Chapter 9;

Applications and the Organization

Lecture 14
TIM 50  Autumn 2012

Tuesday  November 12, 2012
Announcement

1. Assignments submitted later than deadline will have 50% points at most.
2. The Book by By Musacchio is completed today and We will returned to Business Intelligence

DB assignment is due two weeks later (11/20/2012).

3. The grades for every assignment will be given in eCommons.
4. It's important to check webpage to get the latest information and assignments changes.

No Class on Thanks Giving Day, 11/22 Thursday
The Computer and Communications Industries

Review

Computer/Networks/Data Communications

Architects/Layered H/W, S/W Components/Modules

Stovepipe Structured Systems Integrated Infrastructures (Networked/Distributed)

standardizations

Designed/Prepd. Inside My Company Purchase from Suppliers

Interpretability System Integrations Data Formats Data Regenerations

Use Products Across Diff. Industries System integration Interoperability Interface design Reference Model

Component Functionality System of Systems De facto STND De Jure STND

Layered Architecture Components Interface

All Comps. Made in a Company Vertical Integrations

Future I.T. S. Open Structure Flexibility Media Combined Virtual,3D World
Contents are re-edited using materials from Internet Sites
Participant, Products, and Services

Components and Integrations

Components

- Components provide a service without regard to where the component is executing or its programming language
  - A component is an independent executable entity that can be made up of one or more executable objects;
  - The component interface is published and all interactions are through the published interface;
Architecture provides a set of laws/boundaries for solutions, to ensure that all components of the solution will work together.

Catalogs, provide **detailed specifications** of the available parts together with a variety of alternative products implementing those specifications; **plus recommendations on how to use these parts** (reference architectures).

Solutions to many different requirements, all **built from the same common collection of parts**, each designed according to accepted standard reference architectures.
IT Architecture must consider the views of different users and their needs

Each view has a different architectural representation, just as an electrical wiring diagram is different from the interior designer’s “picture”.

- What processes, information, roles and locations must be addressed?
- What information needs to be made available, to whom, and how?
- What are the data elements, where stored, and how are they accessed?
- What infrastructure do we need to provide the required Service levels?
- What are the security and privacy requirements for the infrastructure and applications?
- How do applications support the required functionality?
- How do we integrate with external systems and databases?
- What types of content, how is it maintained, published, and distributed?
- How do you manage and administer the infrastructure and the application?
What is System Integration?

INCOSE SE Handbook:
...establishment of system interfaces, internal and external...
...emphasis on risk management and continuous verification...

The process of putting a system together, with techniques to ensure all the parts work as a whole.
Integration is Hard

Generally, the main contractor for the project is responsible for systems integration.

The sub-contractors will usually be part of the integration team.

Integration is one of the most costly and time-consuming activities in the systems engineering process.

For large and complex systems, up to 40% of the development effort may be used in this activity, mostly in system testing.
What is Interoperability?

“Ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together.”*

*(JCS Pub 1)*
**System-of-Systems**

CJCSI 3170.01D

**System-of-Systems.**

A set or arrangement of interdependent systems that are related or connected, to provide a given capability. The loss of any part of the system will degrade the performance or capabilities of the whole.

**Key Points**

- Arrangement of systems
- Related or connected systems
- Provide a given capability

- Loss of part
- Degrades the whole

**One Function Concept**

Limited Redundancy
The Role of Architecture
from stovepipe to Integrated Infrastructure

Less Vertical Integration and more diversifications

Venture capital and stand-ups

Repositories

Also include content disseminations
Digital Object Architecture Business Case

- Provides useful, authoritative, visible, and dynamic access to information
  - Encourages the innovative, value-added, dynamic and, therefore, effective use of digital information
  - Provides global visibility/coordination over information
  - Controls access to information
  - Reuses generated content, techniques, and disseminations
  - Preserves information functionality
  - Discourages generation of multiple, and potentially different, versions of information
- Provides interoperability and platform independence
- Provides greater security
  - Supports real-time access control changes
  - Supports partial dissemination
Equipment Convergences

- Computing/Communication
- Computing/Communication/Video
- Computing/communication/Media
- ....... For the Future!
<table>
<thead>
<tr>
<th>Functionality</th>
<th>PC/Notebook</th>
<th>Cell Phone</th>
<th>Handheld devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerPoint</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Camera</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mp3 player</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Spreadsheets</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Word processor</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>GPS navigation</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Email</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
What is a standard?

Mostly used in two rough senses:

- it is an agreement developed by several parties with the intent that all parties comply
- it is a product or service with a significant market share

There are many definitions. Most try to specify the first type of standard (i.e. the agreement).
Many definitions of *standard*

Definition depends on who is asked, e.g.

- Formal standards bodies: a standard is “a document established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context” (ISO/IEC, 2004b, p.8)

- Industry: “A standard [can be] of any form or type (...) A standard is also one of the agents used (...) to bring about market change” (Cargill, 1989, p.41)
Benefits of Standardization

For product standards the benefits may be broadly summarized under the headings variety reduction, interchangeability, and availability:

the effects of variety reduction are well known and can mostly be assessed in terms of hard cash, taking due account of the additional cost reduction effects on associated parts and operations;

interchangeability as a result of standardization leads to higher productivity and lower manufacturing costs. These benefits are relayed to the users as a result of increased competition. Interchangeability is also very important to erection, installation, maintenance, and repairs;

easy availability implies that an adequate number of varieties are always in stock. This means shorter lead times and less capital tied up on the user side.
Why are standards important? Standards create *compatibility*

**Compatibility defined as**

- ‘the suitability of products, processes or services for use together under specific conditions to fulfill relevant requirements without causing unacceptable interactions.’ (ISO/IEC, 1991)

- **Two types of compatibility** between components (David & Bunn, 1988)
  - *compatible complements* (e.g. plug and socket)
  - *compatible substitutes* (e.g. plug A and B in respect to socket)
Why are standards important?
Standards create compatibility

• Standards create *Compatibility*

  – Networked environments* like telephone communication and broadcasting require standards
  – Standards coordinate technology (Schmidt & Werle, 1998)
  – Standards coordinate markets, e.g. availability of complementary products
  – Standards facilitate international trade
Many kinds of standards:

• What aspect is standardized? (subject matter classification)
  – Product and performance standards
• What type of standard is at stake? (standard development classification)
  – *De facto* and *de jure* standards
• When does standardization take place? (standard development classification)
  – Anticipatory - Enabling - Responsive standardization
What aspect is standardized?

Product vs. performance standards

• *performance standards* (ISO/IEC, 2004b): standards that specify the required performance of a product or service

• IEC 61753 – Fibre Optic Interconnecting Devices

• *product specifications* (ISO/IEC, 2004b): design or descriptive characteristics of a product or service

• IEC 60908 – Compact Disc Digital Audio System

In general, *product specifications* restrict technology development more than *performance standards*
What type of standard is at stake?

*De facto* standard

- *De facto* versus *de jure* standards incorrectly explained as *market*-versus *committee-based standards*

- *De facto* means: in practice, in reality
  - *de facto standard*: product or service with a large market share
  - Incorrectly associated with (a) proprietary standards [whereas non-propr. Specs, e.g. open source, can also become *de facto* standards]
  - Incorrectly (b) *solely* associated with market standards [whereas committee standards can also become *de facto* standards]
What type of standard is at stake? *De jure* standard

- *De jure* means: by law, by regulation
  - *de jure standard*: standard imposed by law

- Incorrectly associated with (a) *all* committee-based standards of formal standards bodies [whereas only a very small proportion of formal standards is referenced in law]

- Incorrectly associated with (b) *non* market-based standards [whereas standards committees are usually dominated by industry]
Applications and the Organizations

Network Computing Organizations

Central/ Networked Stove piped/Integrated structure

Acquisition Options
Make/Buy, and between

Application Life cycle

Alternative Development Methodology
**Operations through Networks Trends**

Platform-Centric  
Stove piped

Net-Centric  
Integrated structures

Traditional stove-pipe approach v. Fused information available on the Net

Source: Margaret Myers, Principal Deputy OSD DCIO
System-Centric, “Stovepipe” Architecture Problems

- Pre-web, client/server, terminal/mainframe paradigm
  - Content is typically application specific and relies on specific systems and platforms to provide access, access control, and presentation
  - Content is often not visible to the web, i.e. “Deep Web” problem
- Content lacks flexibility and extensibility. Difficult to do versioning, dynamic creation, appropriate disseminations, and support intra/inter-organizational reuse or value-adding
- Content quickly becomes unauthoritative as it is downloaded and reused
  - Outdated
  - Insecure
- Architecture lacks coherent preservation strategy
What are the Characteristics of a Net-Centric Information Space?

- Content is:
  - **Persistent**
  - Visible
  - Authoritative
  - Extensible
  - Reusable
  - Interoperable across platforms

- Content supports:
  - Dynamic relationships
  - Value adding
Vision for future eBusiness Supply Network

(B2B relationships)

Private eMarket (vertical)

Component Supplier

Raw Materials Supplier

Manufacturers

Engineering/Design Partners

Virtual Manufacturers

Virtual Distributors

Distributor/Reseller

Public eMarket (B2B)

Logistics Providers

Logistics Exchange (horizontal)

Customer Exchange (B2C)

Retailers

Consumer

Subcontractor

Manufacturer/OEM

Product/Goods Flow

Information Flow

3P Logistics

(A adapted from Forrester, 06/2000 & AMR SCM Report Jan 2000)
Vision for Collaborative Processes & Models in Business to Business

Collaborative Product Engineering & Design

Collaborative Supply Planning
Collaborative Supply Procurement
Synchronized Production Scheduling

Collaborative Logistics Planning
Collaborative Transportation Management
Collaborative Distribution Services

Collaborative Demand Forecasting
Collaborative Replenishment
Collaborative Promotion Planning
Collaborative Life Cycle Planning

Viewed from the manufacturer’s perspective

Suppliers
Subcontractors
Engineering/Design Partners

eMarket/Private Hub

Manufacturers
Retailers
Distributors

CPFR
Collaborative Planning, Forecasting & Replenishment

CSUP
Collaborative Supply & Production Planning

CTM
Collaborative Transportation Management

CED
Collaborative Engineering & Design
Layered System requires Components/Software Qualities

- must be planned for
  - Reusability
    - ability to construct new software from existing pieces
    - occurs at all levels: from people to process, from requirements to code
  - Interoperability
    - ability of software (sub)systems to cooperate with others
    - easily integratable into larger systems
    - common techniques include APIs, distributed programming interfaces (CORBA, DCOM), plug-in protocols, etc.
Qualities

• Scalability
  – ability of a software system to grow in size while maintaining its properties and qualities
  – assumes maintainability and evolvability
  – goal of component-based development
The Impact of Information Technology

• Who develops Information Systems?
  – In-house applications
  – Software packages
  – Internet-based application services
  – Outsourcing
  – Custom solutions
  – Enterprise-wide software strategies
  – How versus What
Information System Components

• A System is a set of related components that produces specific results

• A Mission-critical system is one that is vital to a company’s operations

• Information systems have five key components: hardware, software, data, processes, and people
Information System Components

Figure 1-9
Information System Components

• Hardware
  – Everything in the physical layer of the information system
  – Moore’s Law accurately predicted that computer processing power would double every 18 to 24 months
Information System Components

• **Software**
  – System software
  – Network operating system
  – Application software
  – Enterprise applications
  – Horizontal system
  – Vertical system
  – Legacy systems
Information System Components

• Data
  – The raw material that an information system transforms into useful information
Information System Components

• **Processes**
  – Describe the tasks and business functions that users, managers, and IT staff members perform to achieve specific results

• **People**
  – Users, or end users, are the people who interact with an information system, both inside and outside the company
Understanding The Business

• New Kinds of Companies
• Companies are classified based on their main activities:
  – Production-oriented
  – Service-oriented
  – Brick-and-mortar
  – Dot-com (.com)
Administrative Justification for Networked Computing

*Specialization
*Compartmentalization
*Locality
*Sharing
*Security
*Availability
Choices in Systems Acquisition

Corporate strategy

- Product scope
  - Specialization, horizontal integration, diversification
- Geographical scope
  - Local market or global markets
- Vertical scope (value chain)
  - Short value chain or long value chain (vertical integration or outsourcing)
Integration of components systems

Horizontal integration

- Cost reduction (economies of scale, reduction of duplication)
- Value (product bundling, cross selling)
- Managing rivalry
- Increased bargaining power
- Drawbacks – implementation (success of M&As, conflict with regulatory authority)
Stove Piped systems

Vertical integration

- Types
  - Backward, forward
  - Full, partial (taper)
- Benefits
  - Barriers to entry
  - Investment in specialized assets
  - Quality
  - Improved scheduling
Disadvantages of vertical integration

- Inefficiency
- Loss of flexibility (technological change)
- Demand uncertainty
- Compounding of risk
- Bureaucratic costs
Acquisition Options

Make Vs. Buy and Gray Region

Purchase/ Outsourced Development/ Internal Development

*Fixed price
*Time and material Prices
Acquisition Options

• Explain the differences among the alternatives to tailored system development:
  outsourcing, licensing ready-made software, contracting with an application service provider, and encouraging users to develop their own applications

• List the business trade-offs in the various methods of acquiring systems
Objectives

• Describe which systems acquisition approach is appropriate for a particular set of circumstances
• Discuss organizational policies on employee computer use
Options and Priorities

• Four alternatives to in-house development
  – Outsourcing
  – Licensing
  – Using ASP
  – Allow users to develop

• Deciding factor usually cost

ASP: Application service provider, a business that provides computer-based services to customers over a network
* License low cost and immediately available
  Best choice

* ASP immediately available and small startup fee

* Third best option is allowing users to develop

Outsource if non-IT employees cannot develop IS
Many factors in addition to cost and quality
Alternatives not fully comparable
Outsourcing

• Outsourcing – the practice of turning over responsibility of some to all of an organization’s information systems development and operations to an outside firm
Outsourcing

• Why Outsource?
  – Cost and quality concerns
  – Problems in IS performance
  – Supplier pressures
  – Simplifying, downsizing, and reengineering
  – Financial factors
  – Organizational culture
  – Internal irritants
Outsourcing

- **Outsourcing** has two meanings:
  - Commission development of application to other organization
  - Hire services of other company to manage services
- May not encompass development
Outsourcing
Custom-Designed Applications

• **Custom-designed**: developed specifically for organization
  – More expensive
  – Several advantages
    • Good fit to need
    • Good fit to culture
    • Dedicated maintenance
    • Smooth interface
    • Specialized security
    • Potential for strategic advantage
Outsourcing
Custom-Designed Applications

• Tailored development requires organization to fund all costs
• IS personnel constantly in use
• Production schedules may be delayed
• Less likely to be compatible with other organizations
• Offshoring: outsourcing to India, China, Philippines, etc.
Outsourcing IT Services

• Many *businesses turn to IT companies for long-term services*
  – Purchasing and maintaining hardware
  – Developing
  – Maintaining and operating Web sites
  – Staffing help desks
  – Running IT daily operations
  – Managing customer and supplier relations

• **Business process outsourcing**: outsourcing routine processes
Outsourcing IT Services

- Some companies realize IT is not primary focus
- Pace of development in IT requires high level of expertise
- Growing portion of IS budget allocated for outsourced services

- Popular IT service providers
  - IBM
  - EDS
  - Accenture
  - Unisys
  - AT&T
Outsourcing IT Services

- **Outsourcing companies known as vendors**
- **Typically long-term contractual relationship**
  - 7-10 years
- **Clients often bound by obsolete contracts**
  - Can renegotiate
Advantages of Outsourcing IT Services

• Several advantages of outsourcing
  – Improved financial planning
    • Know exact cost of IS functions
  – Reduced license and maintenance fees
    • IS professionals pay discounted prices for tools
  – Increased attention to core business
    • Executives don’t have to manage
Advantages of Outsourcing IT Services

• Advantages of outsourcing
  – Shorter implementation cycles
    • IT vendors complete new applications faster
  – Reduction of personnel and fixed costs
  – Increased access to highly Availability of ongoing consulting as part of qualified know-how
  – standard support

• Sometimes outsourcing does not save client money
Risks of Outsourcing IT Services

• Disadvantages of outsourcing
  – Loss of control
    • High risk in quickly changing industry
  – Loss of experienced employees
    • Transfer employees to vendor
  – Risks of losing a competitive advantage
    • May disclose trade secrets
Risks of Outsourcing IT Services

• Disadvantages of outsourcing
  – High price
    • Outsourcing may cost more than in-house development
    • Important to clearly define contract terms

• Service-level agreement: most important element of agreement
  – List of services vendor will meet
Licensing Applications

• Purchased software usually licensed
• Large selection of high-quality packaged software
• Two groups
  – Inexpensive office helpers
  – Large applications supporting whole organization
    • May cost millions
Software Licensing Benefits

• Licensing benefits
  – Immediate system availability
  – High quality
  – Low price
  – Available support

• **Beta version:** software to be tested by companies

• Up to 1 year of free service

• Installation specialists needed
Software Licensing Risks

• **Software licensing has risks**
  – Loose fit between needs and features
    • Comply with company needs
  – Difficulty in modifications
  – Bankruptcy of vendor
    • Maybe left without support and maintenance
  – High turnover of vendor personnel
    • Turnover among IS professionals is high
Steps in Licensing *Ready-Made Software*

- Selecting software involves large money investment
- **Project management team** responsibilities
  - **Identify problem or opportunity**
    - Define functional requirements
  - **Identifying potential vendors**
    - Trade shows
  - **Soliciting vendor information**
  - **Request for information (RFI)**: request informal information about product
Steps in Licensing Ready-Made Software

- Responsibilities of selecting vendor
  - Defining system requirements
  - Requesting vendor proposals
    - Request for proposal (RFP): specifies system requirements
  - Reviewing proposals and screening vendors
  - Visiting sites
  - Selecting the vendor

COTS; Commercial – Off – The - Self
Steps in Licensing Ready-Made Software

• Responsibilities while selecting vendor
  – Benchmarking
    • Benchmarking: quantify performance
  – Negotiating a contract
  – Implementing new system
  – Managing post-implementation support
Software As a Service

- **Application service provider (ASP):** organization that offers software on Web
- **Software as a service (SaaS):** applications through Web
- No software installed
- Files may be stored on local storage device
- ASP may rent software
Software As a Service

- Renting software has benefits and risks
  - No need to maintain
  - No large startup fee
  - Storage hardware unnecessary
  - Software available sooner
  - Good option for small companies
  - Software on demand approach
  - Lack of control may be issue
User Application Development

- **User application development**: nonprogrammer users write own applications
  - Simple and limited in scope
  - Develop small applications for immediate needs
  - Maintained by end users
Managing User-Developed Applications

- **Challenges of user-developed applications**
  - Managing reaction of IT professionals
  - Providing support
  - Compatibility
  - Managing access
Advantages and Risks

• **Advantages of user development**
  – Shortened lead times
  – Good fit to needs
  – Compliance with culture
  – Efficient utilization of resources
  – Acquisition of skills
  – Freeing up IS staff time
Advantages and Risks

• Disadvantages of user-developed applications
  – Poorly developed applications
  – Islands of information
  – Duplication
  – Security problems
  – Poor documentation
Then, What to select?

The Nature of Decision Making

- Making effective decisions, as well as recognizing when a bad decision has been made and quickly responding to mistakes, is a key ingredient in organizational effectiveness.

- Some experts believe that decision making is the most basic and fundamental of all managerial activities.

- Decision making is most closely linked with the Planning function.
What is Decision Making?

• **Decision making** is the act of choosing one alternative from among a set of alternatives.

• We have to **first** decide that a decision has to be made and **then secondly** identify a set of feasible alternatives before we select one.
Decision-Making Process

- Decision-Making Process includes:
  - recognizing and defining the nature of a decision situation
  - identifying alternatives
  - choosing the ‘best’ [most effective] alternative and
  - putting it into practice.
Sometimes effective decisions must be made to:

- **Optimize** some set of factors such as profits, sales, employee welfare and market share **or**
- **Minimize** loss, expenses or employee turnover **or**
- **Select best methods** for business, employees, or a strategic alliance.
Managers make decisions about both problems (undesirable situations) and opportunities (desirable situations).

- Cutting costs by 10%
- Learning that the company has earned higher-than-projected profits

It may take a long time before a manager can know for sure if the right decision was made.
Types of Decisions

• **Programmed decision** is one that is fairly structured or recurs with some frequency (or both).

• **Nonprogrammed decision** is one that is unstructured and occurs much less often than a programmed decision.
A View of Decision-Making Conditions

The decision maker faces conditions of:

- Certainty
- Risk
- Uncertainty

Level of ambiguity and chances of making a bad decision:

- Lower
- Moderate
- Higher
System Applications

• The process of IS management
• The system development life cycle (SDLC)
• Alternative approaches to system development
• In-house system development
• External acquisition, outsourcing, and end-user development
Application Life Cycle Considerations

Management
End-users
Operations and Administrations
Maintenance and Organization
Suppliers and Customers
The Context of **Project Management** — Project Attributes

- Time Frame
- Purpose (to provide value!)
- Ownership
- Resources (the triple constraint)
- Roles
  - Project Manager
  - Project Sponsor
  - SME (domain & technical)
- Risk & Assumptions
- Interdependent Tasks
- Planned Organizational Change
- Operate in Environments Larger than the Project Itself
The Triple Constraint

Figure 1.2
Why Do IT Projects Fail?

• Larger projects have the lowest success rate and appear to be more risky than medium and smaller projects
  – Technology, business models and markets change so rapidly that a project that takes more than a year can be obsolete before they are completed.
• The Chaos study also provides some insight as to the factors that influence project success.
The Project Life Cycle and IT Development

• Project Life Cycle (PLC)
  – A collection of logical stages or phases that maps the life of a project from its beginning to its end in order to define, build and deliver the product of the project – i.e., the information system

• Projects are divided into phases to increase manageability and reduce risk
  – Phase exits, stage gates, or kill points are decision points at the end of each phase to evaluate performance, correct problems or cancel the project
  – Fast tracking is the overlapping of phases to reduce the project’s schedule
    • Can be risky!
What Is an IT Application?

Definition

• **Single-user System/Personal System:** An IT system used by only one person. A system that stands alone and is not interconnected with other companies or shared by other people.

• **Enterprise System:** Usually involves the same types of components as a personal system, plus server or mainframe, database, and network. It is generally a shared system.
What Is an IT Application?

*Characteristics of Personal Systems*

- Designed for Hands-On Usage
  - *Hands-on System*: A system in which a user enters data and information, directs processing, and determines the types of output to be generated.

- Tailored to Personal Requirements and Preferences

- Used to Improve Personal Performance
What Is an IT Application?

*Characteristics of Enterprise Systems*

- Designed for Shared Use
- Designed for Sharing Data Resources
- Designed to Connect a Variety of Users
- Designed for Larger Size and Scope
What Is an IT Application?

Impact of IT Applications

• Improved Productivity
  – *Productivity*: The relationship between the results of an activity (output) and the resources used to create those results (inputs).
  – *Personal Productivity Software*: Software packages that permit activities to be completed more quickly, allow more activities to be completed in a particular period of time, or allow a task to be completed with fewer resources.
What Is an IT Application?

Impact of IT Applications (Continued)

• Greater Effectiveness
  – Effectiveness: The extent to which desirable results are achieved.

• Increased Creativity and Innovation
The Systems Development Life Cycle (SDLC)

- *Systems development life cycle (SDLC)* – the overall process for developing information systems from planning and analysis through implementation and maintenance
The Systems Development Life Cycle (SDLC)

1: Planning
   - Does system make sense? Feasibility. Scheduling.

2: Analysis
   - How can system solve business problem? LAYPERSON language.

3: Design
   - How can system solve business problem? TECHIE language.

4: Development
   - Build the system.

5: Testing
   - Test the system. 3 times the time and resources of programming!!

6: Implementation
   - Convert from old system to new system.

7: Maintenance
   - Fix, maintain, and improve system.
Relative Costs of Fixing Software Faults

Requirements | Specification | Planning | Design | Implementation | Integration | Maintenance

1 2 3 4

1 2 3 4 10 30 200
The Systems Development Life Cycle (SDLC)

1. **Planning phase** – involves establishing a high-level plan of the intended project and determining project goals>> *conceptualization*

2. **Analysis phase** – involves analyzing end-user business requirements and refining project goals into defined functions and operations of the intended system
   - **Business requirement** – detailed set of business requests that the system must meet in order to be successful
PLANNING: Assess Project Feasibility

• *Feasibility study* – determines if the proposed solution is feasible and achievable from a financial, technical, and organizational standpoint

• Different types of feasibility studies
  – Economic feasibility study
  – Operational feasibility study
  – Technical feasibility study
  – Schedule feasibility study
  – Legal and contractual feasibility study
Analysis: Examining Business Processes

• **Business process** - a standardized set of activities that accomplish a specific task, such as processing a customer’s order

• Business processes transform a set of inputs into a set of outputs (goods or services) for another person or process by using people and tools
The Systems Development Life Cycle (SDLC)

• *Continuous process improvement* - attempts to understand and measure the current process, and make performance improvements accordingly.

• *Business process reengineering (BPR)* – “Blows away” the current processes, enables new ones. More radical change.
Analysis: Examining Business Processes

- **Business process modeling** (or **mapping**) - the activity of creating a detailed flow chart or process map of a work process showing its inputs, tasks, and activities, in a structured sequence

- **Business process model** - a graphic description of a process, showing the sequence of process tasks, which is developed for a specific
  - **As-Is process model**
  - **To-Be process model**
Example: Ford - BEFORE and After

500 AP employees! Most time spent on mismatches.
Example: Ford - Before and AFTER

Purchasing → Purchase Order → Vendor → Goods → Payment

“Don’t send us invoices”

75% reduction in head count.
The Systems Development Life Cycle (SDLC)

3. *Design phase* – involves describing the desired features and operations of the system including screen layouts, business rules, process diagrams, pseudo code, and other documentation>

Architectural Design

4. *Development phase* – involves taking all of the detailed design documents from the design phase and transforming them into the actual system>

Development Evaluation
The Systems Development Life Cycle (SDLC)

5. **Testing phase** – involves bringing all the project pieces together into a special testing environment to test for errors, bugs, and interoperability and verify that the system meets all of the business requirements defined in the analysis phase.>> Testing and Evaluation

6. **Implementation phase** – involves placing the system into production so users can begin to perform actual business operations with the system.>> Deployment
Kinds of System Deployment

(a) Parallel
- Old System
- New System

Description: Old and new systems are used at the same time.

(b) Direct
- Old System
- New System

Description: Old system is discontinued on one day and the new is used on the next.

(c) Phased
- Old System
- New System

Description: Parts of the new system are implemented over time.

(d) Pilot (single location)
- Old System
- New System

Description: Entire system is used, is used in one location.
The Systems Development Life Cycle (SDLC)

7. **Maintenance phase** – involves performing changes, corrections, additions, and upgrades to ensure the system continues to meet the business goals

>> operations, Maintenance and Upgrade
Regardless of your major, you will be involved in Steps 1, 2, 5, 6 and 7

• Technology is a huge part of business and our world in general.
• You can’t “not like computers” anymore and survive.
• Minimally, you need to understand technology and its consequences for business and life.
• That’s true even if you outsource!
What’s “outsourcing”?

- Generally that means your organization is not able/willing to complete steps 3 and 4.
- You STILL will be involved in 1, 2, 5, 6 and 7.
- Even if you hire a consultancy you can’t escape completely!!
Sourcing’s New Surge - Offshoring

• **Offshore outsourcing** – using organizations from developing countries to write code and develop systems

• According to Forrester Research, nearly half of all businesses use offshore providers, and two-thirds plan to send work overseas in the near future
Examples

Customer Care
On-Line Book Seller
Stock Trading
Floral Delivery

Observations

Various Cases/Combinations
Tech. Development, Networked Computing Evolution
Legacy System; Constraints/Obstacles
Still, Not sufficient Functionality
Best Development Methodology
Software Engineering

The problem is *complexity*
Many sources, but *size* is a key:
Mozilla contains 3 Million lines of code
UNIX contains 4 million lines of code
Windows 2000 contains $10^8$ lines of code

Second is role and combinatorics of “state”

Third is uncertainty of “inputs” and their timing
Fourth is the continuing changing “environment” and demands.

Software engineering is about managing all the sources of complexity to produce effective software.
Software Engineering in a Nutshell

• Development of software systems whose size/complexity warrants team(s) of engineers
  – multi-person construction of multi-version software [Parnas 1987]

• Scope
  – study of software process, development/management principles, techniques, tools and notations

• Goal
  – production of quality software, delivered on time, within budget, satisfying customers’ requirements and users’ needs
What does a software engineer do?

Software engineers should

– adopt a systematic and organised approach to all aspects of software development.

– use appropriate tools and techniques depending on
  • the problem to be solved,
  • the development constraints and
  • the resources available

– Understand and communicate processes for improved software development within their organization

– Be effective team members and/or leaders.

– Can be very technical or more managerial depending on organizational need.
What is the difference between software engineering and computer science?

**Computer Science**
- theory
- fundamentals

**Software Engineering**
- the practicalities of developing
- delivering useful software

*Computer science theories* are currently insufficient to act as a complete underpinning for software engineering, BUT it is a foundation for practical aspects of software engineering.
What is the difference between software engineering and system engineering?

- **Software engineering** is part of System engineering
- **System engineering** is concerned with all aspects of computer-based systems development including
  - hardware,
  - software and
  - process engineering
- **System engineers** are involved in
  - system specification,
  - architectural design,
  - integration and deployment
Software Engineering ≠ Software Programming

• Software programming
  – Single developer
  – “Toy” applications
  – Short lifespan
  – Single or few stakeholders
    • Architect = Developer = Manager = Tester = Customer = User
  – One-of-a-kind systems
  – Built from scratch
  – Minimal maintenance
Software Engineering ≠ Software Programming

• Software engineering
  – Teams of developers with multiple roles
  – Complex systems
  – Indefinite lifespan
  – Numerous stakeholders
    • Architect ≠ Developer ≠ Manager ≠ Tester ≠ Customer ≠ User
  – System families
  – Reuse to amortize costs
  – Maintenance accounts for 60%-80% of overall development costs
Software Qualities

• Qualities are goals in the practice of software engineering, and directly relate to many of the guiding principles.

• External vs. Internal qualities

• Product vs. Process qualities
Software Qualities

• Critical Quality Attributes
  – Correctness
  – Maintainability
  – Dependability
  – Usability
  – Reliability

• Other Attributes
  – Completeness
  – Compatibility
  – Portability
  – Internationalization
  – Understandability
  – Scalability
  – Robustness
  – Testability
  – Reusability
  – Customizability
  – Efficiency
Review of Class
Applications and the Organizations

Network Computing Organizations

Central/Networked
Stove piped/Integrated structure

Acquisition Options
Make/Buy, and between

Application Life cycle

Alternative Development Methodology
Summary

• Several alternatives to having applications developed in-house
• Outsourcing can mean commissioning development or assigning services to vendor
• Outsourcing custom-designed applications gives good fit
• Outsourcing IT services benefits are great
Summary

• Outsourcing IT services has potential risks such as loss of control
• Licensing software advantages are immediately available and low priced
• Disadvantage of licensing software is loose fit to organization
• ASP is popular method of obtaining software
• Client pays monthly fees for ASP
Summary

• User application development advantages are short lead time and good fit
• User application development disadvantage is poor quality
• Over half of America’s office workers have rich computer resources
• Policies established to prevent computer abuse