

I. Introduction

Types of Software
Information Systems in Organizations
Requirements Analysis and Design
Roles of a Systems Analyst
Notations, Methods and Tools
Information System Lifecycle Phases



Software Everywhere!

- ❑ Software started off as an artifact used only by IT professionals in the '60s and '70s.
- ❑ With PCs and breakthroughs in usability, it became a fundamental component of our work day in the '80s and '90s.
- ❑ In the new millennium, it literally threatens to run our lives!
- ❑ There are different types of software:
 - ✓ System software: Operating systems, network drivers,...
 - ✓ Middleware: Compilers, database management systems, office software,...
 - ✓ Applications: Accounting, inventory control, personnel management,...
- ❑ **Information systems** are software applications which manage large amounts of data.
- ❑ Most of the software out there is information systems software, written in languages such as COBOL, RPG, 4GLs and the like.

Information Systems

- ❑ Large organizations own and operate hundreds or even thousands of information systems.
- ❑ By-and-large, these were developed independently of each other, are legacy systems (sometimes more than 30 years old) and are not well integrated.
- ❑ Large organizations spend as much as 10% of their budget for IT, much of which goes towards the development, operation and maintenance of their information systems.
- ❑ Worldwide spending on IT is ~\$2TUS (in 2000).
- ❑ IT accounts for 6.6% of GDP worldwide; it's 9% in the US.
- ❑ Systems analyst jobs grew by ~30% between 1985 and 2000; during the same period, overall job market grew by ~15%.

Why Do Organizations Need Information Systems?

- ❑ For feedback!
- ❑ A large multinational bank wants to know its assets at the end of the year/month/day(...??)
- ❑ A car manufacturer who is supposed to produce 1,000,000 cars by the end of the year, wants to know how many it has produced so far, how many it has sold so far, actual (as opposed to budgeted) production costs,...
- ❑ (Closer to home...) A software company that is supposed to release a new version of its popular software product by April 1, wants to know whether development is on target or behind, so that it can take appropriate measures.

Information systems make organizations run!

The Bad News

- ⊗ 30% of large IT projects are cancelled before completion
 - ⊗ 50% of IT projects are over-budget by more than 200%
 - ⊗ The majority of completed projects deliver 60% or less of prescribed functionality
 - ⊗ Many delivered information systems are under-used because they don't meet user needs and/or expectations
 - ⊗ Legacy systems are a serious and growing bottleneck to organizational evolution
- etc.

Information Technology is failing us!

Why is this Course Important?

- Most errors (54%) are detected after coding and testing.
- Almost half of all errors in software (45%) are in requirements and design.
- Most errors made during requirements analysis are non-clerical (77%) and may arise because of incorrect facts, inconsistencies, omissions and ambiguities.
- Requirements errors can cost up to 100 times more to fix than implementation errors -- if they are not caught early on.
- Requirements errors **can** be detected, because inspection techniques have proven most effective for any software, and inspection techniques can be applied to requirements as well as design and code.

Need to do requirements and design right!

What is Systems Analysis?

- The collection of notations, methodologies and tools used to gather details and analyze a problem situation prior to information system design and implementation
- **Systems analysis** (or, **requirements analysis**) must ensure that the proposed information system meets user needs, can be delivered on time, and can be updated inexpensively.
- Problems in "getting the systems analysis right", such as ill-defined situations, ambiguities, inconsistencies, mixing requirements with design

**Remember, finding and fixing
a fault after software delivery
is 100 more expensive than finding and fixing it
during systems analysis or early design phases**

What is the Result of Systems Analysis?

- The result of an information system analysis is a **requirements definition** (or, "requirements")
 - How is a requirements definition used?
 - As a statement of the problem to be solved
 - For communication between designer and end-users
 - To support information system evolution
 - To support design validation
- What goes in a requirements definition?
 - Functional requirements:** What does the system do? What information is maintained? What activities are carried out? What interfaces are supported?
 - Non-functional requirements:** Global constraints on the system, such as performance constraints, (resource constraints, security, reliability,...), operational constraints (hardware requirements, personnel,...), life cycle constraints etc.

What is System Design?

The specification of the information system to be built. This specification includes:

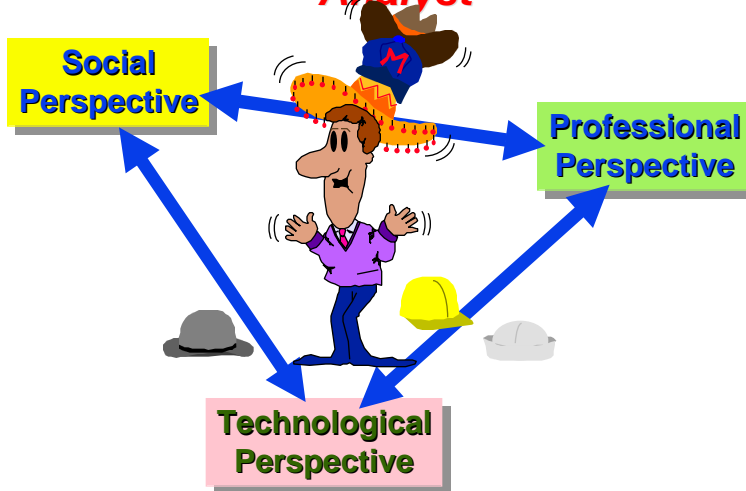
- The **hardware configuration** on which the system will run, including network interfaces.
- The **software platform** on which the system will run, i.e., operating system, DBMS, programming language, etc...
- The **software architecture** of the proposed system, including interfaces between the system modules.
- The **function** of each module, i.e., what does each module do, i.e., transformations it performs on its inputs.
- The **database(s)** that will be part of the information system, stored in database management systems (DBMSs) or in files
- **User interfaces** that need to be in place to facilitate use of the system by different user groups

Roles of the Systems Analyst

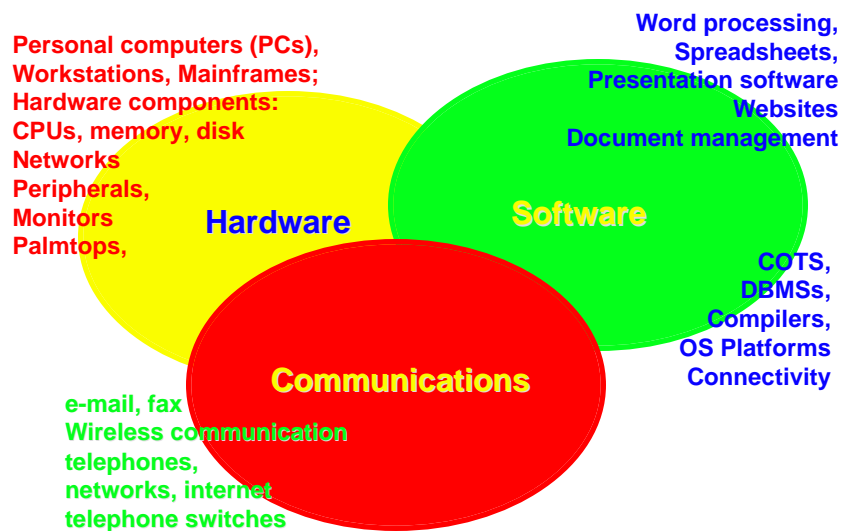
- **Consultant** -- often hired from outside, specifically for a project; this means that she brings a new perspective but will not be familiar with company culture/politics.
- **Supporting Expert** -- knows well relevant hardware and software technologies, advises on alternative hardware/software configurations.
- **Change Agent** -- will be expected to suggest alternative business processes which improve on current practices (business reengineer).

Systems analysis is, above all, a problem solving activity

The Background of the Systems Analyst

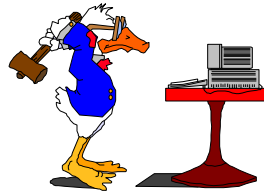


Technologies for System Analysis



Trends

- Hardware, connectivity, portability increasingly taken for granted.
- Less emphasis on implementation, more emphasis on design and analysis, for **work processes** and **information services**, not just information systems!
- Greater demand for “people skills” as opposed to “technological skills”
- More packaged applications (Enterprise Resource Planning, or ERP, systems) sold by companies like SAP, PeopleSoft and Oracle.



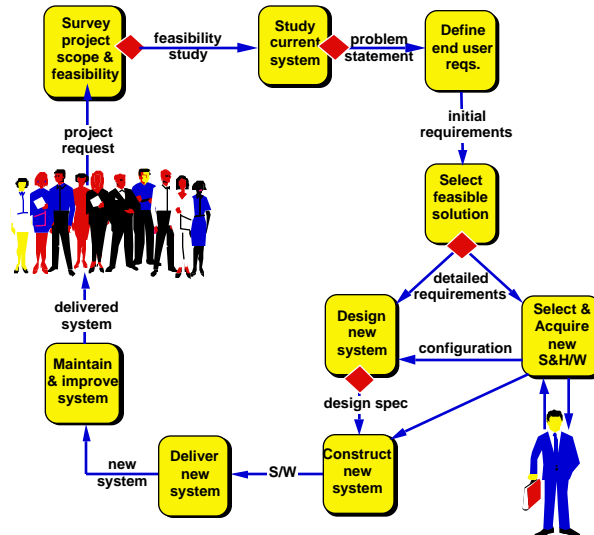
Notations, Methodologies, Tools

Systematic information system development is based on notations, a methodologies and associated tools

- **Notation** -- used to describe the information captured during different phases; notations range from natural language, to diagrammatic notations (such as entity-relationship or data flow diagrams), or formal languages, such as programming languages.
- **Methodology** -- this determines the process whereby the software developer creates, refines, analyzes and validates a software system; methodologies are often project- or situation-specific
- **Tools** -- introduced to support the creation, refinement, analysis and validation of software (such as CASE tools).

Generally, software engineering practice does poorly with respect to all of the above!

Information System Lifecycle Phases



Readings

1. Lecture units 1-4 (available at the course website.)
2. Textbook, chapters 1-3.