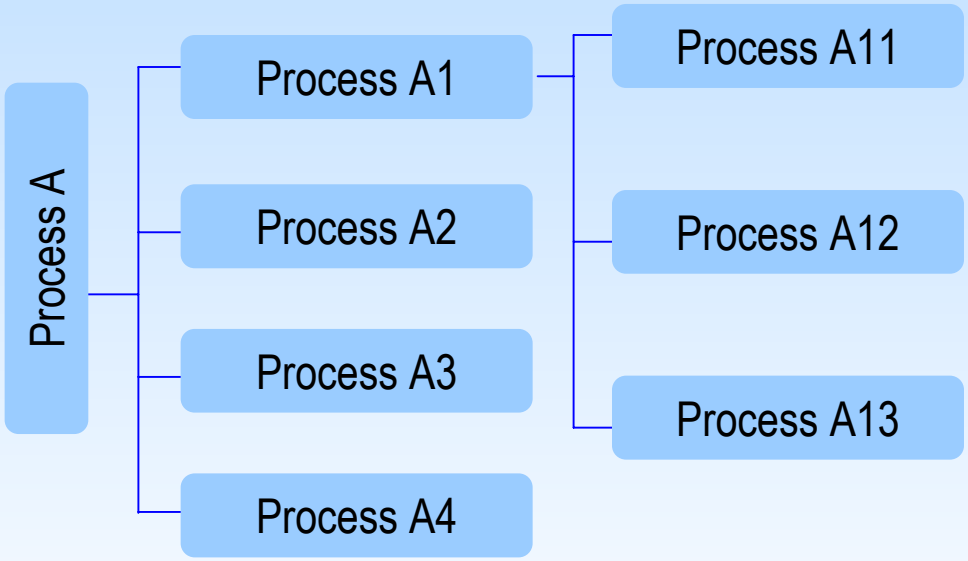
BIT

Model-Driven

  - Modern structured design
  - Information engineering
  - Prototyping
  - Object-oriented
- JAD
- RAD

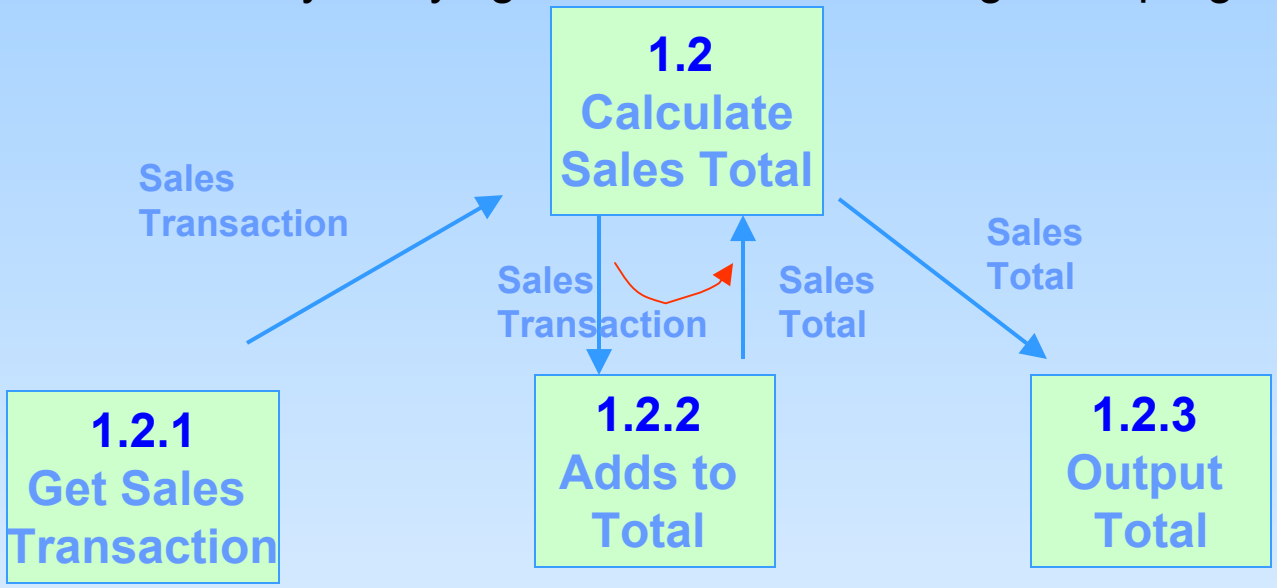
Modern Structured Design

Modern Structured Design is a process-oriented technique for breaking up a large program into a hierarchy of modules that result in a computer program that is easier to implement and maintain (change). Synonyms (although technically inaccurate) are top-down program design and structured programming.



Structure Chart Design

The software model derived from structure design is called a **structure chart.** It is derived by studying the flow of data through the program.



Top Module calls three lower level program modules to accomplish its task.
It calls *Get Sales Transaction* module to read individual sales transactions.
It then calls module *Add to Total* to sum the amount in each transaction.
Finally it calls module *Output Total* to output the sum

UCSC

BIT

- ***Parameter Passing***

The calling module passes a set of values to the called module and receives a set of values in return

These values are passed as parameter values

eg. A value of '*sales transaction'* is passed from module *Get Sales transaction* to module *Calculate Sales Total*

Module *Calculate Sales Total* then passes the value of '*sales transaction*' to module *Add to Total* and get a value of '*sales total'* in return

The value of '*sales total'* is then passed from module *Calculate Sales Total* to module *output total*

- ***Execution Sequence***

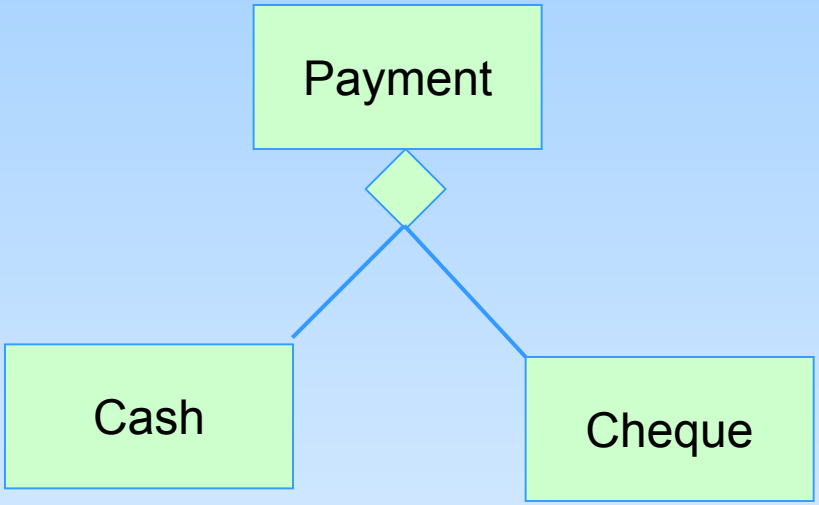
By convention, modules are executed from left to right in each level.

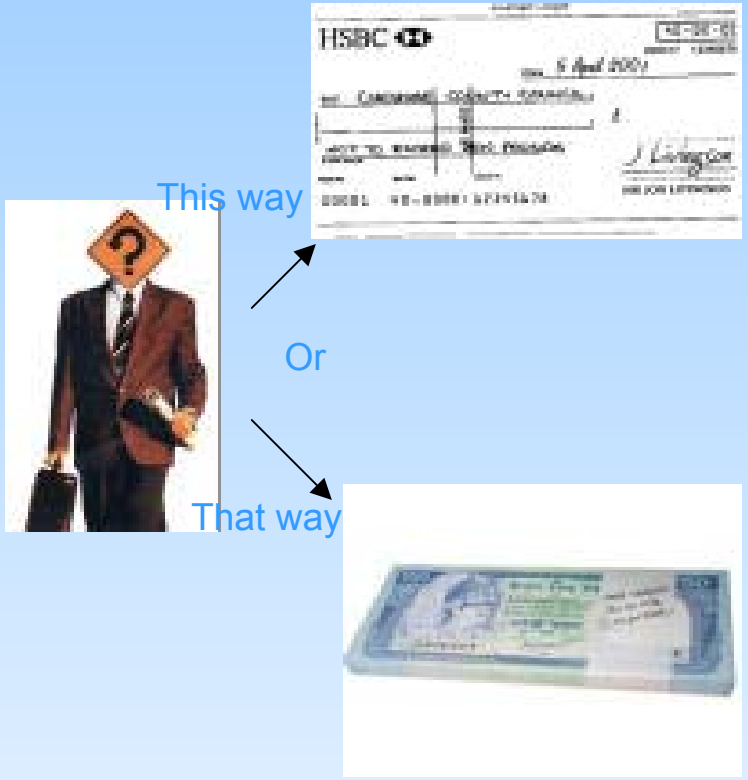
#Thus in the given example, module *Get Sales Transaction* is called before module *Add-To-Total*. Module *Output Total* is the last module to be called.

Certain conventions are also used to represent decisions and repetition.

Decisions occur whenever a calling module has to decide to call only one of a number of modules.

Decisions are modeled by a diamond symbol.





@ Payment pays either cash or Cheque

Repetition occurs when some modules are called repetitively by the calling module.

Repetition is modeled by a looping arrow

Structure Design

Structure Charts are developed by a process called ***Structured Design***
The objective of structured design is to produce a good design.

Good Design

be *understandable*

be *cohesive* -All parts of the component should have a close logical relationship each module should accomplish one and only one function. this makes the modules reusable in future programs.

adaptable - Design should be well documented. It should be easy to incorporate changes made to the design in all design documents.

be *loosely coupled* -Coupling is a measure of independence of components. Looser the coupling easier it is to adapt the design. Modules should be minimally dependent on one another. This minimizes the effect the future change in one module will have on another modules

Prototyping

Prototyping:

The prototyping approach is an iterative process involving a close working relationship between the designer and the users.

UCSC

BIT

Prototyping..

Key Benefits:

Prototyping encourages and requires active end-user participation.

Iteration and change are a natural consequence of systems development thus, it accommodates end-users whom tend to change their minds.

Prototyping endorses the philosophy that end-users wont know what they want until they see it.

Prototypes are an active, not passive, model that end-users can see, touch, feel, and experience.

An approved prototype is a working equivalent to a paper design specification, with one exception -- errors can be detected much earlier.

Prototyping can increase creativity because it allows for quicker user feedback, which can lead to better solutions.

Prototyping accelerates several phases of the life cycle, possibly bypassing the programmer.

UCSC

BIT

Rapid Application Development (RAD)

RAD is the merger of various structured techniques (especially the data-driven information engineering) with *prototyping* techniques *and joint application development* techniques to accelerate systems development.

After 40 days

UCSC

BIT

Joint Application Development (JAD)

JAD is a technique that complements other systems analysis and design techniques by emphasizing *participative development* among system owners, users, designers, and builders.

During the JAD sessions for systems design, the systems designer will take on the role of facilitator for possibly several full-day workshops intended to address different design issues and deliverables.

System Design

Design Database

- The design of the data goes far beyond the simple layout records. Database are shared resource. To designing database that are adaptable to future requirements and expansion.
  - Designer must also design internal controls to ensure proper security and disaster recovery technique, in case data are lost or destroyed.
  - During the database design include record size and storage volume requirement.
  - Database has been designed and possibly a prototype built.
  - Designer can work closely with system user to develop input and output specifications.

UCSC

BIT

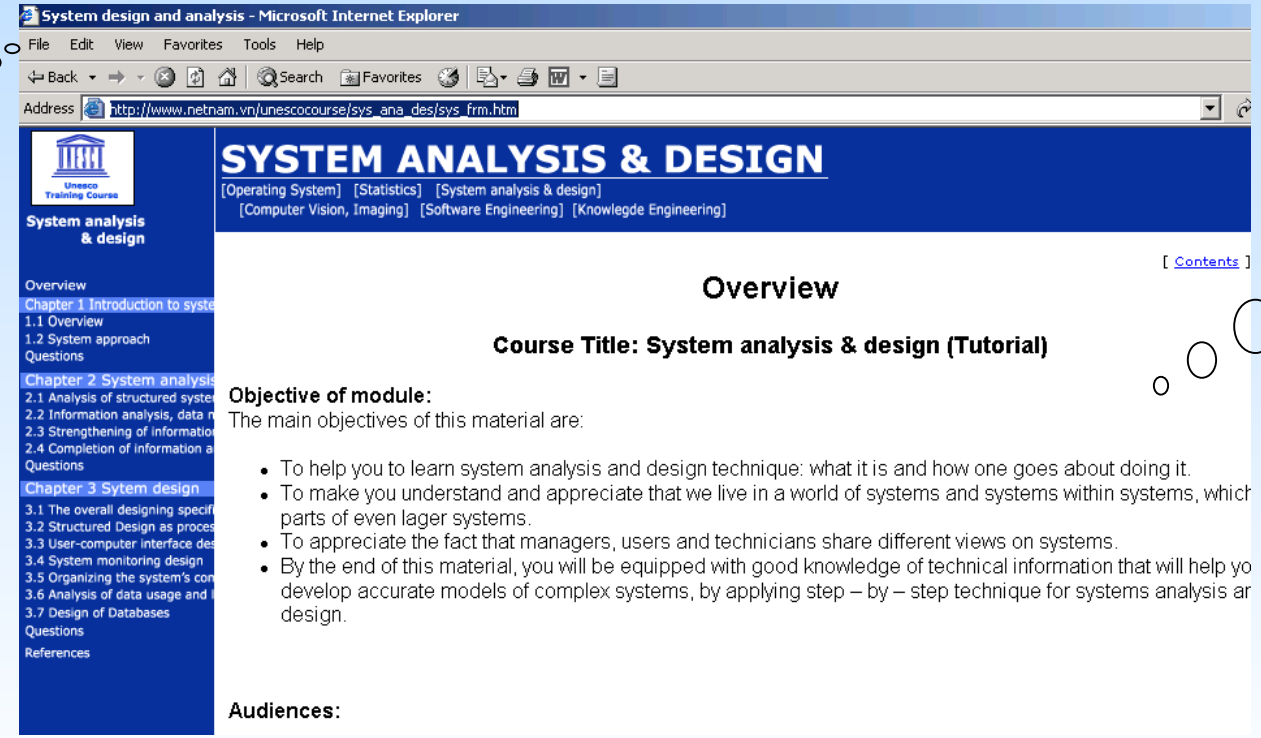
Design Interface

- This activity is omitted many designers. However for on line system the development of interface design specification from interface design requirement may be the most critical design activity.
  - To build an easy to learn and easy to use dialogue for the user's new system.

  This dialogue must consider factors such as

  # Terminal familiarity
  # Possible error and misunderstandings that the end user may have
  # Need to foe additional or help at certain points
  # Screen content and layout.

Terminal familiarity



Screen content and layout

UCSC

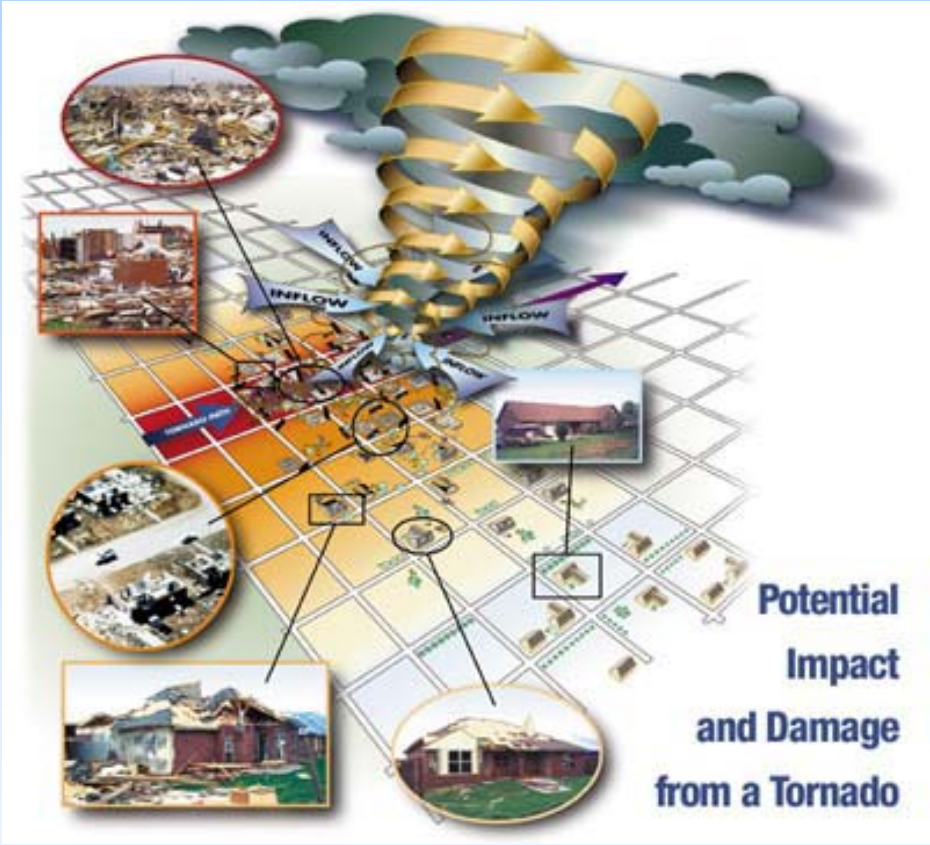
BIT

Application Architecture and Modeling

Application Architecture

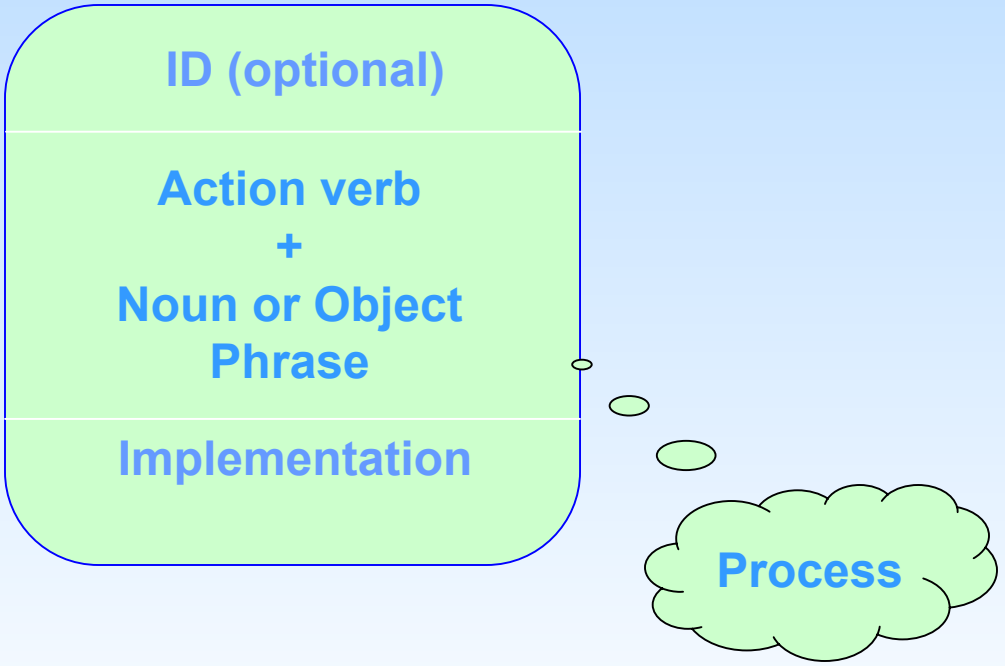
✓ An application architecture defines the technologies to be used by one or more (possibly all) information system in terms of its *data, process, interface* and how these components interact across a network.

✓ It serves as an outline or blueprint for detailed design and implementation.



Data Flow Diagrams (DFDs)

Physical data flow diagrams (DFDs) model the technical and human decisions to be implemented as part of an information system. They communicate technical choices and other design decisions to those who will actually construct and implement the system.



Physical Process

A **physical process** is either a *processor*, such as a computer or person, or a technical implementation of specific work to be performed, such as a computer program or manual process.

Logical processes may be assigned to physical processors such as PCs, servers, mainframes, people, or devices in a network. A physical DFD would model that network structure.

Each logical process requires an implementation as one or more physical processes. Note that a logical process may be split into multiple physical processes:

To define those aspects that are performed by people or computers.

To define those aspects to be implemented by different technologies.

To show multiple implementations of the same process.

To add processes for exceptions and internal control (e.g., security).

Name
acct.No
date
amount

Customer

Withdrawal Details

Approval slips

1

Check Customer Credit

Acct. Clerk

Money

UCSC

BIT

Sample Physical Data Flow Diagram

Item ID
Quty.

Items
Order

a
Customer

1

Supply

Sales manager

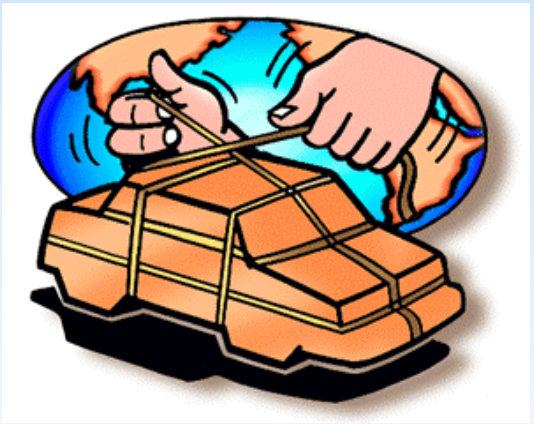
Process

Item ID
Quty.

Packing
Order

Entity

b
Ware House



Physical Data Stores

It represents the implementation of one of the following

database

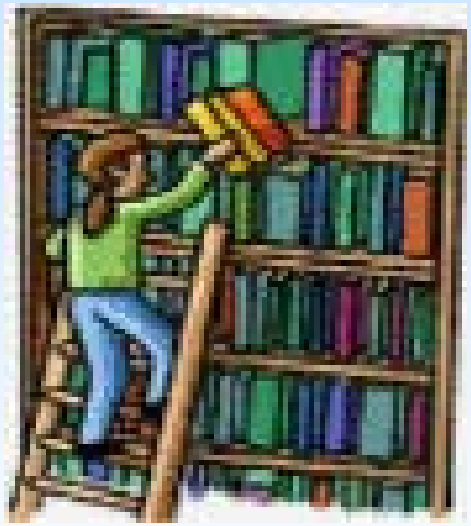
a table in a database

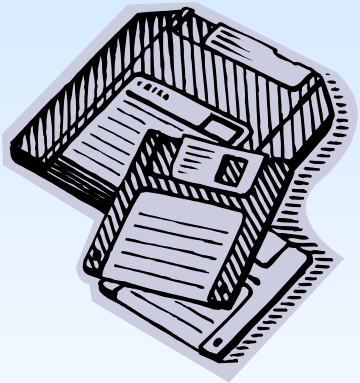
a computer file

a type or media backup of anything important

any temporary file or batches as needed by a programs

any type of non computerized file





Data Store Notation

ID (opt)	Implementation method

UCSC

BIT

Information Technology Architecture

It will attempt to summarize contemporary IT alternatives and trends that are influencing design decision as we go to press. IT architecture should be noted that new alternative are continuously evolving

Distributed System:

A **distributed system** is one in which the DATA, PROCESS, and INTERFACE components of an information system are distributed to multiple locations in a computer network. Accordingly, the processing workload is distributed across the network.

Centralized Systems:

In **centralized systems**, a central, multi-user computer hosts all the DATA, PROCESS, and INTERFACE components of an information system. Users interact with the system via terminals (or terminal emulators).

Why Distributed Systems?

Modern business systems are already decentralized and distributed.
Distributed computing moves information and services closer to the customers and users who need them.
Distributed computing consolidates the power of personal computers across the enterprise.
Distributed computing solutions are more user-friendly because they utilize the PC as the end user interface.
Personal computers and network servers are cheaper than centralized mainframe computers.

Distributed computing layers

Presentation layer—the user interface

Presentation layer logic—such as input editing

Application logic layer—the business rules, policies, and procedures

Data manipulation layer—to store and retrieve data to and from the database

Data layer—the actual business data

UCSC

BIT

Distributed information system architecture are three flavors of distribution information systems

File server computing

Client/server computing

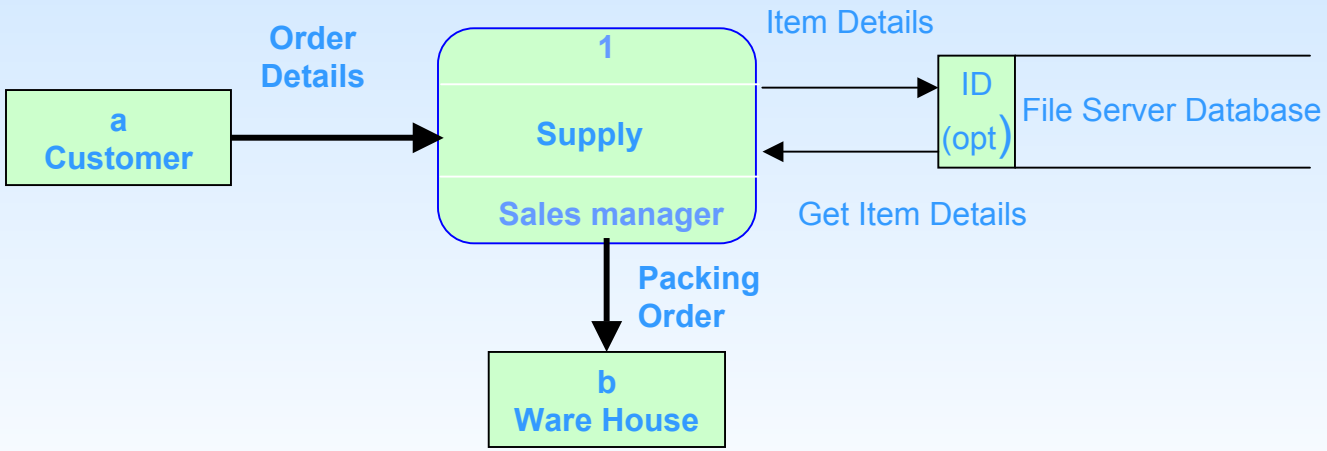
Internet based computing

File server computing

A **file server system** is a LAN-based solution in which a server hosts only the data layers of an information system. All other layers are implemented on the client computers.

Disadvantages include:

Frequently excessive network traffic to transport data between servers and clients.

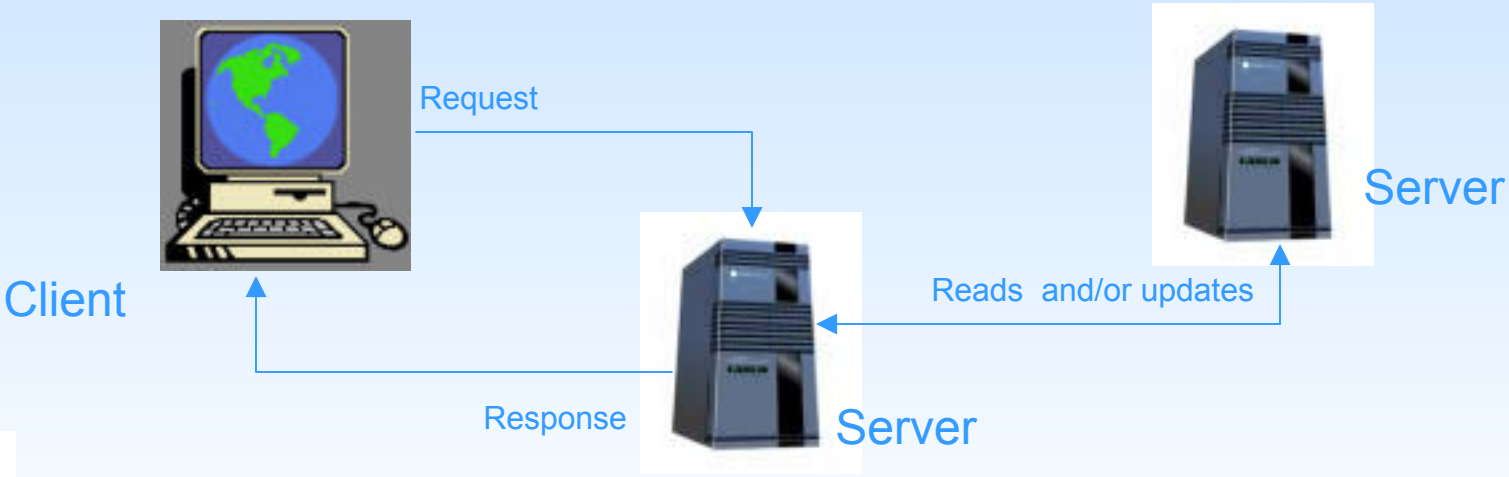


Client/Server Architecture -The Clients

A **client/server system** is a solution in which the presentation, presentation logic, application logic, data manipulation, and data layers are distributed between client PCs and one or more servers.

A **thin client** is a personal computer that does not have to be very powerful (or expensive) in terms of processor speed and memory because it only presents the user interface to the user.

A **fat client** is a personal computer or workstation that is typically more powerful (and expensive) in terms of processor speed, memory, and storage capacity. Most PCs are considered to be fat clients.



UCSC

BIT

Client/Server Architecture-The Servers

A **database server** hosts one or more shared databases but also executes all data manipulation commands. Most database servers host an SQL database engine such as Oracle, Microsoft SQL server, or IBM Universal Database.

A **transaction server** hosts services that ultimately ensure that all database updates for a single transaction succeed or fail as a whole.

An **application server** hosts the application or business logic and services for an information system. An application server is often integrated with the transaction server

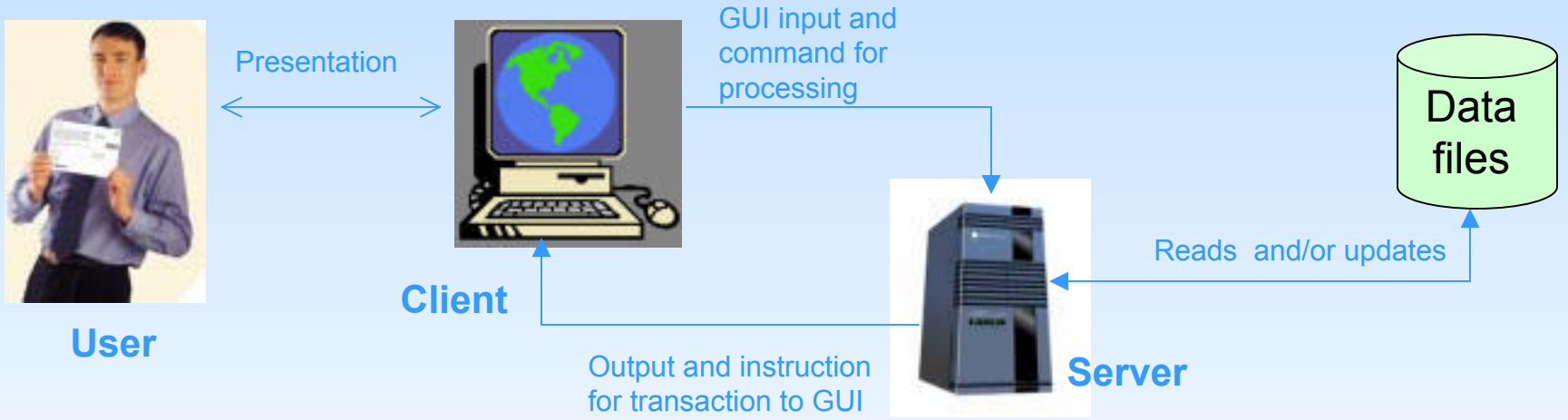
A **messaging or groupware server** hosts services for e-mail, calendaring, and other work group functionality.

A **web server** hosts Internet or intranet web sites and services, communicating through thin-client interfaces such as web browsers. Some web server are specifically designed to host e-commerce applications.

Client/Server—Distributed Presentation

•A **distributed presentation** client/server system is a solution in which the presentation and presentation logic layers only are shifted from the server to reside on the client.

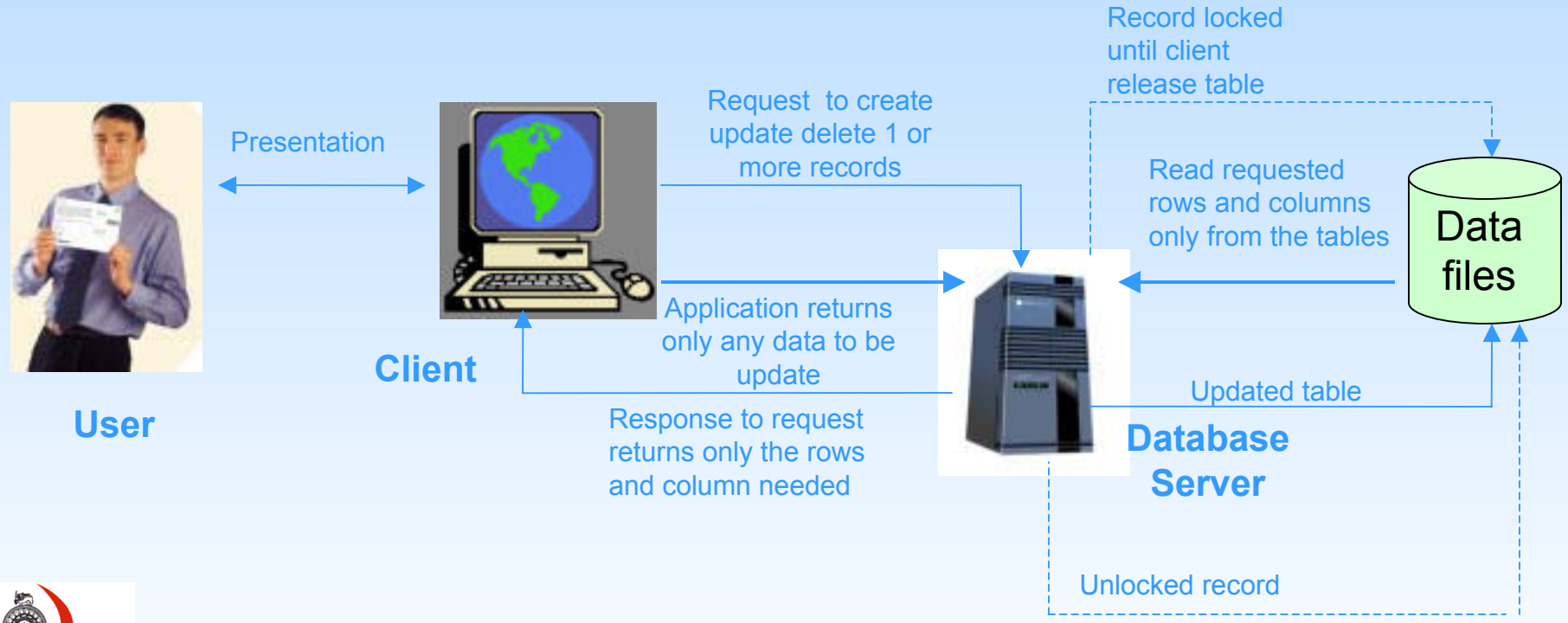
•The application logic, data manipulation, and data layers remain on the server (frequently a mainframe).



Client/Server—Distributed Data

•A **distributed data** client/server system is a solution in which the data and data manipulation layers are placed on the server(s), and the application logic, presentation logic, and presentation layers are placed on the clients.

•This is sometimes called two-tiered client/server computing.



Client/Server – Distributed Data and Application

•A **distributed data and application** client/server system is a solution in which: (1) the data and data manipulation layers are placed on their own server(s), (2) the application logic is placed on its own server, and (3) the presentation logic and presentation layers are placed on the clients.

•This is sometimes called **three- or n-tiered client/server computing**. It requires design partitioning.

•**Partitioning** is the art of determining how to best distribute or duplicate application components (DATA, PROCESS, and INTERFACE) across the network.

Client

Presentation

User

Data and service responses

Information and service responses

Record locked until client release table

Request to create update delete 1 or more records

Read requested rows and columns only from the tables

Data files

Application Server

Updated table

Response to data manipulation request

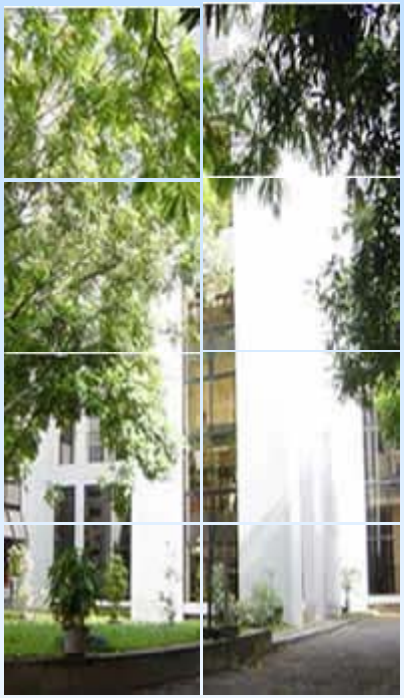
Database Server

Unlocked record

UCSC

BIT

Project Management and Documentation



UCSC

BIT

Project Management

- Demand for Project managers in the Information system community is strong
- Comes from experienced IS developers such as system analysts (SA)
- SA aware of Project management processes, tools and techniques
- Combined knowledge with development experience plus your own observation of Project Managers

UCSC

BIT

Project Management…

It is the process of

Staffing

Scoping

Planning

Organizing

Directing

Controlling

the development of an acceptable system

at a minimum cost

within a specified time frame

UCSC

BIT

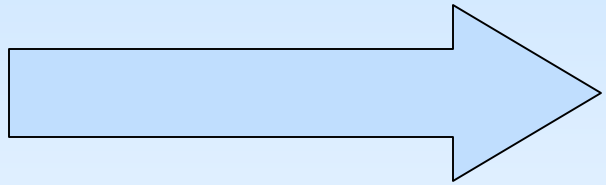
Project Management…

Project is successful if..

Delivered on time

Delivered within Budget

Resulting Information System



Customer

Acceptable to the customer

Minimal impact on ongoing business operation

UCSC

BIT

Project Management..

Causes of failed projects

- Failure to establish upper-management commitment to the project
- Lack of organization's commitment to the system development methodology
- Taking shortcuts through or around the system development methodology
- Poor expectations management
- Premature commitment to a fixed budget and schedule
- Poor estimating techniques
- Etc.

UCSC

BIT

Project Management…

Project Manager Competencies

- Some of these can be taught in courses, books and workshops
- Some come only with professional experience in the field

Influence Competencies

Problem-Solving Competencies

People Management Competencies

Business Achievement Competencies

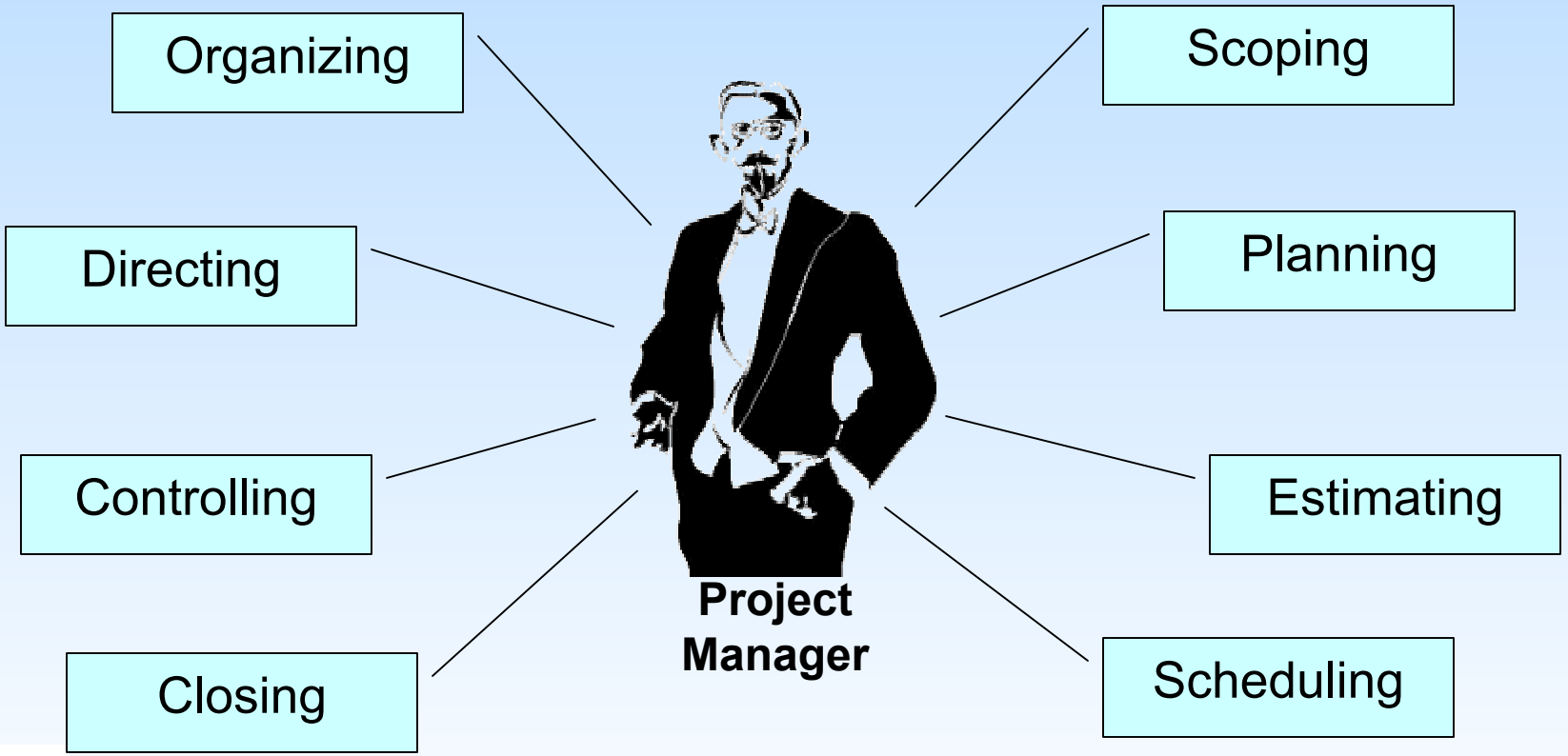
Self-Management Competencies

UCSC

BIT

Project Management..

Project Management Functions



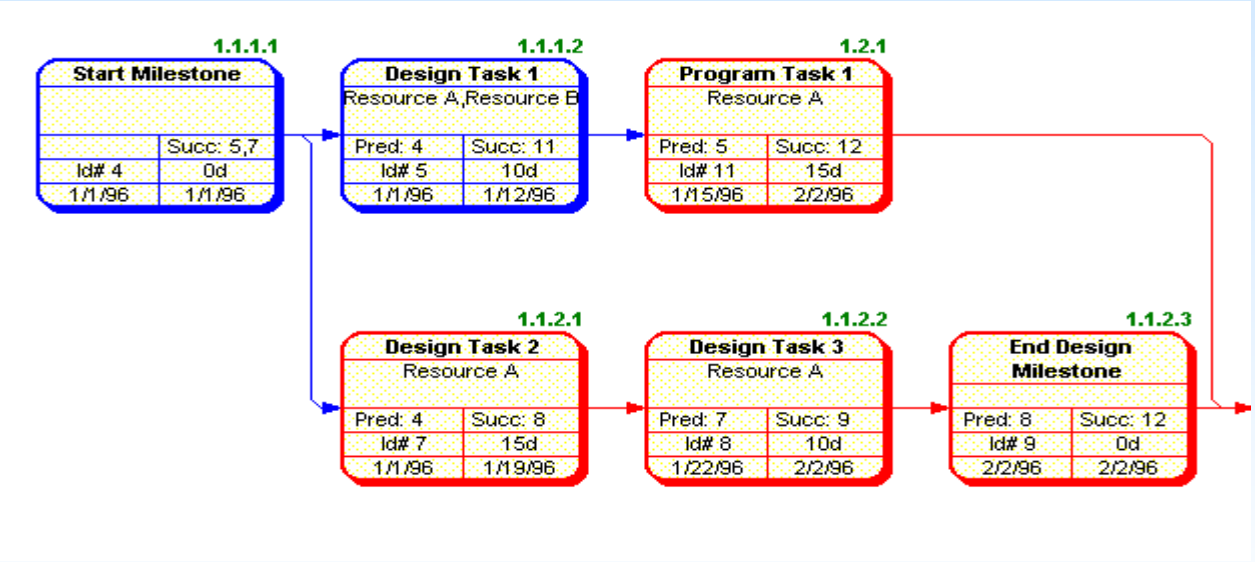
Project Management..

Project Management Tools and Techniques

- PERT chart
- Gantt chart

A PERT chart is a graphical network model that depicts a project's tasks and the relationships between those tasks.

Eg.



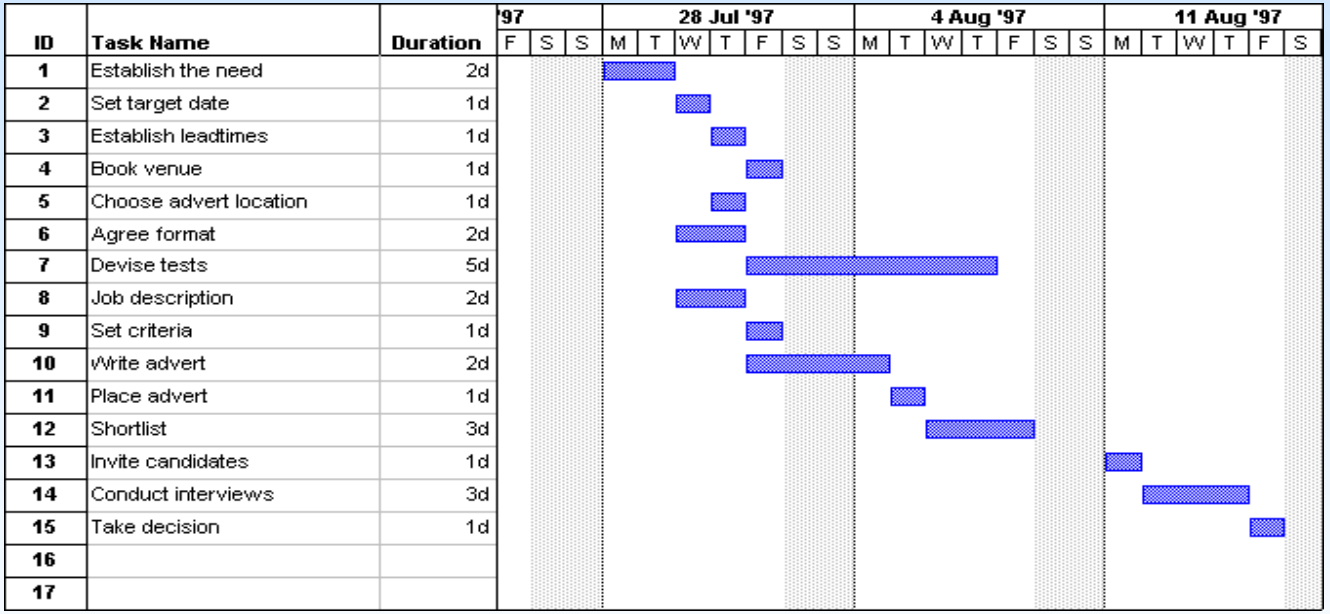
Project Management..

Project Management Tools and Techniques…

A Gantt chart is a simple horizontal bar chart that depicts project tasks against a calendar. Each bar represents a named project task.

Eg.

ID	Task Name	Duration
1	Establish the need	2d
2	Set target date	1d
3	Establish leadtimes	1d
4	Book venue	1d
5	Choose advert location	1d
6	Agree format	2d
7	Devise tests	5d
8	Job description	2d
9	Set criteria	1d
10	Write advert	2d
11	Place advert	1d
12	Shortlist	3d
13	Invite candidates	1d
14	Conduct interviews	3d
15	Take decision	1d
16		
17		



For more details : Ref1 p122 - 131

Documentation

Documentation is both a communication tool and a management tool.

- It is a communication tool :
  - because it contains a repository of all work done to date and makes it available to all persons working on related parts of a large project.
  - Such a repository can prevent unnecessary repetitions when someone leaves the project team.
  - Proper documentation ensures that all the information developed about the system is always available to new people joining the project.

UCSC

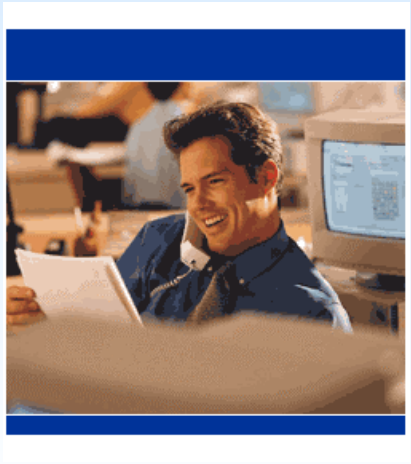
BIT

Documentation...

- Documentation is also a management tool.

- It supports management in two ways:
  - gives access to the latest work to all project personnel and thus reduces the chance of work having to be repeated.
  - is the only project deliverable, specially in the early project phases, and thus serves to determine project status and progress.

  Is also a part of the phase output.



UCSC

BIT

Documentation…

Different organizations use different terms to refer to the collection of all documents about the system.

- *System directory*
- *Project dictionary*
- *Document Configuration*
- *Configuration Management*

UCSC

BIT

Documentation…

Document Configuration

Configuration Management System:

- Store all documents created during the development process.
- Provides access to the latest versions of documents.

- Organizes documents in ways that allow work to be monitored throughout the development process, usually by linking documents to the development phases.

- Supports changes by maintaining document versions (variation of the same document) as well as their status.

Documentation…

Document Configuration..

Configuration

The Configuration contains the major documents for each phase. Including

- Statement of requirements – Concept Formation phase
- System specification, analysis and requirements models – system Specification phase
- Design models
- System modules

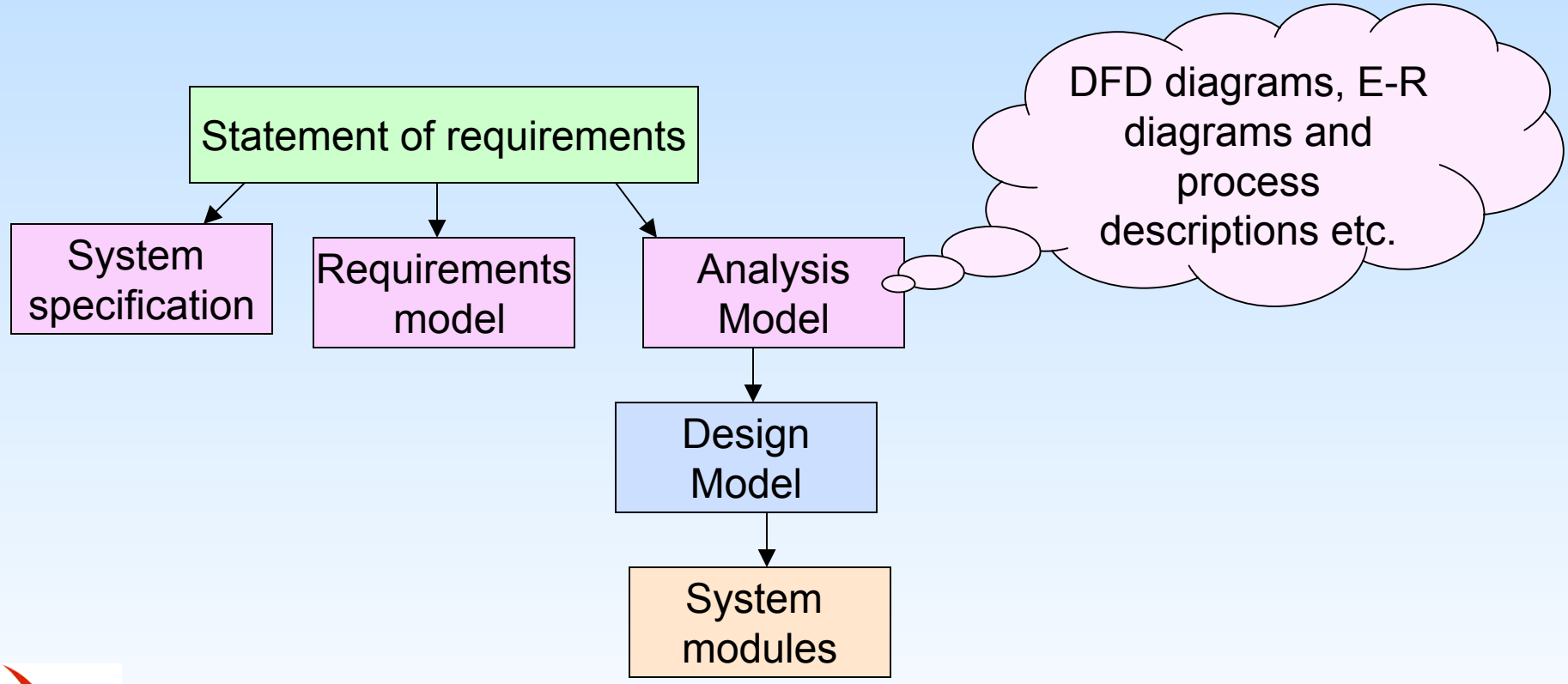
Documents stored in a configuration management system depend on the development process and the Modeling methods used

Documentation…

Document Configuration..

Configuration

The Configuration contains the major documents for each phase



Documentation…

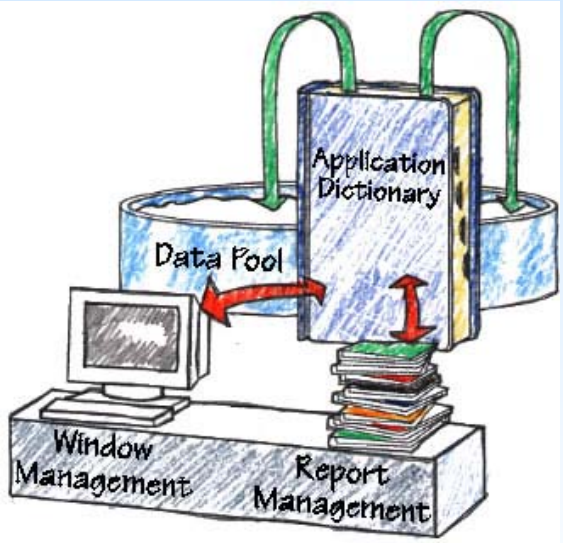
Document Configuration..

In Structured System Analysis

Data Dictionary

A document that contains the DFD and a description of all its components

- Data elements
- Data structures
- Data stores
- Data flows
- Process descriptions
- E-R diagram



Documentation…

Document Configuration..

Class Eg.

CAR
speed fuel color
Steer() PressGasPedal() PressBreak()

In Object Modeling

- Classes and their descriptions
- Attributes
- Methods

Other kinds of entries

- System dictionary for system users
- Data users can access
- Processes users use
- Forms
- Structure charts, program modules and file structures

UCSC

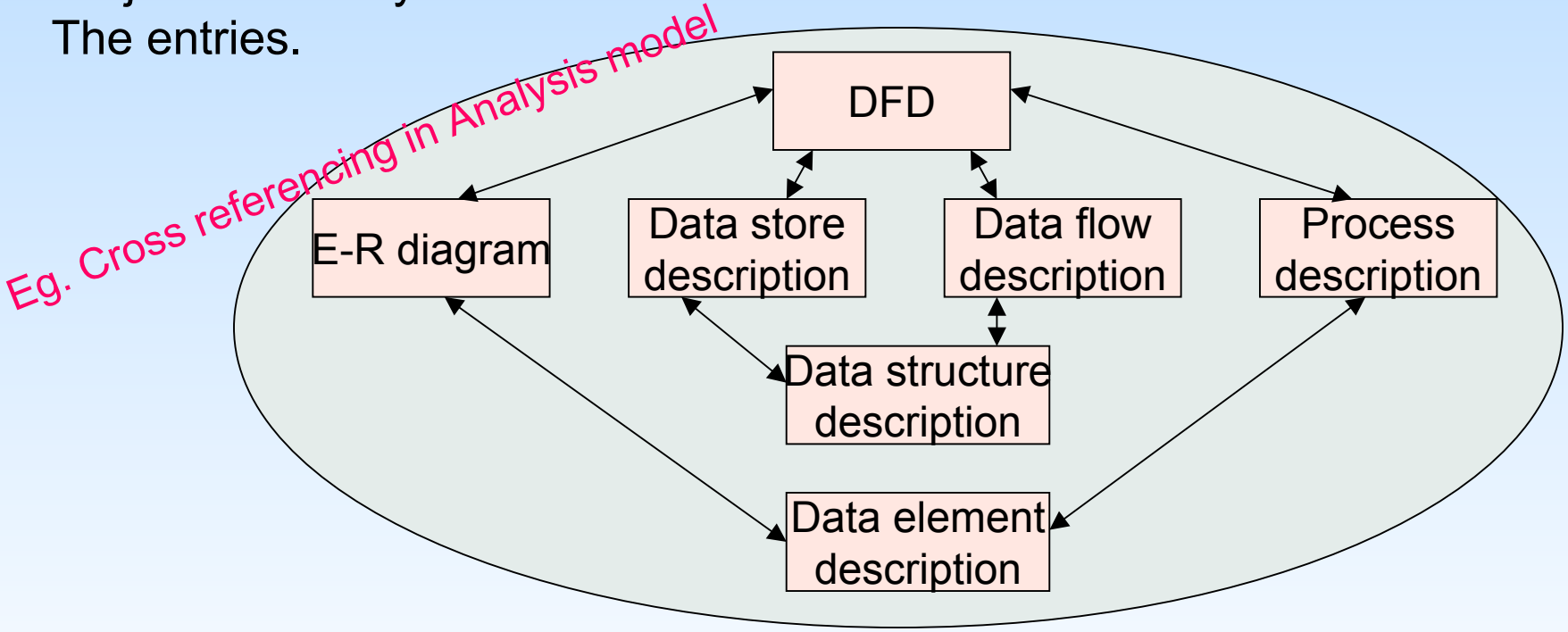
BIT

Documentation…

Document Configuration..

Cross Referencing

Project dictionary also maintains cross-references between The entries.



Cross –referencing between different parts of a model makes it easier to trace components related to any part of model

Documentation…

Using Document Configuration

Used for two purposes

- To support the actual analysis and design process and any changes to the system

- To produce reports needed by management

UCSC

BIT

Documentation…

Maintaining a Configuration:

– *Manual systems* :

Manual project dictionaries usually have one page for each project dictionary entry.

There is a separate entry for each data element, data structure, process description, user data flow and store.

Manual configuration systems are sometimes awkward to maintain.

UCSC

BIT

Documentation…

Maintaining a Configuration..

- *Automated System*

  Every change requires a form to be updated, and if there is more than one copy of the system dictionary this form must be updated in each copy.

  Any cross-references must also be updated.

Documentation…

Maintaining a Configuration..

– *Automated System…*

Many organization are now beginning to use computer tools to maintain document configuration. Thus:

- Easy to keep track of the documents
- Can use standard document structures
- Seen as extensions of the file management systems
- Have additional facilities to automatically keep track of versions

UCSC

BIT

Documentation…

Maintaining a Configuration..

–Automated System..

Document Structures are of two kinds

1. Word Processor systems
2. CASE tools

CASE tools will provide:

1. A repository of information
2. A tool for finding information
3. Carrying out cross checking
4. Providing design support

UCSC

BIT

Documentation…

Maintaining a Configuration..

–Automated System..

Computer support in structured system analysis

1. Provide a graphical interface for DFDs and E-R diagrams
2. Assist users to develop correct models

Eg. Easy Case tool

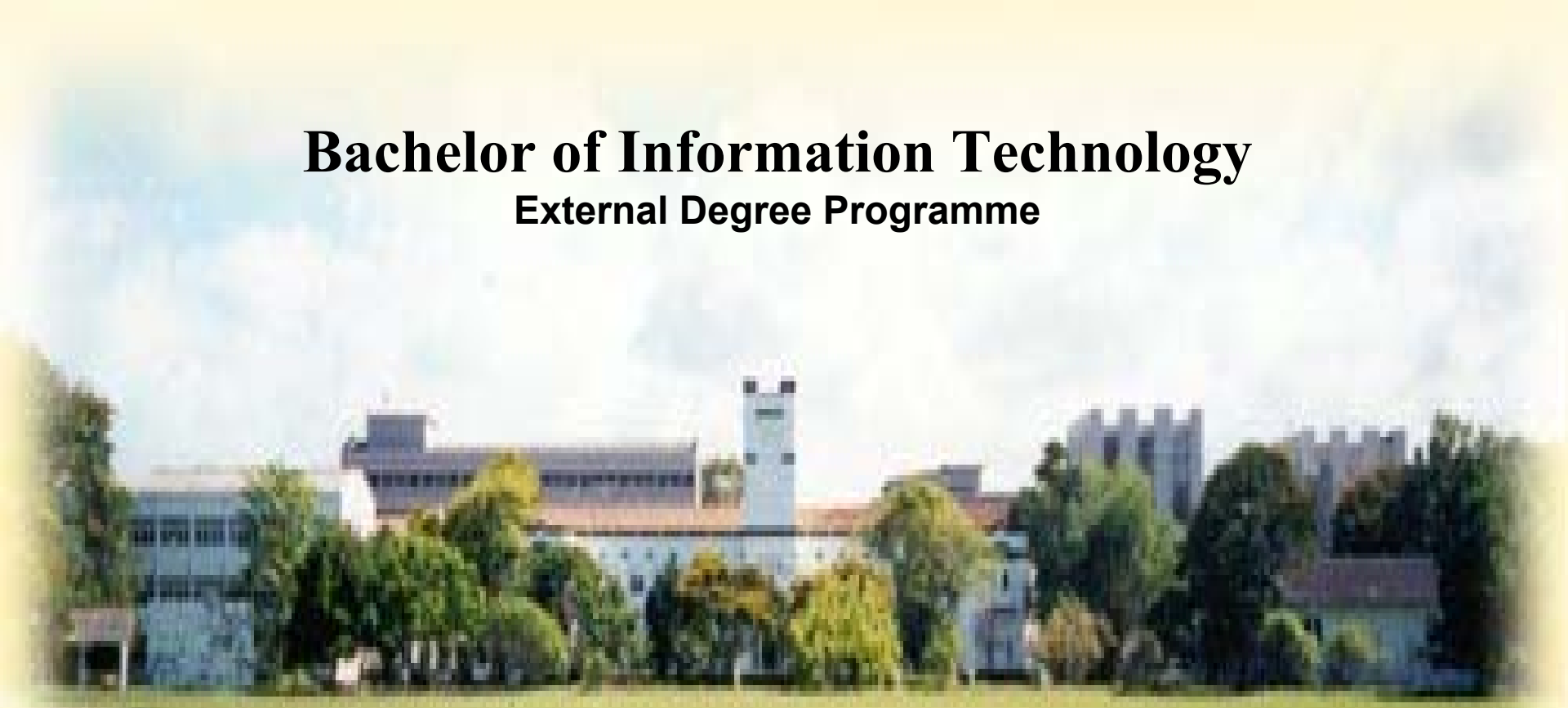
Components Of a tool of this kind

1. Various input modes
2. An evaluation mode
3. A reporting mode
4. A generation mode

For more details : Ref2 p263 - 276

Bachelor of Information Technology

External Degree Programme



Systems Analysis and Design

Ref_1 : System Analysis and Design Methods **By** Whitten, Bentley, Dittman ISBN 0-07-044539-7 (5th Edition)
Ref_2 : System Analysis and Design **By** Igor Hawryszkiewycz (4th Edition) ISBN 81-203-1670-3
Ref_3 : http://www.mhhe.com/whitten
Ref_4 : Practical SSADM Version 4 A Complete Tutorial Guide **By** Philip L Weaver ISBN 0-273-60095-8

UCSC

University of Colombo School of Computing

BIT

Automated Tools and Technology

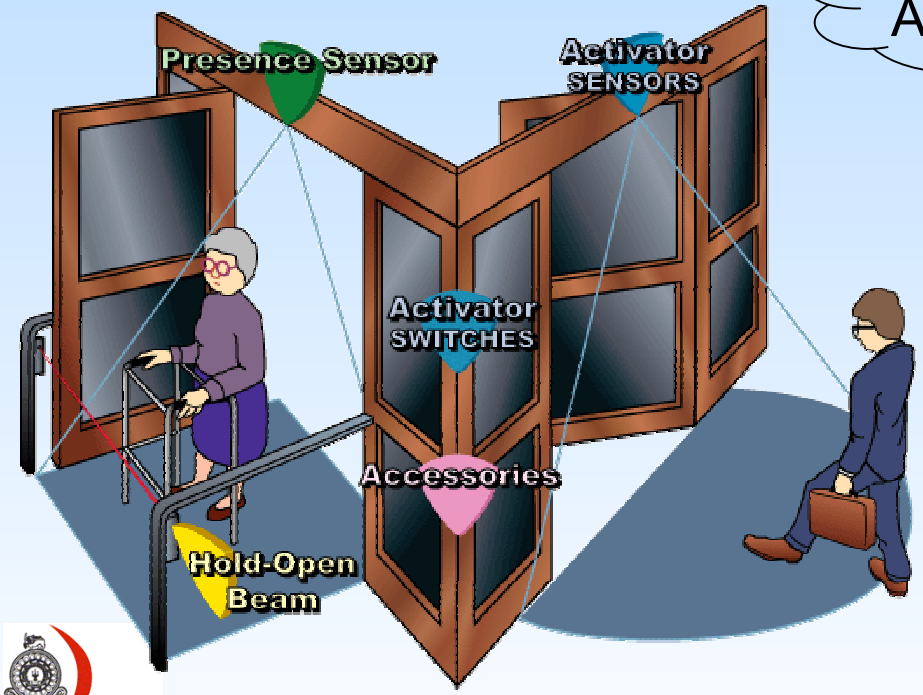
UCSC

BIT

Automated tools and technology

Computer-aided systems engineering (CASE)

It is the application of information technology to system development activities, technique and methodologies. *Case tools* are programs that automate or support phases of a system development life cycle.



It is filling Automatically

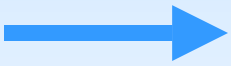
UCSC

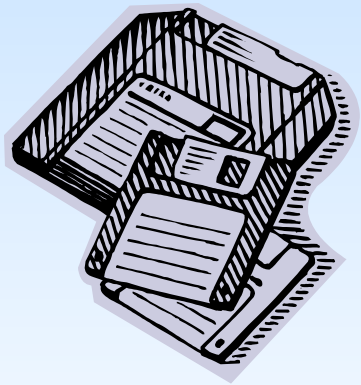
BIT

Case Repositories.

At the centre of the any true case tools architecture is a database called a repository.

a case repository is a system developers' database. It is a place where developer can store system models, detailed descriptions and specifications and other products of systems development.

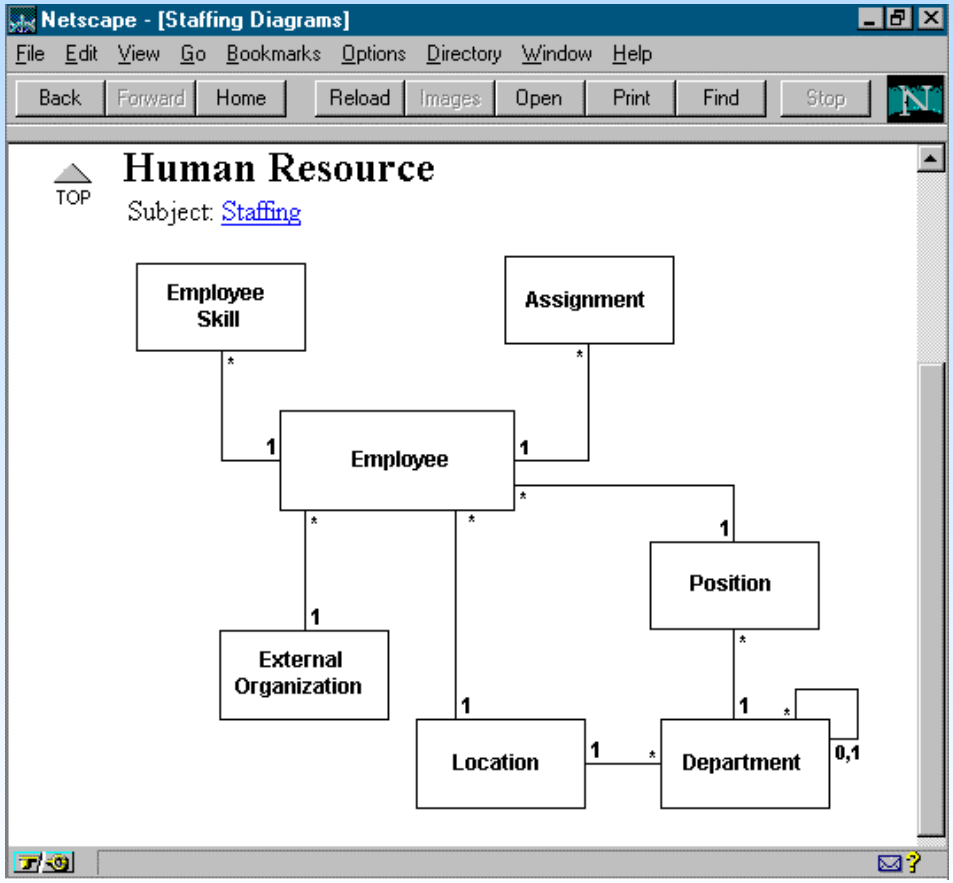




Case Facilities

✓ Diagramming tools:

They are used to draw the system models required or recommended most system development methodologies. Usually the shapes on one system model can be linked to other system models and to detailed descriptions.



Case Facilities….

✓ Description tools:

They are used to record, delete, edit, and output detailed documentation and specifications. The description can be associated with shapes appearing on system models that were drawn with the diagramming tool.

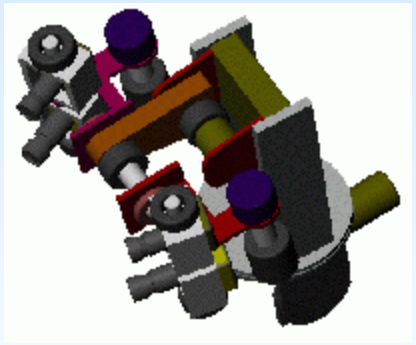
UCSC

BIT

Case Facilities….

✓ Prototyping tools:

They are used to construct system components including inputs outputs and programs. Today most of these prototypes can evolve into the final, working system.



UCSC

BIT

Case Facilities….

✓ Quality management tools:

They analyze models, descriptions and specifications, and prototype for completeness, consistency, and conformance to accepted rules of the methodologies.

UCSC

BIT

Case Facilities….

✓ Documentation tools:

They are used to assemble, organize and report repository information that can be reviewed by system owners, users, designers and builders.

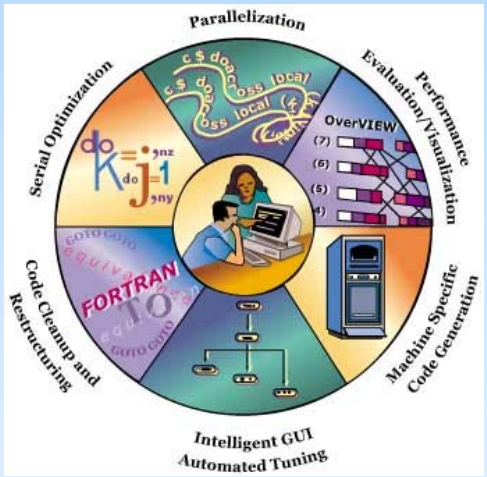
UCSC

BIT

Case Facilities….

✓ Design and Code generation tools:

They automatically generate database design and application programs or signification portion of those programs.



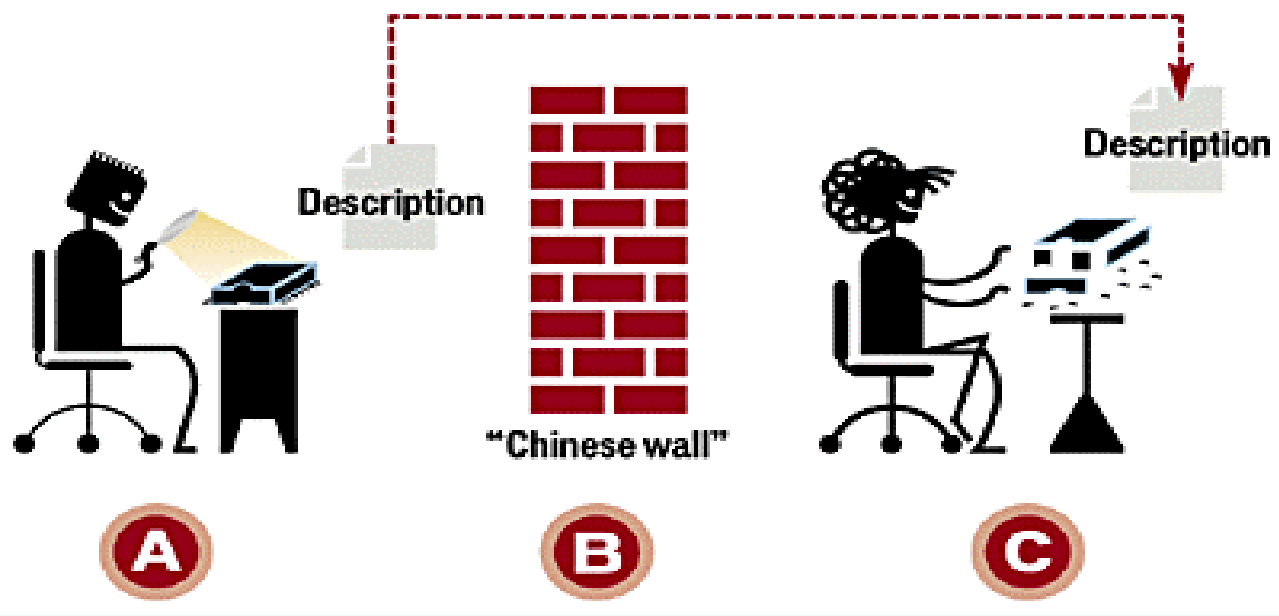
Forward and Reverse Engineering

Forward engineering:

It requires the systems analyst to draw system models, either from scratch or from templates. The resulting models are subsequently transformed into program code.

Reverse engineering:

It allows a CASE tool to read existing program code and transform that code into a representative system model that can be edited and refined by the systems analyst.



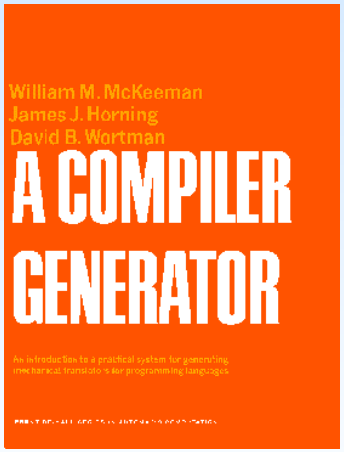
Application Development Environment (ADE)

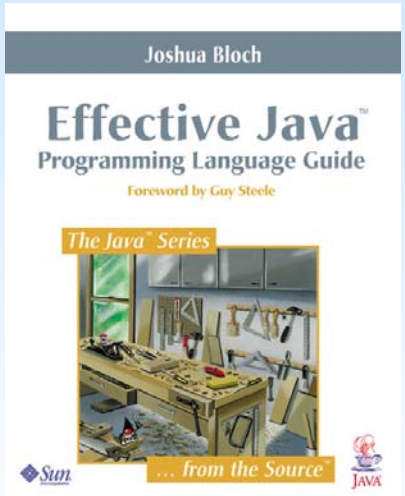
It is integrated software development tools that provide to all the facilities necessary to develop new application software with maximum speed and quality.

ADE facilities

Programming language or interpreters:

They are heart of the ADE . Powerful debugging features and assistance are usually provide to help programmer quickly identify and solve programming problems.

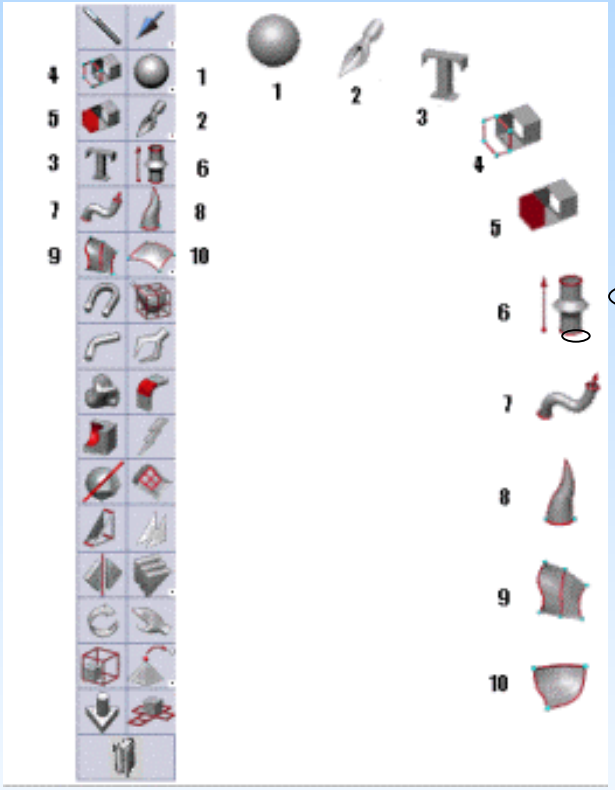


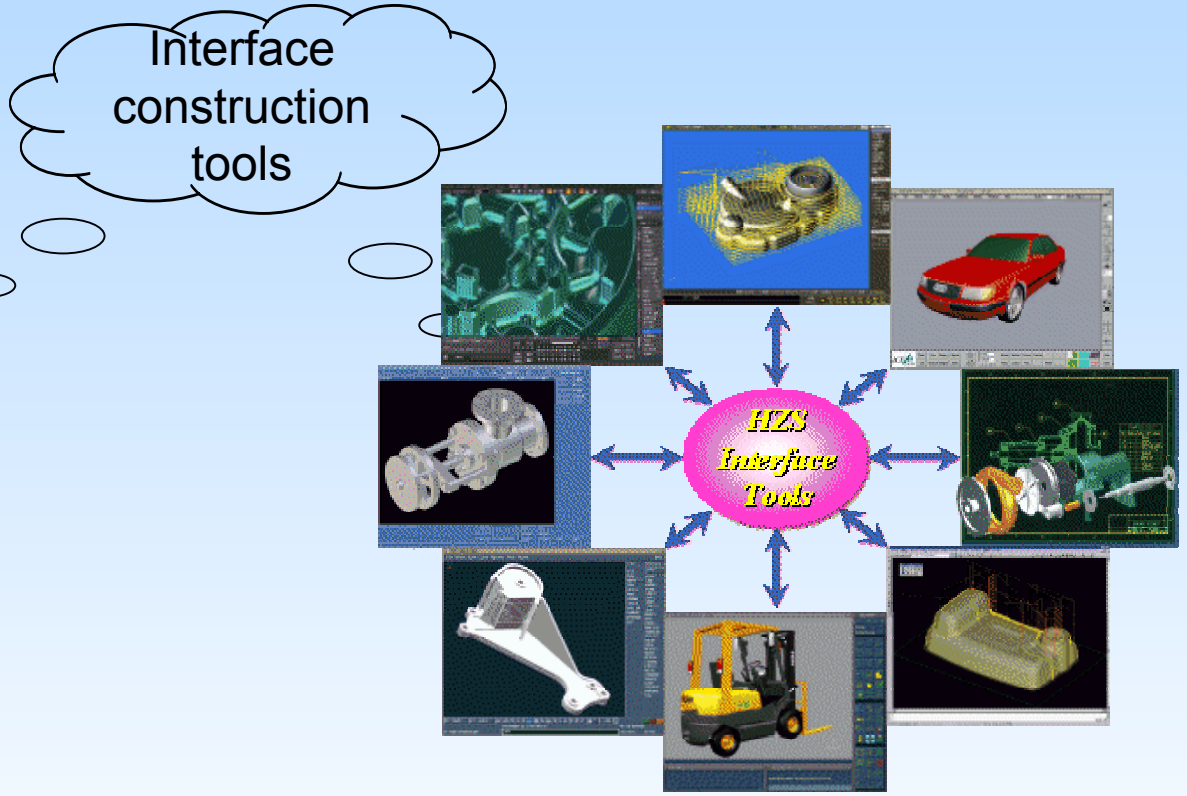


ADE facilities…..

Interface construction tools:

They help programmer quickly build the user interface using a component library.





ADE facilities…..

Middleware:

It is software that helps programmers integrate the software being develop with various database and computer network.

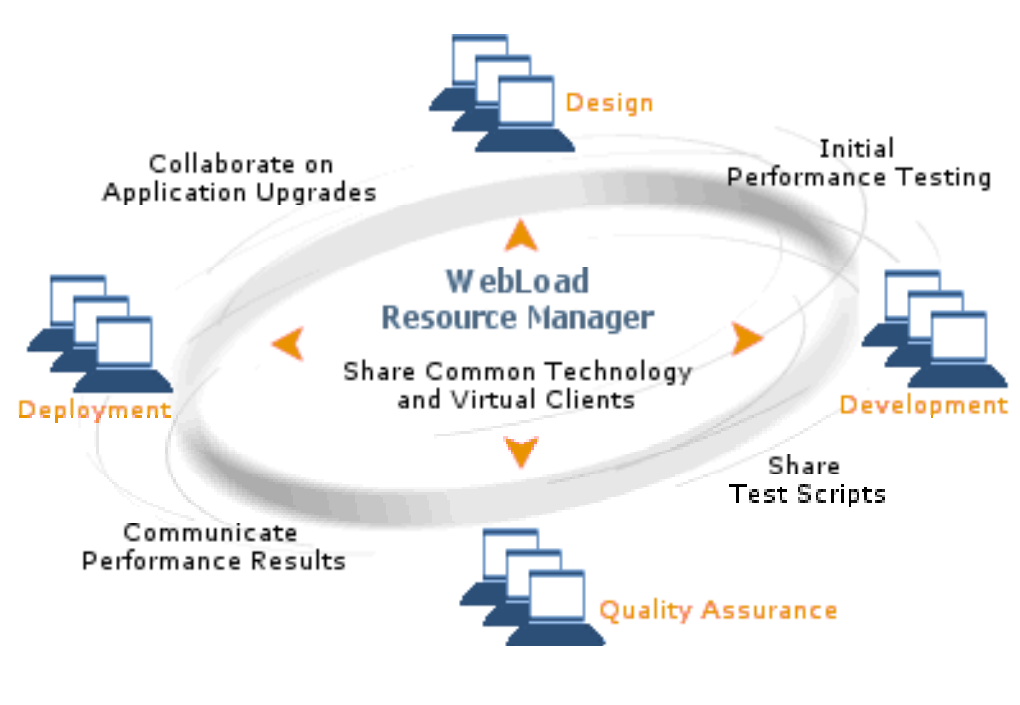
UCSC

BIT

ADE facilities…..

Testing Tools:

They are used to build and execute test script that can consistently and thoroughly test software.



ADE facilities…..

Version Control Tools:

They help multiple programmer teams manage multiple version of a program, both during development and after implementation.

Help Authoring Tools:

They are used to write online help systems, user manuals, and online training.

Repository Links:

They permit the ADE to integrate with case tool products as well as other ADEs and development tools.

Process management

It is ongoing activity that document manages the use of and improves an organization's chosen methodology for the system development.

It is concerned with

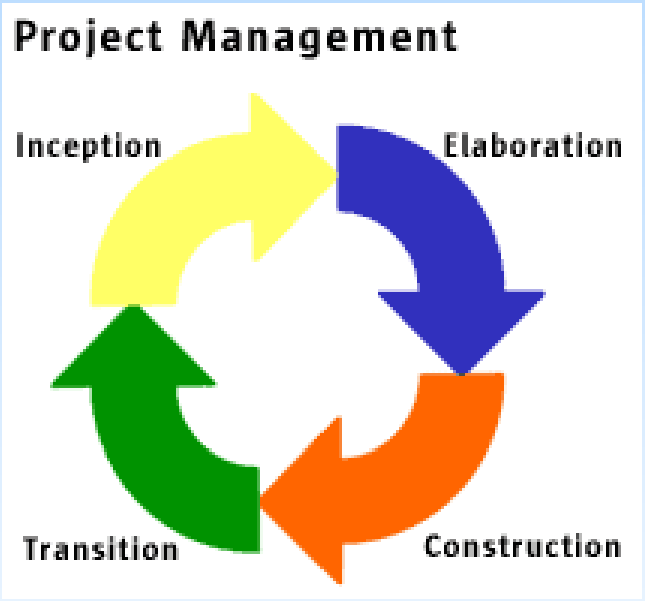
- ✓the activities
- ✓ deliverables
- ✓ quality standard to be applied to all projects

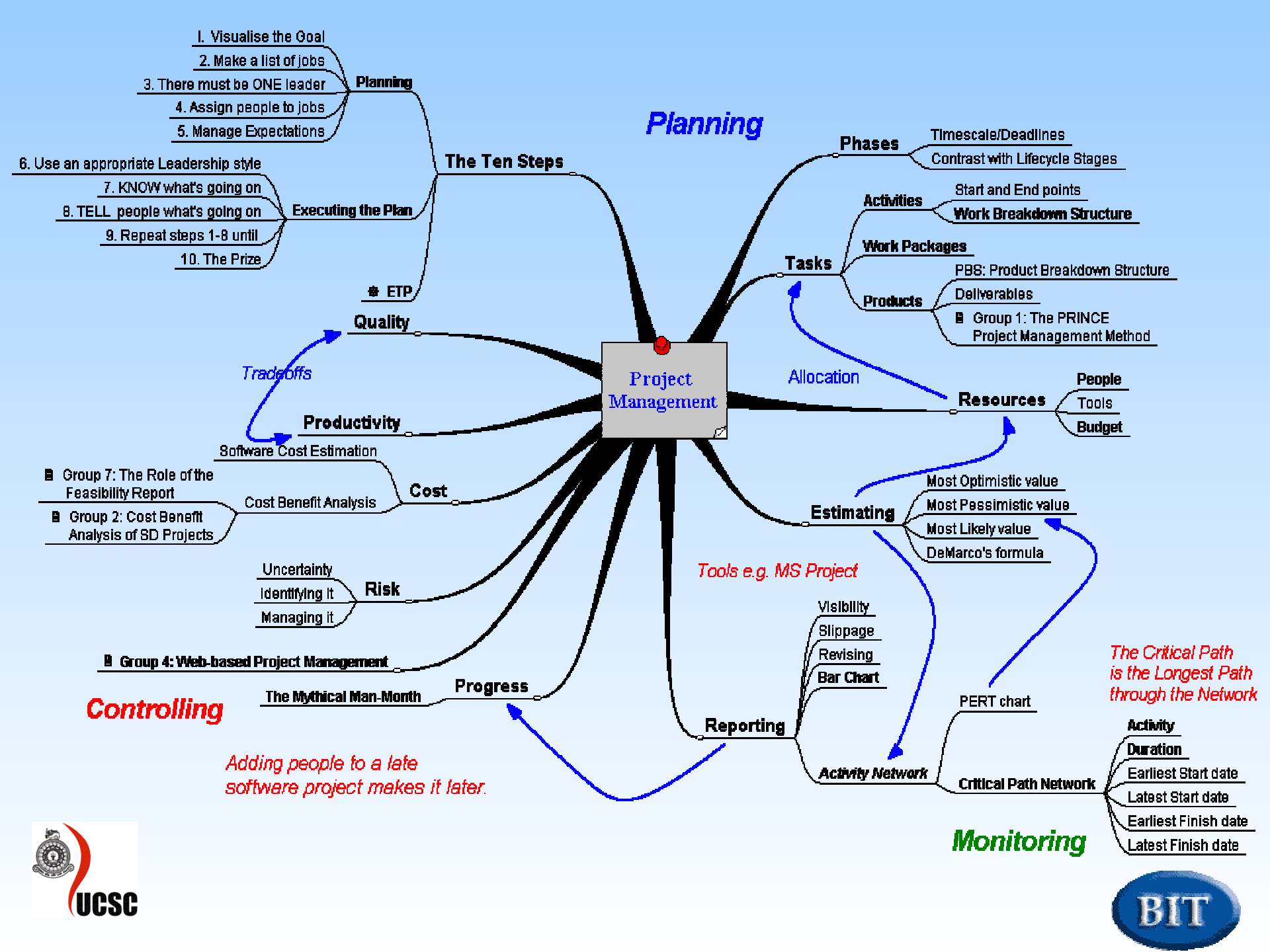
UCSC

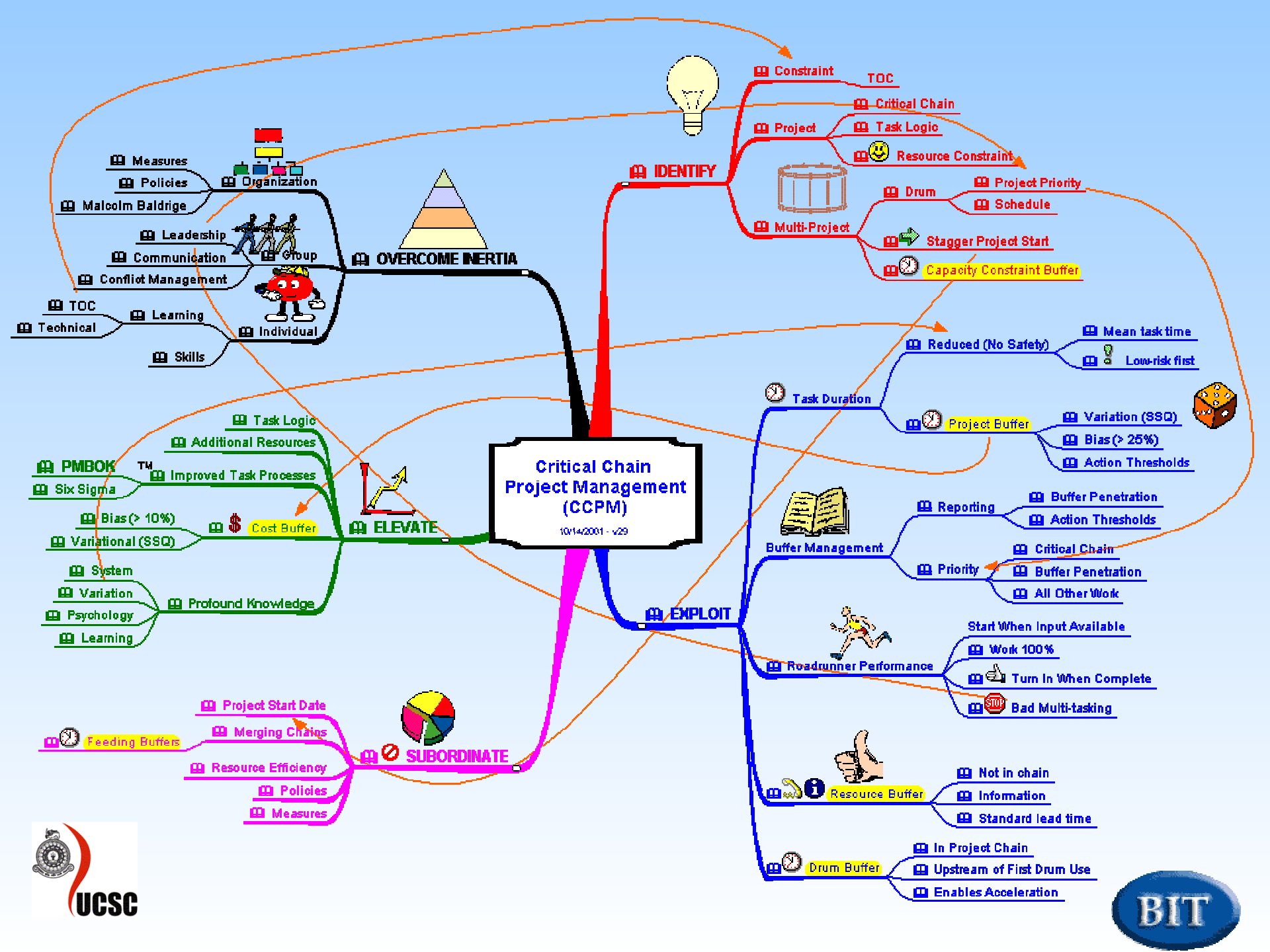
BIT

Project Management

It is the activity of defining ,planning directing monitoring and controlling a project to develop an acceptable system within the allocated time and budget.




Planning
Project Management
The Ten Steps
1. Visualise the Goal
2. Make a list of jobs
3. There must be ONE leader
4. Assign people to jobs
5. Manage Expectations
Planning
6. Use an appropriate Leadership style
7. KNOW what's going on
8. TELL people what's going on
9. Repeat steps 1-8 until
10. The Prize
Executing the Plan
ETP
Quality
Tradeoffs
Productivity
Software Cost Estimation
Group 7: The Role of the Feasibility Report
Cost Benefit Analysis
Group 2: Cost Benefit Analysis of SD Projects
Cost
Uncertainty
Identifying it
Managing it
Risk
Group 4: Web-based Project Management
The Mythical Man-Month
Progress
Controlling
Adding people to a late software project makes it later.
Phases
Timescale/Deadlines
Contrast with Lifecycle Stages
Activities
Start and End points
Work Breakdown Structure
Work Packages
Tasks
PBS: Product Breakdown Structure
Deliverables
Products
Group 1: The PRINCE Project Management Method
Allocation
Resources
People
Tools
Budget
Estimating
Most Optimistic value
Most Pessimistic value
Most Likely value
DeMarco's formula
Tools e.g. MS Project
Visibility
Slippage
Revising
Bar Chart
Reporting
Activity Network
PERT chart
The Critical Path is the Longest Path through the Network
Activity
Duration
Earliest Start date
Latest Start date
Earliest Finish date
Latest Finish date
Critical Path Network
Monitoring


Critical Chain Project Management (CCPM)
10/14/2001 - v29
IDENTIFY
Constraint
TOC
Critical Chain
Project
Task Logic
Resource Constraint
Multi-Project
Drum
Project Priority
Schedule
Stagger Project Start
Capacity Constraint Buffer
OVERCOME INERTIA
Organization
Measures
Policies
Malcolm Baldrige
Group
Leadership
Communication
Conflict Management
Individual
Learning
TOC
Technical
Skills
ELEVATE
Task Logic
Additional Resources
PMBOK
Six Sigma
Improved Task Processes
Cost Buffer
Bias (> 10%)
Variational (SSQ)
Profound Knowledge
System
Variation
Psychology
Learning
EXPLOIT
Task Duration
Reduced (No Safety)
Mean task time
Low-risk first
Project Buffer
Variation (SSQ)
Bias (> 25%)
Action Thresholds
Buffer Management
Reporting
Buffer Penetration
Action Thresholds
Priority
Critical Chain
Buffer Penetration
All Other Work
Roadrunner Performance
Start When Input Available
Work 100%
Turn In When Complete
Bad Multi-tasking
Resource Buffer
Not in chain
Information
Standard lead time
Drum Buffer
In Project Chain
Upstream of First Drum Use
Enables Acceleration
SUBORDINATE
Project Start Date
Merging Chains
Feeding Buffers
Resource Efficiency
Policies
Measures

Process Manager

Involved with the project's process,

- ✓It is an automated tool that helps to document and
- ✓ manage methodology and
- ✓ routes its deliverables and
- ✓quality management standards.

Eg. CS/10000

Project Manager

Involved with the project management,

It is an automated tool

- ✓to help plan system development activities estimate and assigns resources
- ✓ schedule activities and resources
- ✓ monitor progress against schedule and budget
- ✓ control and modify schedule
- ✓resources and report project progress.

Eg. Microsoft's Project

UCSC

BIT

Benefits of Case Tools

- ✓**improve productivity :-** through automation of tasks.
- ✓**Improve quality:-**because automated tools check for completeness consistency and contradictions.
- ✓**Better and more consistent documentation:-** because the tools make it easier to create and assemble consistent high quality documentation.
- ✓**Reduced lifetime maintenance :-**because of the aforementioned system quality improvement combined with better documentation.
- ✓**Methodologies that really work :-** through rule enforcement and build– in expertise.

UCSC

BIT