The System Development Life Cycle

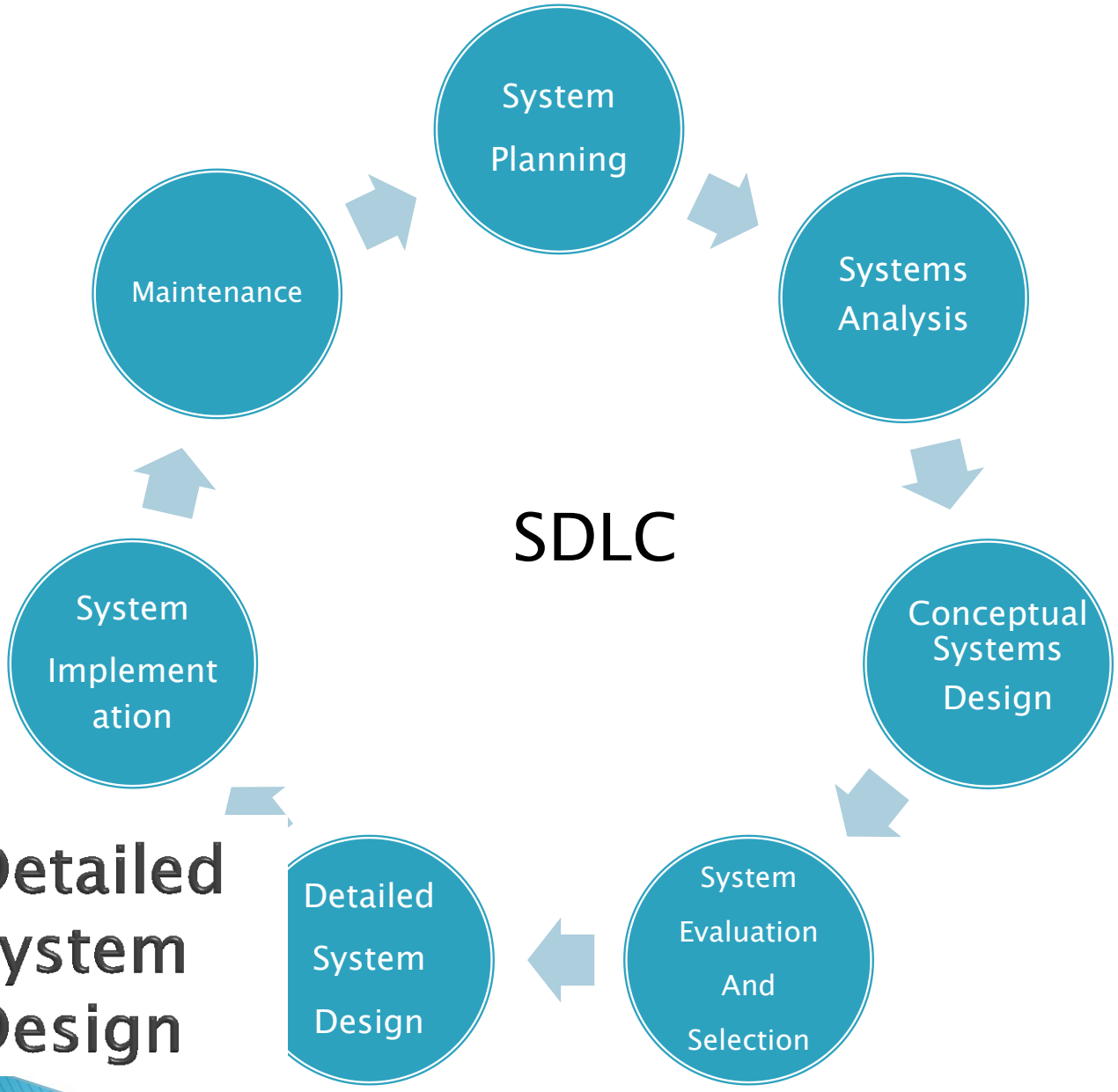
Detailed System Design

Part I Objectives

- Discuss the need for high-quality and usable information and identify the system design component that controls it the most.
- Explain important differences between reports
- Explain design considerations and trade-offs between user interface options
- Discuss the three-dimensions of *process design*

Part II Objectives

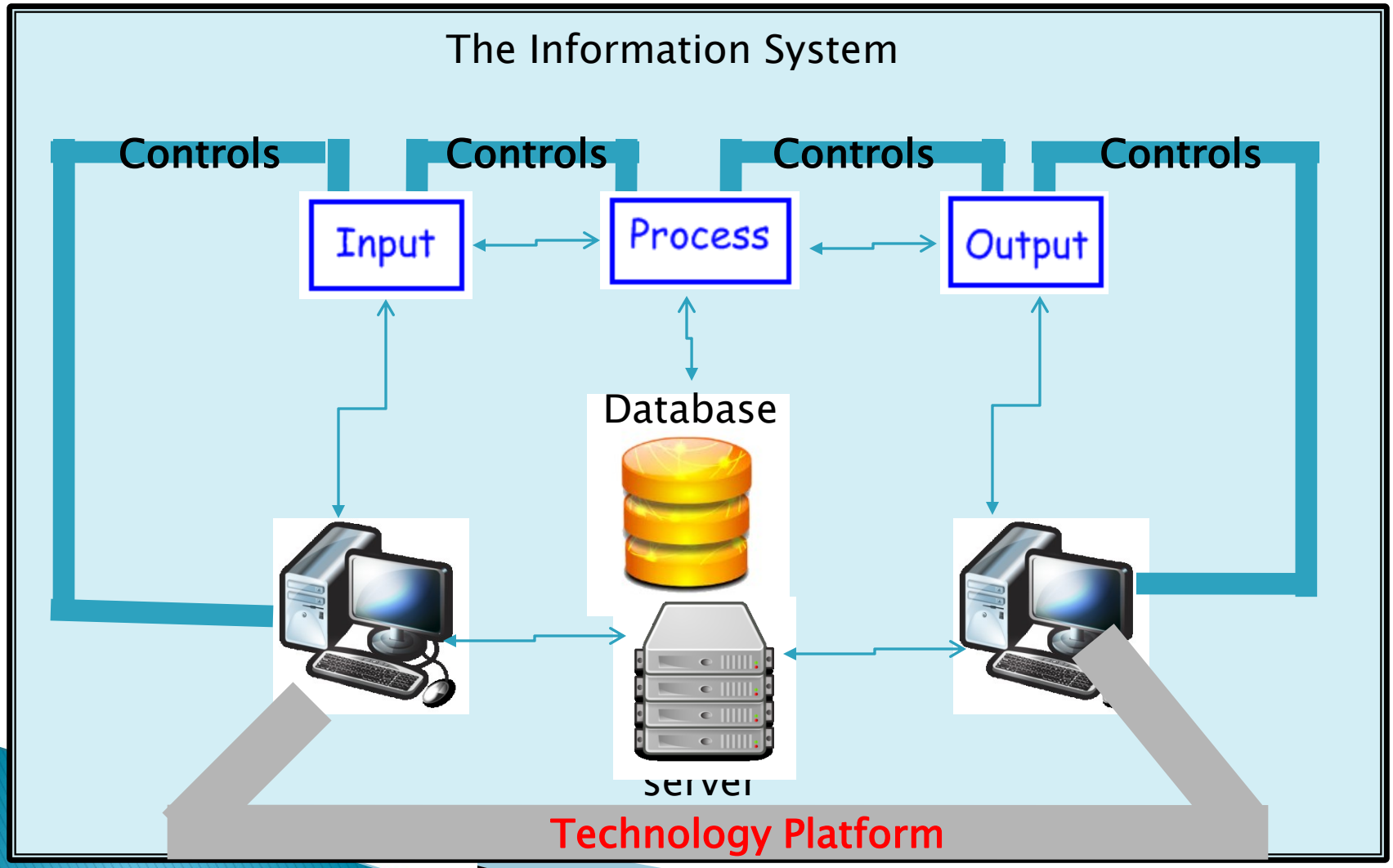
- Identify threats to information systems
- Explain the input-controls can filter out invalid data.
- Explain the use and importance of an audit trail
- Discuss destructive software and how to protect a system from having it enter the system
- Discuss real-time and batch processing controls
- Preview the Network Design topics that will be studied in the second year of IBCS
- Discuss Computer Architecture designs and how IT professionals evaluate and acquire computer architectures.



Detailed System Design

Phase 5: The Detailed Systems Design

The systems design components are interrelated to form an integrated information system.



Detailed System Design Components

- System Output Design
- System Input Design
- System Process Design
- System Database Design
- Designing System Controls
- System Network Design
- Designing the Computer Architecture

The Detailed System Design

Part I

- System Output Design
- System Input Design
- System Process Design
- System Database Design

System Output Design

Objectives

- Overview
- Providing information for making decisions
- Report Differentiation (for varying management levels)
- Reports that compare data
- Reports that monitor data variations
- High-quality display screens
- Graphs/Numeric Data representations
- File output

System Output Design

Overview

- Systems design should begin with output because output is the controlling system design component. All other components are design to produce useful output (or to support the processes that produce useful output)
- If the output of the system does not satisfy the users then the system can not be sucessful.

System Output Design

Providing information for making decisions

High-quality information possesses the following characteristics:

- Accessibility
- Timeliness
- Relevance
- Accuracy

High-quality information must be both substantive and in a form that appeals to the cognitive style of the user.

System Output Design

Report Differentiation (for varying management levels)

- Hierarchical reports are tailored to the appropriate user level
  - Filter reports screen out extraneous data that is too detailed for a particular user
  - Responsibility reports include (or limits) the information that pertains to tasks or decisions appropriate for particular user
- Data is not information

System Output Design

Reports that compare data

- Comparative reports present and examine sets of data side-by-side to establish similarities and differences.

- Types of comparative reports:
  Horizontal, vertical, counterbalance
  (Details will be provided as needed)

System Output Design

Reports that monitor data variations

Monitoring reports are for the systems management to keep track of processes.

A variance report shows the difference between actual and standard operation.

An exception report contains information for a process that as exceeded or fallen below established thresholds or limits.

System Output Design

High-quality display screens

- Users can easily locate and understand information presented on a screen.
- Minimizing eye movement assist scanning
- Eye movement should be consistent
- Color is useful for directing a users attention and to convey information

System Output Design

Graphs/Numeric Data representations

- Graphs are often a preferred way to summarize large quantities of data which can be analyzed to quickly see problems, conditions, trends, opportunities
- Tabular reports, matrices, charts can show breakdowns of data, collect / sort data by categories or conditions, perform calculations, and summarize data by displaying data in rows and columns.

System Input Design

Objectives

- Paper Forms
- Electronic Forms
  - Designing electronic forms
  - Data Entry (forms/ devices)
  - Codes (to represent data)
  - Menus/system navigation
- Human language interaction
  - Written
  - Typed
  - Scanned
  - Spoken

System Input Design

How to design paper forms and source-documents based on data-entry screens

- Select appropriate paper
- Sizing
- Manifolding
- Zoning
- Preparing instructions
- Captioning and describing data fields
- Spacing
- Sequencing

System Input Design

Advantages of electronic forms over paper forms

- Reduce data-entry costs (opposed to first using paper)
- Reduce errors – form validation can occur instantly
- Reduce time : for correcting errors, getting help, transmitting to authorized users workstation for processing, by performing calculations, automatically pulling data from other sources
- Reduce cost

System Input Design

What are direct-entry devices?

- Keyboard, mouse, voice recognition, stylist, touch screen
- Point-of-sale devices, card readers, Automatic Teller Machines (ATM)
- Scanner, digitizer, magnetic ink character recognition, optical character recognition
- USB devices
- Anything that a user directly interfaces with

System Input Design

Using coding for input design

- Codes are used to represent keys, identifiers, account numbers and abbreviations to make it easier for users to enter data or query databases.
- Code are used to condense data input, classify and identify data items, retrieve or select specific data items, trigger events or a course of action (such as –A for “Add”)
- Bar-codes, color-codes, zip codes – examples to code a large number of items with as little data as possible

System Input Design

Menu Interfaces

- Menus are designed to assist users interact with the system by displaying choices to the user they can easily select.
- Menu Guidelines: they should be properly titled and identified with meaningful captions and appropriate aligned and spaced.
- Common selection method: keying, pointing, touching, voice input
- Types of on-screen menus: pull-down, nested, shingled, tiled, icon, touch

System Input Design

Natural Language Interfaces

- Enable queries to be expressed in conventional language
- When the system detects and ambiguity or can not infer what the user intends, the system can enter a dialog with the user to resolve uncertainty.
- Natural language can be both text and spoken
- Examples: siri (cell phone), voice recognition automated phone systems; typing software

System Design : Human Interaction



What is *usability?*

- Usability is the ease of use and learn-ability of a human-made object.
- The object of use can be a software application or anything a human interacts with.
- The more efficient, more satisfying and more memorable a program is, the more usable it is.

System Design : Human Interaction



Identify a range of usability problems with commonly used digital devices.

- Consider digital cameras: They often have small buttons that users with physical disabilities have trouble using. Additionally, blind users cannot use the view finder.

System Design : Human Interaction



Identify methods that can be used to improve the accessibility of systems.

- *Accessibility* defines how the system can be accessed for information exchange and operation.
- This usually involves a keyboard, mouse and a screen. However, for some people using these methods to access the system is difficult or impossible.
- Ways to improve accessibility of a system:
  - Touch screens
  - Voice recognition
  - Text-to-speech
  - Braille keyboards
  - Braille printers



Identify a range of *usability* problems that occur in a system.

- Voice recognition systems often misinterpret accents/dialects and do not deal with ambient noise well.

System Design : Human Interaction



Discuss the moral, ethical, social, economic, and environmental implications of the interaction between humans and machines

- **Environmental**: Data centers use large amounts of electrical power. However, countries like Iceland have already invested in 100% renewable energy data centers. Computers have also reduced pollution costs of travel through introduction of telecommuting.
- **Moral**: Dragnet surveillance programs, such as NSA's PRISM and XKeyScore programs, arguably breach the right to privacy of internet users around the world.
- **Social**: Globalization of culture. Allows people from distance parts of the world to meet instantly. Easy, cheap entertainment for all.
- **Economic**: People may lose jobs. Typists are the canonical example.

System Process Design

Objectives

- Three process design dimensions:
  1. Time
  2. Technology
  3. Modeling tools

- Time Dimensions
  - Real-time process
  - Near-real-time processing
  - Batch processing

System Process Design

- Process design specifies when and how things should be done to support user requirements.

System Process Design

Time Dimensions

- Does the process require in interactive interface?
- Do transactions get processed as soon as they occur?
- Do transactions accumulate over some period of time and then get processed?
  - Real-time process
  - Near-real-time processing
  - Batch processing
- Real-time process may have different technology platform requirements than batch processing.

System Process Design

Real-Time Processing

- Hard- the application fails completely if the process does not always meet its time constraint.

  Example: flight control system
- Soft- system performance is degraded if the process does not always meet its time constraint, but requirements can still be met.

  Example: airline registration system

System Process Design

Characteristics of Real-Time Processing

- Process orientation – data is continually processed and is integral to the organization
- Online file availability – the database is always available to users for updating, querying, other interactions
- Short response times/short transactions
- Constant updates
- Data is organized for rapid access

System Process Design

What are the characteristics of batch processing?

- Interaction is NOT ongoing
- Example : A once-per-week payroll system)
- Data is stored as an input into a process, which is read as the batch program executes, then output is produced.
- Some systems could be considered interactive, but not real-time, and not batch, such as some searching/streaming

System Process Design

Modeling Tools

- State-transition diagrams (STD)
  - Model dynamic data flow, as transactions are defined and the systems moved through a finite number of different “states”. Example: ATM
- Data Flow Diagrams
  - DFD models data frozen in time (static)
- Process Specifications
  - DFD describe ***what*** is done
  - STD describe ***when*** it is done
  - Process specification identify ***how*** it is done

System Database Design

Objectives

- What is a database management system?
- What is a relational database system
- Designing a relational database system
- What is normalization
- Systems for Storing and Manipulating Data
- Flat-file systems vs Relational Database Systems
- Big Data
- NoSQL Databases



Database Design is the process of determining the content and arrangement of data needed to support system designs.

The first level of the database design occurs in the front-end, independent of any specific database design or database management system (DBMS).

The second level of the database design occurs by taking the general designs (such as an entity-relation diagram) and transform it into a detailed database design specific to a DBMS.

When is the database designed?

- After the design of output, input and processes
- The database design process uses the user requirements as input
- Having these prerequisite steps completed will provide confidence that the database design will be successful.

How is the database designed?

- Defining data structures
  - C Struct or database tables
  - Defining Fields and attributes
- Defining or Modeling relationships
- There are Three primary database models
  - A hierarchical model
  - A network model
  - A relational model
  - (The internal details for these database models will be covered next year)

System Database Design

Systems for Storing and Manipulating Data

- Flat-File (traditional file system approach)
- Database management system approach

System Database Design



Traditional (Flat-file) systems

- Example: The Online Grade Book Project
  - We rely in C programs that interface with the operating system in order to access files.
  - The operating system controls the files, out C programs make calls to create files, add data to the files, read files, open/close/ftell files.
  - The operating system does not manage the details of what is stored inside our files (the C struct details are unknown to the Linux environment)

System Database Design



What is a database management system?

- A system to separate applications from the internal storage details of a database.
- A collection of utility programs designed to manage data
- Software that enables users with different application needs to create, access, modify, and maintain data in a database.

System Database Design



What is a relational database system

- ‘. database system based on mathematical set theory.
- Tables are used to represent data as well as relationships between sets of data.
- Tables are defined in terms of rows and columns, where columns define one attribute, and one row represents a “record” such as a C structure.
- Structure Query Language (SQL) is the most common method for manipulating relational database. [See Lab]

System Database Design

Designing a relational database system

- Complete the system planning, analysis, general system design, evaluation and selection processes of the SDLC
- Complete output design, input design, and process design and transactions in detail.

System Database Design

Designing a relational database system

- Database Design Steps
  1. Model entities and map to tables
  2. Designate primary keys to each table
  3. Model relationships between tables
  4. Model attributes from the domain of data types for all tables
  5. Normalize the relational database model
  6. Prepare a data dictionary

System Database Design

What is normalization

- A process for optimizing the database design
- A process of preventing insert, delete, and update anomalies
- A process for providing database design stability over time.
- Details of first normal form (1NF), second normal form (2NF), and third normal form (3NF) will be covered next year.

System Database Design

Flat Files Advantages

- Simple data design
- Fast data access
- Inexpensive supporting technology
- Good choice when there is little data sharing
- Good when users are not cross-departmental

DBMS Advantages

- Improved data integration (relations)
- Increase data accessibility – using high level query languages
- Improve data integrity
- Simplifies development
- Added security
- Logical vs physical independence

System Database Design

Flat Files Disadvantages

- Lack of data relations
- Redundant data
- Lack of standards
- Development time for new applications

DBMS Disadvantages

- Complex data design
- Performance considerations
- May be expensive





System Database Design

- Big Data

System Database Design

- NoSQL Databases

Database Vocabulary

- Attribute
- Domain
- DBMS/RDBMS
- Entities
- Primary Key
- Foreign Key
- Relationship
- Normalization (first normal form, second normal form, third normal form)
- NULL,
- Relations, relationships
- Tables
- Traditional file system (flat file)
- Models (hierarchical, network, relational)

The Detailed System Design

Part II

- Designing System Controls
- System Network Design
- Designing the Computer Architecture

Designing System Controls

- Controlling the data integrity of data entering the system
- Guarding against unreliable software
- Controlling the data integrity of data stored in the system
- Controlling the data integrity of data leaving the system
- Protecting the database
- Protecting the hardware
- Controlling database access

Designing System Controls

Design and maintain a system of controls to protect information systems from threats such as:

- Corruption and loss of data
- Unauthorized access
- Destructive and fraudulent software
- Human errors
- Wiretapping/data interception
- Poorly designed technology environment (fire,theft,power..)

Designing System Controls



Three categories of system controls:

- Preventive – help to keep threats from happening
- Detective – help spot threats and reduce or eliminate exposure to the threats
- Corrective – help the system recover from the occurrence of a threat

Designing System Controls

	Preventive	Detective	Corrective
Input controls			
Process controls			
Output controls			
Database controls			
Technology platform controls			
Access control			

Designing System Controls

Factors controlling the integrity of system Input

- Purpose: screen out errors and prevent garbage-in, garbage-out
- Input controls include:
  - Coding (standard codes, check-digits)
  - Validation (three levels)
    - Field validation; record validation; file validation
  - Identification
  - Batch (control each transaction completes only once)
  - Audit trail

Designing System Controls

Guarding against unreliable software

- Application software faults – designing, implementing, testing stages (Phase 6)
- Destructive and Fraudulent software
  - Viruses (attach themselves to other programs)
  - Worms (stand alone, replicate over network)
  - Logic Bombs (intentional , non-intentional y2k)
  - Trojan horses (other stuff too)
  - Salami Techniques (Superman plot)
  - Malware/adware/spyware

Designing System Controls



Access control

- Based on what users know (passwords, security questions. policies)
- Based on what users have (smart cards)
- Based on what users are (human-fingerprints, retina signatures, hand geometry)

Designing System Controls

Factors controlling the integrity of system output

- Sensitivity levels (top secret, secret, restricted, public)
- Process design (real-time, batch)

Where is the output? Screens, paper, electronic?

Telecom control – encryption/single key/double key/spoofing/

File Controls – usb drives/cloud/CD/DVD or other removable storage devices

Designing System Controls

Factors controlling the integrity of system output – Batch Process Controls

- Control where documents, programs, files are loaded
- Controls at the processing stage
- Controls at the spooling stage
- Controls at the printing stage
- Controls at the output distribution stage
- During transportation (cc)

Designing System Controls

Controlling database access

- Concurrency control / lock/olgb
- Encryption techniques
- Backup and recovery
  - Offsite/onsite/televaulting

- SQL grants/users/views

Designing System Controls

Protecting the Hardware Technology

- Data center considerations
  - Location ; physical construction (DIGEX); filtration; AC;Backup water supply; fire suppression ;UPS
- PC Controls
  - Physical controls/database controls
  - Backup/recovery/administration
- Backup and recovery plans-a chapter of their own

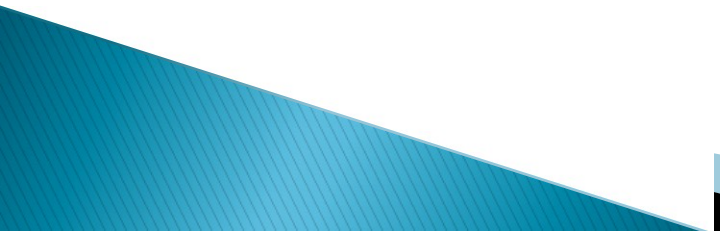
System Network Design

Objectives

- Network Design Steps
- Networking Elements
- Type of Transmission Cables
- Designing LANs
- Designing WANs/MANs
- Designing Wireless networks
- The OSI Model

Networking Topics

Will be covered next year; for now, we'll just add networking design as one of the elements of the detailed system design.



Designing the Computer Architecture

Checklist for Computer Architecture Design

1. Understand functions of mainframes, minicomputers, microcomputers, network/cloud computers
2. Configure computers, storage media, and other peripherals is a cooperative design
3. Prepare RFP to send to vendors
4. Evaluate proposals for cost-effective solutions
5. Consider outsourcing versus insourcing alternatives
6. Acquire selected compute architecture through lease or purchase.

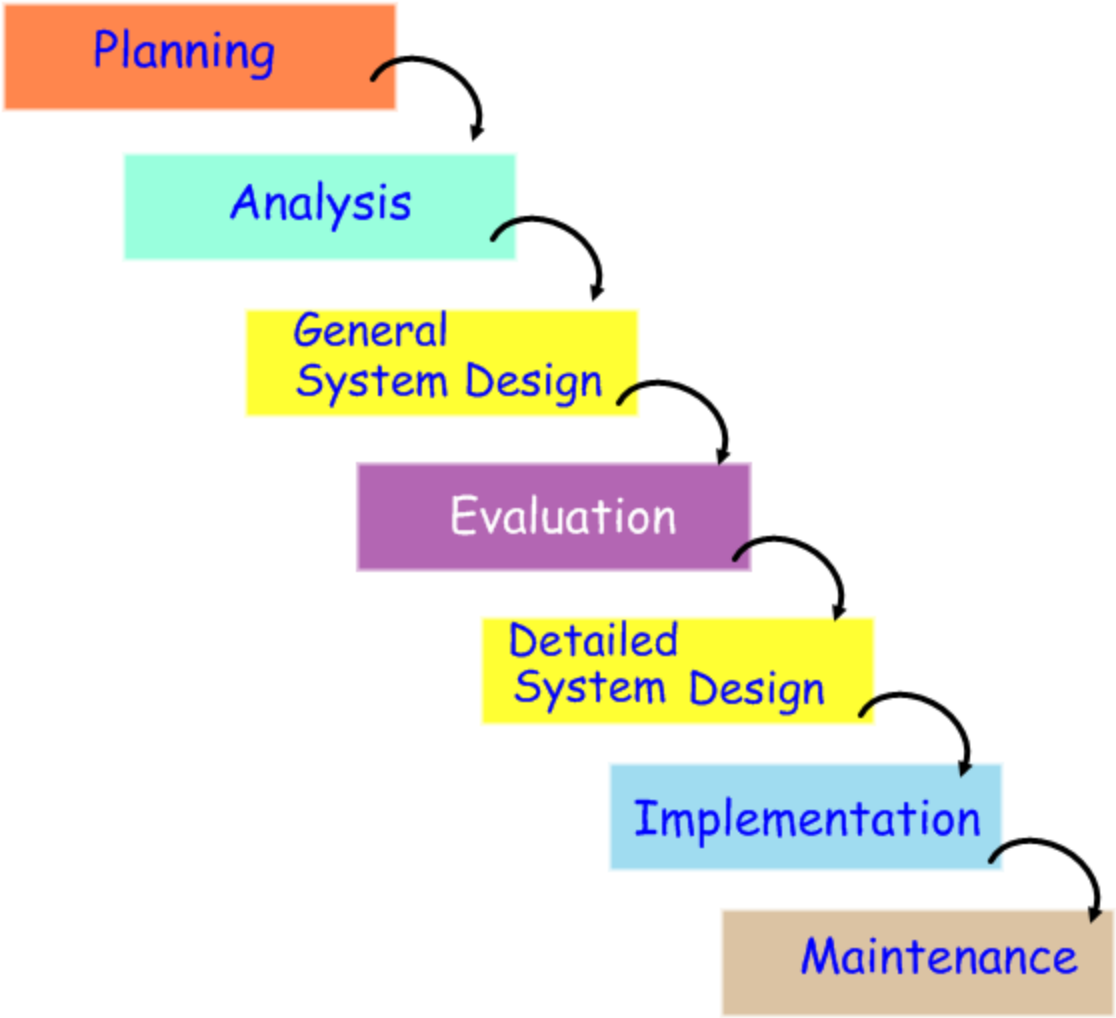
Designing the Computer Architecture

- Computing Components
  - Mainframes, minicomputers, microcomputers, personal computers, cloud computing resource
- Connecting/networking computing resources
- Acquiring Computer Architecture
  - RFPs
  - Vendor Selection and Evaluation
  - Service level agreements

Designing the Computer Architecture

- Acquiring Computer Architecture
  - RFPs
  - Vendor Selection and Evaluation
    - Vendor personnel
    - Customer interviews
    - Vendor documentation
    - Testing / validation procedures
    - Vendor finances
    - Vendor legal/business procedures
  - Service level agreements

The Waterfall Model



Key Terms

Terms to know, or be familiar with now				Terms for Next Year
Bottom-up design	Global-based Systems	Object	RAD	Encapsulation
Classes	Group-based systems	Object-class library	Specialists with Advanced Tool teams	Inheritance
Data Dictionary	Local-based systems	Object-oriented design	Structured-oriented design	Polymorphism
Data-oriented approach	Messages	Object-oriented programming	Top-down design	Subclass
Executive Information System (EIS)	Modularity	Process-oriented approach		Superclass