**Chap 1: : Foundation of system development**

1. **Information Systems Analysis and Design**

Defined as the complex, challenging, and simulating organizational process that a team of business and systems professionals uses to develop and maintain information systems

1. **Application Software**

Software designed to support organizational function or process

1. **Systems Analyst**

Organizational role most responsible for analysis and design of information systems

1. **Define information systems analysis and design**

* Get the system user involved
* Use a problem-solving approach
* Establish phases and activities
* Document through development
* Establish standards
* Manage process and the projects
* Justify systems as capital investments
* Don’t be afraid to cancel or revise scope
* Divide and conquer

1. **Describe the information systems development life cycle (SDLC)**

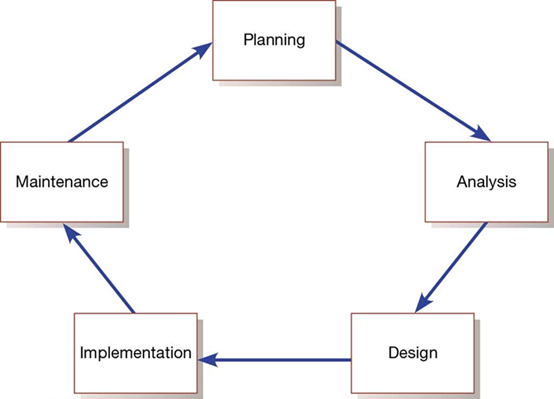
* **Systems development methodology**

A standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems

* **The systems development life cycle (S D L C)**
* The traditional methodology used to develop, maintain, and replace information systems

1. Describe the information systems development life cycle (S D L C)

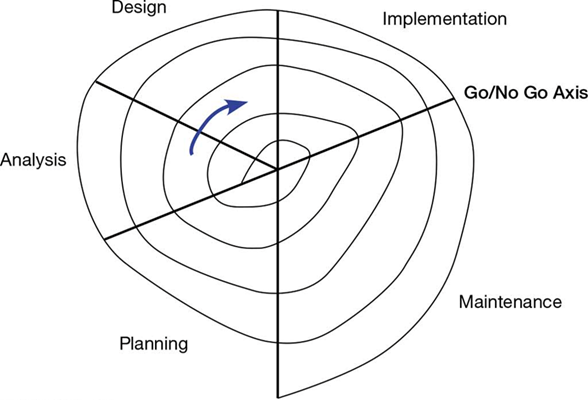
* A circular process, with the end of the useful life leading to the start of another
* At any given phase the project can return to a previous phase when needed
* Can be an iterative process



1. Evolutionary Model

* Describe the information systems development life cycle (SDLC)

A spiral process in which one is constantly cycling through phases at different levels

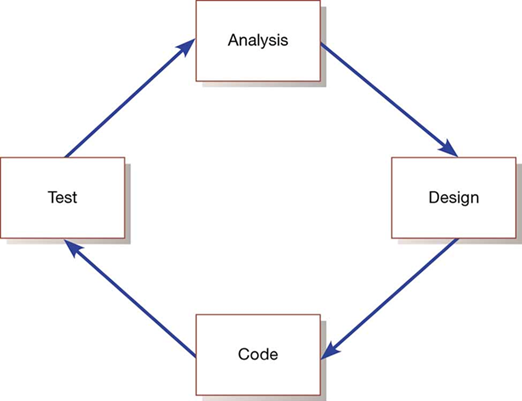


1. **Phrase of the SDLC**

* **Planning:**
* Need for a new or enhanced system is identified
* Needs are identified, analyzed, prioritized, and arranged
* Determine the scope of the proposed system
* Baseline project plan is developed
* **Analysis**
* System requirements are studied from user input and structured
* Requires careful study of current systems, manual and computerized, that might be replaced or be enhanced
* Output is description of the alternate solution recommend by the analysis team
* **Design**: Analyst converts the alternate solution into logical and physical specifications
* **Logical Design**: The design process part that is independent of any specific hardware or software platform
* **Physical Design:** The logical specifications of the system from logical design are transformed into technology-specific details from which all programing/system construction can be accomplished
* **Implementation**
* Occurs when the information system is coded, tested, installed, and supported in the organization
* New systems become part of the daily activities of the organization
* **Maintenance**
* The phase in which an information system is systematically repaired and improved
* Organization’s needs may change over time requiring changes to the system based on user’s needs

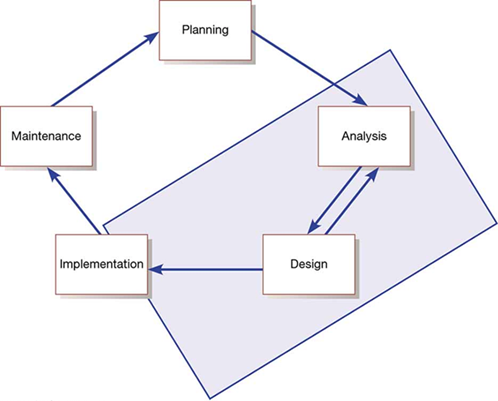
| **Phase** | **Products, Outputs, or Deliverables** |
| --- | --- |
| Planning | •Priorities for system and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems  •Detailed steps, or work plan, for project  •Specification of system scope and planning and high-level system requirements or features  •Assignment of team members and other resources  •System justification or business case |
| Analysis | •Description of current system and where problems or opportunities exist, with a general recommendation on how to fix, enhance, or replace current system  •Explanation of alternative systems and justification for chosen alternative |
| Design | •Functional, detailed specifications of system elements (data, processes, inputs, and outputs)  •Technical, detailed specifications of all system elements (programs, files, network, system software, etc.)  •Acquisition plan for new technology |
| Implementation | •Code, documentation, training procedures, and support capabilities |
| Maintenance | •New versions or releases of software with associated updates to documentation, training, and support |

1. **Analysis-Design-Code-Test Loop**

****

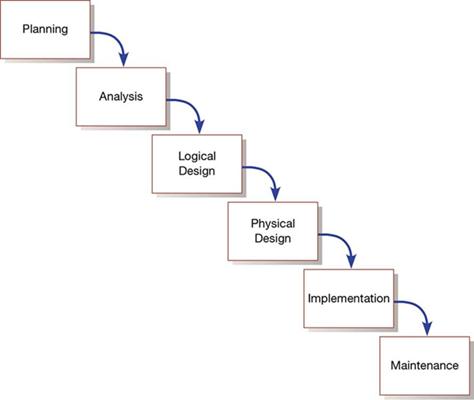
1. **Heart of Systems Development**

Current practice combines analysis, design, and implementation into a single process



1. **The SDLC Traditional Waterfall Problems**

* Once one phase ends another begins, going downhill until complete
* Makes it difficult to go back
* Results in great expense to make changes
* Role of system users or customers narrowly defined
* Focused on deadlines

****

1. **Agile methodologies share three key principles:**

* A focus on adaptive rather than predictive methodologies
* A focus on people rather than roles
* A focus on self-adaptive processes

1. **The Agile Manifesto**

The agile methodologies group argues that software development methodologies adapted from engineering generally do not fit with real world software development

**The Manifesto for Agile Software Development**

* Seventeen anarchists agree
* We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan
* **Principles Agile Manifesto**
* The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
* Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
* Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
* Businesspeople and developers work together daily throughout the project.
* Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
* The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
* Working software is the primary measure of progress.

1. **Describe the agile methodologies**

* Agile methodologies are not for everyone
* Fowler recommends an agile process if your project involves
* unpredictable or dynamic requirements
* responsible and motivated developers
* customers who understand the process and will get involved

1. **Five Critical Factors that Distinguish Agile and Traditional Approaches to System Development**

| **Factor** | **Agile Methods** | **Traditional Methods** |
| --- | --- | --- |
| **Size** | Well matched to small products and teams Reliance on tacit knowledge limits scalability | Methods evolved to handle large products and teams Hard to tailor down to small products |
| **Criticality** | Untested on safety-critical products  Potential difficulties with simple design and lack of documentation | Methods evolved to handle highly critical products Hard to tailor down to products that are not critical. |
| **Dynamism** | Simple design and continuous refactoring  are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments | Detailed plans and Big Design Up Front, excellent for highly stable environment but a source of expensive rework for highly dynamic environments |
| **Personnel** | Requires continuous presence of a critical mass of scarce experts  Risky to use non-agile people | Needs a critical mass of scarce experts during project definition but can work with fewer later in the project, unless the environment is highly dynamic |
| **Culture** | Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom (thriving on chaos) | Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear practices and procedures (thriving on order) |

1. **Describe eXtreme Programming**

* Short, incremental development cycles
* Focus on automated tests written by programmers
* Emphasis on two-person programming teams
* Customers to monitor the development process
* Relevant parts of eXtreme Programming that relate to design specifications are
* How planning, analysis, design, and construction are all fused into a single phase of activity
* Its unique way of capturing and presenting system requirement and design specifications

1. **Scrum** designed for speed and multiple functional product releases

Primary unit is the Sprint (runs two weeks to a month)

1. **Object:** A structure that encapsulates attributes and methods that operate on those attributes
2. **Inheritance:** Hierarchical arrangement of classes enabling subclasses to inherit properties of superclasses
3. **Object Class:** Logical grouping of objects that have the same attributes and behaviors
4. **Rational Unified Process (RUP)** is an object-oriented systems development methodology
5. **RUPs four phases** (each can be further divided)

* Inception
* Elaboration
* Construction
* Transition

1. **Criticisms of the SDLC include**

Forced timed phases on intangible and dynamic processes were doomed to fail

Life-cycle reliance has resulted in massive amounts of process and documentation

Cycles are not necessarily waterfalls

**Quiz:**

* **Systems Development Lifecycle:** The Systems development lifecycle (SDLC) is the process of developing software or information systems from start to finish.
* **Is the system realistic, and de we have the expertise to develop it?** Technical Feasibility
* **Do benefits exceed costs?** Economic Feasibility
* **Does the system meet all regulations and laws?** Legal Feasibility
* **Will our organization be able to operate the system?** Operational Feasibility
* **Is the project timeline realistic given our resources?** Scheduling Feasibility
* **do individual components of the code work properly?** System Testing
* **Information System:** collects, stores, and processes data to provide useful, accurate, and timely information, typically within the context of an organization.
* **Strategic planning**: emphasis on long-range and future goals.
* **Tactical planning:** a process in which managers set incremental goals that can be achieved in a year or less.
* **Operational planning**: scheduling employees, ordering supplies, and other activities that make day-to-day operations run smoothly.
* Structured problem: an everyday, run-of-the-mill, routine problem.
* Semi-structured problem: has a known procedure for arriving at a solution; however, the process might involve some degree of subjective judgment.
* Unstructured problem: requires human intuition as the basis for finding a solution.
* Transaction: an exchange between two parties that is recorded and stored in a computer system.
* Transaction Processing System (TPS): provides a way to collect, process, store, display, modify, or cancel transactions.
* Management information system (MIS): refer to a type of information system that uses the data collected by a transaction processing system, and manipulates that data to create reports used by managers to make routine business decisions in response to structured problems.
* Structured methodology: focuses on the processes that take place within an information system.
* Information engineering methodology: focuses on the data an information system collects before working out ways to process that data.
* Object-oriented methodology: treats an information system as a collection of objects that interact to accomplish tasks.
* Project management software: an effective tool for planning and scheduling.
* Analysis phase: the goal is to produce a list of requirements for a new or revised information system.
* System requirements: the criteria for successfully solving problems identified in an information system.
* Success factors: serve as an evaluation checklist at the end of the development project.
* System Requirements Report: describes the objectives for an information system.
* Design phase: the project team must figure out how the new system will fulfill the requirements specified in the System Requirements Report.
* Centralized processing: data is processed on a centrally located computer.
* Distributed processing: processing tasks are distributed to servers and workstations.
* What is the most popular agile method? XP extreme programming
* **Rolls in scrum organisation:**

Product owner.

Scrum Master.

Scrum team or teams.

* **What are 6 common agile methodologies**

1) Extreme Programming (XP)

2) Scrum

3) Crystal

4) Dynamic Systems Development Method (DSDM)

5) Lean Software Development (Kanban)

6) Feature-Driven Development (FDD)

* **What are the four core values of the agile manifesto**

1) Individuals and interactions over processes and tools.

2) Working software over comprehensive documentation.

3) Customer collaboration over contract negotiation.

4) Responding to change over following a plan.

* **What are 12 principles of the agile manifesto**

1) Satisfying 'customers' through early and continuous delivery of valuable work.

2) Breaking big work down into smaller components that can be completed quickly.

3) Recognizing that the best work emerges from self-organizing teams.

4) Providing motivated individuals with the environment and support they need and trust them to get the job done.

5) Creating processes that promote sustainable efforts.

6) Maintaining a constant pace for completed work.

7) Welcoming changing requirements, even late in a project.

8) Assembling the project team and business owners on a daily basis throughout the project.

9) At regular intervals, having the team reflect upon how to become more effective, then tuning and adjusting behavior accordingly.

10) Measuring progress by the amount of completed work.

11) Continually seeking excellence.

12) Harnessing change for competitive advantage.

* **Name 5 Scrum Events (Ceremonies)**

1) Sprint

2) Sprint Planning

3) Daily Scrum (Daily Standup)

4) Sprint Review (Sprint Demo)

5) Sprint Retrospective

* **Name 3 Scrum Artifacts**

1) Product Backlog

2) Sprint Backlog

3) Increment

* **Name 3 parties involved in scrum**

1) The product owner

2) The scrum master

3) The scrum team

* **Agile software engineering:** represents a reasonable compromise between the conventional software engineering for certain classes of software and certain types of software projects
* Agile development processes can deliver successful systems quickly. **True**
* agile development stresses continuous communication and collaboration among developers and customers. **True**
* **Describe the five philosophies embraced by Agile software engineering**

1. Encourages customer satisfaction

2. Incremental software delivery

3. Small project teams (composed of software engineers and stakeholders)

4. Informal methods

5. Minimal software engineering work products

* **What does the Manifesto for Agile Software Development propose?**

**it may be better to value:**

- Individuals and interactions over processes and tools

- Working software over comprehensive documentation

- Customer collaboration over contract negotiation

- Responding to change over following a plan

**- What three assumptions are Agile Processes based on?**

1. It is difficult to predict in advance which requirements or customer priorities will change and which will not

2. For many types of software design and construction activities are interleaved

3. Analysis, design, and testing are not as predictable from a planning perspective as one might like them to be

* Agile processes must be adapted incrementally to manage unpredictability. **True**
* **Quality Constraints:** Time/Cost/Scope. One major consideration throughout the SDLC is the project quality constraint triangle shown below. These three variables are interdependent. You cannot change one without changing the others.

Project management during the SDLC is the science of making intelligent trade-offs among time, cost, and scope.

* List 10 Agility Principles

1. Highest priority is to satisfy customer

2. Welcome changing requirements

3. Delivering working software frequently

4. Stakeholders must work together daily during the project

5. Build projects around motivated individuals

6. Face-to-face communication is the most effective method

7. Working software is the primary measure of progress

8. Support sustainable development

9. Continuous attention to technical excellence

10. Simplicity by maximizing is essential

* **Traits that need to exist in members of agile development teams:**

Competence

Common focus

Collaboration

Decision-making ability

Fuzzy-problem solving ability

Mutual trust and respect

Self-organization

* **Agile Process Models**

1. Extreme Programming (XP)

2. Adaptive Software Development (ASD)

3. Dynamic Systems Development Method (DSDM)

4. Scrum

5. Crystal

6. Feature Driven Development (FDD)

7. Agile Modeling (AM)

* **What are the key activities of Extreme Programming?**

Planning

Designing

Coding

Testing

**Chap 2: Software development life cycle**

1. **RUP Methodologies**

**Rational Unified Process:** A software engineering process based on best practices in modern software development.

* A **disciplined approach** to **assigning and managing tasks** and responsibilities in a development organization
* Focus on high quality software that **meets the needs of its end users within a predictable schedule and budget**

A process framework that can be tailored to specific organization or project needs.

RUP is a methodology for delivering projects in a maximum performance manner.

**Perspectives:**

* **Dynamic perspective** - Shows the phases of the model over time
* **Static perspective** - Shows the development process activities that take place (workflows). There are six core workflows.
* **Practice perspective** - Suggests good practices to be used during the process

**4 phases:**

* **Inception -** Define scope of project
* **Elaboration -** Plan project, specify features, baseline architecture
* **Construction -** Build product
* **Transition -** Transition product to end user community

RUP is described at two levels: **the discipline level and the workflow detail level using Unified Modeling Language (UML) activity diagrams.**

**A Workflow** is a grouping of activities that are often performed "together" to produce a specific result. In particular, workflow details describe groups of activities performed together in a discipline.

**6 core workflows:**

* **Business modelling -** B processes are modelled using B use-cases
* **Requirements -** Actors who interact with the system are identified. Use cases are developed to model B requirements
* **Analysis and design -** design model is created and implemented using architectural models, component models, object models and sequence models
* **Implementation -** Components are implemented into sub-systems. Automatic code generation from design models helps speed this up.
* **Testing -** Iterative process that is carried out in conjunction with implementation. Testing follows the completion of the implementation.
* **Deployment -** Product is released and distributed.

**3 supporting workflows:**

* **Configuration and change management**
* **Project management**
* **Environmental (tools)**

**RUP good practices:**

* **Iterative development:** Plan increments based on customer priorities and deliver highest priority increments first
* **Manage requirements:** Document customer requirements and track its changes
* **Use component-based architectures:** Organize system architecture as reusable components (OOP)
* **Visually model software:** Use graphical UML (unified modelling language) models of the software
* **Verify software quality:** Enforce development quality standards
* **Control changes to software:** Team communication, Manage changes using a change management system

1. **Agile Methodologies**

**Agile** is a time boxed, iterative approach to software delivery that builds

software incrementally from the start of the project, instead of trying to delivery it all at once near the end.

It works by breaking projects down into little bits of user functionality called **user stories**, prioritizing them, and then continuously delivering them in short two week cycles called **sprints.**

**Agile Methodologies:**

aims for customer satisfaction through early and continuous delivery of useful software components developed by an iterative process with a design point that uses the bare minimum requirements (RAD, XP, RUP, & Scrum)

**Which methodology is fast and efficient, small and nimble, lower cost, fewer features, and shorter projects?:**

Agile Methodology

**What are the primary forms of agile methodology?**

RAD, XP, RUP, & Scrum

**Examples of Agile Approaches**

Extreme Programming (XP).

Scrum.

Lean Development (LD).

Agile Unified Process (AUP).

Adaptive Software Development (ASD).

Crystal family if methodologies.

Dynamic Systems Development Method (DSDM).

Feature-Driven Development (FDD).

Internet-Speed Development (ISD).

Pragmatic Programming (PP).

Test Driven Development (TDD).

XBreed.

**What is the most popular agile method?**

XP extreme programming

**What are some possible limitations with agile?**

Limited support for:

Projects with distributed development teams and resources .

Emphasis on co-location and face-to-face communication doesn't fit well with distributed projects.

Outsourcing.

Projects involving large teams.

Building or using reusable artifacts.

Development of large software systems.

Development of safety-critical software systems.

**Principles:**

* **Experiment & Learn Rapidly:** Is a guiding principle of Modern Agile because it protects us from wasting time and helps us discover success faster
* **Make Safety a Prerequisite:** Means establishing safety before engaging in potentially hazardous work
* **“Make People Awesome” :** If you make customers awesome, they tend to be natural promoters of your products or services
* **Deliver Value Continuously:** Anything valuable that hasn’t been delivered isn’t helping anyone. How might we deliver the right outcomes faster

**What are the characteristics of Agile Methods?**

Modularity

Iterative

time-Bound

Incremental

Convergent

People-oriented

Collaborative

**Extreme Programming (XP)**

A pragmatic approach to program development that emphasizes business results first an takes an incremental, get-something-started approach to building the product, using continual testing and revision.

**Scrum**

Involves implementing a small number of customers requirements in sprint cycles

**RUP vs Agile**

|  | **RUP** | **AGILE** |
| --- | --- | --- |
|  | determined a project life-cycle consisting of four phases. These phases can break down into small iterations if needed | So long as there are features to build, these activities continue for the duration of the project |
| **Main characteristics** | **Heavyweight**, Is a **Framework**  Use-case driven from inception to deployment, small team size  Architecture-centric, model visually, well-documented, lots of artifact  Iterative and incremental, where large projects are divided into smaller projects  Long-term detailed planning | **Lightweight, Like a Philosophy**  Small cross-functional teams are used  Daily status meetings are held, less artifact can be better  Short timeframe increments for each  change, short-term planning  A working project is completed at  end of each iteration and  demonstrated to stakeholders |
| **Main problems** | * It mostly **relies on the ability of experts** and professionals to assign the activities to individuals * It is a rather complex method which **makes its implementation challenging,** particularly for smaller businesses, teams or projects * A **long time is spent doing requirements** or design work before programming starts * Too **many roles, artifacts** required * **High cost** in managing, following RUP project * Tool-set are useful but **expensive** | * It can be difficult to **keep the interest of customers / users** who are involved in the process. * Team members may be unsuited to the **intense involvement** that characterizes agile methods. * **Prioritizing** changes can be difficult where there are **multiple stakeholders.** * Maintaining **simplicity requires extra work.** * **Contracts** may be a problem as with other approaches to iterative development. * Because of their focus on small, tightly-integrated teams, there are problems in **scaling agile methods to large systems.** * **Less emphasis on documentation** - harder to maintain when you get a new team for maintenance |

**Which of the following is one of three key principles shared by the Agile Methodologies?**

A focus on self-adaptive processes

**Which of the following is NOT valued according to the Agile Manifesto for software development?**

Prioritizing the plan over the change required

**Which of the below is NOT considered as the principle as per the agile manifesto?**

Businesspeople and developers work separately throughout the project.

**There are five factors that separate agile from traditional approaches to systems development. The factor "Criticality" describes which of the following agile methods factors?**

Untested on safety-critical products

**According to Fowler, which of the following is NOT considered a valid individual methodology that comes under the umbrella of Agile Methodologies?**

Passive Software development

**Which of the following is the best known Agile development methodology?**

eXtreme Programming

**eXtreme Programming is not an instance of one of the Agile Methodologies.**

FALSE

**The Agile Methodologies focus on the roles that people perform rather than placing the emphasis on individuals.**

FALSE

**The critical factor "Dynamism" refers to which of the following agile methods critical factors?**

Simple design and continuous refactoring are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments.

**A logical grouping of objects that have the same attributes and behaviors is known as a(n) \_\_\_\_\_\_\_\_.**

Object class

**\_\_\_\_\_\_\_\_ is/are the property that occurs when entity types or object classes are arranged in a hierarchy and each entity type or object class assumes the attributes and methods of its ancestors, that is, those higher up in the hierarchy.**

Inheritance

**\_\_\_\_\_\_\_\_ is/are often called the third approach to systems development, after the process oriented and data-oriented approaches.**

Object-oriented analysis and design

**One of the most popular realizations of the iterative approach for object-oriented development is the \_\_\_\_\_\_\_\_.**

RUP

**Which of the following is NOT a phase in the Rational Unified Process?**

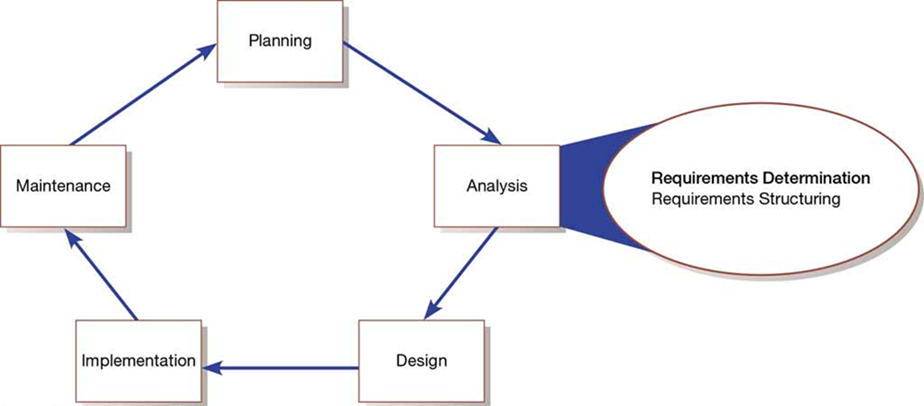
Calculation

**In the inception phase, which of the following is NOT performed by the analysts?**

Implement the project

**Chap 3: Determining software requirements**

1. **Analysis - 2 subphases:** Requirements determination & Requirements structuring
2. **System Development Life Cycle with Analysis Phase Highlighted**



1. **The Process of Determining Requirements**

* Impertinence – question everything
* Impartiality – consider all issues to find the best solution
* Relax constraints – assume anything is possible and eliminate the infeasible
* Attention to detail – every fact must fit with every other fact
* Reframing – challenge yourself to look at the organization in new ways

1. **Organizational Components to Understand**

* ***Business objectives*** that drive what and how work is done
* ***Information*** people need to do their jobs
* ***The data (definition, volume, size)*** handled in support of jobs
* ***Data transformation and storage*** (when, how, by whom)
* ***Data handling dependencies and sequences***
* ***Data handling and processing rules***
* ***Policies and guidelines*** that describe the nature of the business and market and the environment it operates in
* ***Key events*** that affect data values and when they occur

1. **Deliverables for Requirements Determination**

* ***Information collected from conversations with or observations of users:***
* interview transcripts
* notes from observation
* meeting minutes
* ***Existing written information:***
* business mission and strategy statements
* sample business forms and reports and computer displays
* procedure manuals, job descriptions
* training manuals
* flowcharts and documentation of existing systems
* consultant reports
* ***Computer-based information:***
* results from JAD sessions
* reports of existing systems, and displays and reports from system prototypes

1. **Traditional Methods of Collecting System Requirements**

* ***Individually interview people*** informed about the operation and issues of the current system and future systems needs
* ***Interview groups of people*** with diverse needs to find synergies and contrasts among system requirements
* ***Observe workers at selected times*** to see how data are handled and what information people need to do their jobs
* ***Study business documents*** to discover reported issues, policies, rules, and directions as well as concrete examples of the use of data and information in the organization

1. **Guidelines for Effective Interviewing**

* Plan the Interview
* Prepare interviewee: appointment, priming questions
* Prepare checklist, agenda, and questions
* Listen carefully and take notes (record if permitted)
* Review notes within 48 hours of interview
* Be neutral
* Seek diverse views

1. **Interviewing and Listening**

* ***Open-ended questions*** – questions in interviews that have no prespecified answers
* ***Closed-ended questions*** – questions in interviews that ask those responding to choose from among a set of specified responses

1. **Interviewing Guidelines**

* Don’t phrase a question in a way that implies a right or wrong answer
* Listen carefully to what is being said
* Record notes within 48 hours after an interview
* Don’t set expectations about the new system unless you know these will be deliverables
* Seek a variety of perspectives from the interviews

1. **Interviewing Groups**

* Drawbacks to interviewing individuals:
* Reconciling contradictions in information collected
* New interviews may require new questions
* Not an efficient process
* Group interview advantages:
* More effective use of time
* Allows synergy when groups can hear each other
* Primary disadvantage is difficulty in scheduling with multiple people involved

1. **Nominal group technique (N G T)**

facilitated process that supports idea generation by groups. At the beginning of the process, group members work alone to generate ideas. The ideas are then pooled under the guidance of a trained facilitator.

* End result is a listing of either problems or features generated and prioritized by the group
* Can be used as part of a J A D effort

1. **Direct observation of workers:**

* ***Watching users work*** at their jobs
* Observe actual measure of how ***employees interact*** with information systems and how they do their jobs
* ***More accurate*** than interview
* People can ***change their normal behavior*** when they know they are being observed
* Observation cannot be continuous, thus you are getting only a ***snapshot of how they work***

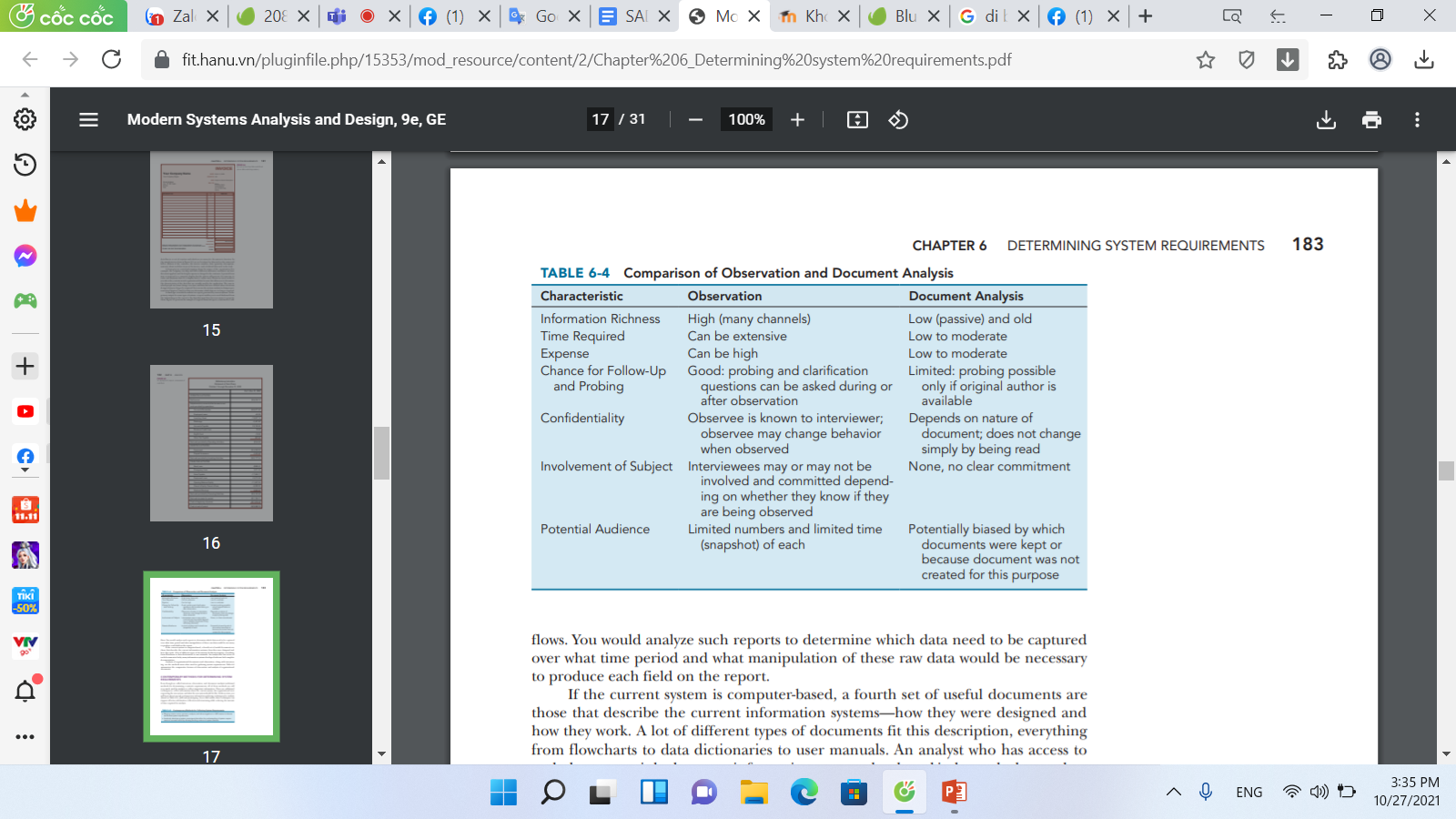
1. **Analyzing Procedures and Other Documents**

* An analysis of existing documents can give you a ***wealth of information***:
* ***Problems with existing systems***
* Opportunities to ***meet new needs*** with critical information
* ***Identify key people*** of current system
* ***Values of organization*** who help determine priorities desired by different users
* ***Special information processing circumstances*** that might not otherwise be identified
* ***Identify left out features of current software*** that may lead to needed features in future systems
* ***Identify processing rules*** that must be enforced

1. ***A written work procedure*** describes how a job or task is performed
2. ***Formal system*** – official way a system works as described in organizational documentation.
3. ***Informal system*** – way a system actually works
4. **Four major documents analyzed when creating a new system:**

* ***Written work procedure***
* ***A form such as the invoice form*** on the previous slide. Gives crucial information about the nature of the organization
* ***A report*** such as the one on the next slide. Can be used to analyze to determine which data to capture
* ***Documents used to describe the system*** and how it is used. Examples include flowcharts, data dictionaries, user manuals

1. **Comparison of Observation and Document Analysis**



1. **Contemporary Methods for Collecting System Requirements**

* Bringing session ***users, sponsors, analysts, and others together*** in a ***JAD session*** to ***discuss*** and ***review*** system requirements
* Iteratively developing ***system prototypes*** that refine the understanding of system requirements in concrete terms by ***showing working versions*** of system features

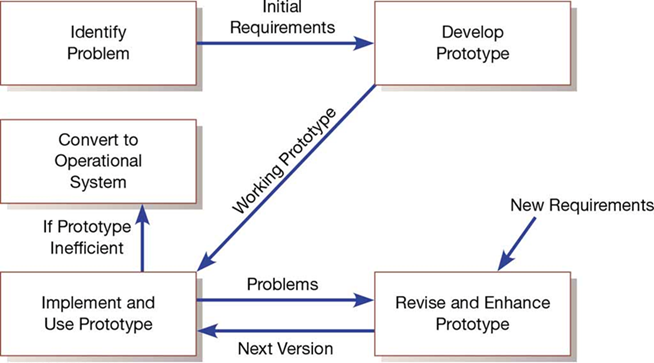
1. **J A D session leader** – trained individual who plans and leads Joint Application Design sessions
2. **Scribe** – person who makes detailed notes of the happenings at a Joint Application Design session
3. **End results of a J A D:**

* Documentation detailing existing system
* Features of proposed system

1. **Prototyping** – iterative process of systems development in which requirements are converted to a working system that is continually revised through close collaboration between an analyst and users

* Quickly converts basic requirements into working, limited version of final information system
* Viewed and tested by the user
* Prompts user for modifications for final system

1. **The Prototyping Methodology**



1. **Evolutionary Prototyping**

* Begin by modeling part of the target system
* If successful, evolve rest of the system from those parts
* Prototype becomes the actual production system

1. **Throwaway Prototyping**

* Prototype is not preserved once system is built
* Quickly developed as a mockup

1. **Prototyping is most useful when:**

* User requirements are not clear
* Few users are involved in the system
* Designs are complex and require concrete form to evaluate
* All want specific system requirements as communication problems have existed in the past
* Tools and data are readily available to rapidly build a prototype

1. **Drawbacks of prototyping as a tool include:**

* A tendency to avoid creating formal documentation
* Difficult to adapt to other potential users
* Built as standalones makes it difficult to adapt to other users
* S D L C checks are often bypassed

1. **Business process reengineering (B P R)** – search for, and implementation of, radical change in business processes to achieve breakthrough improvements in products and services

* Reorganize data flow to eliminate unnecessary steps
* Achieve synergy between previously separate steps
* Become more responsive to future changes
* Can be achieved through creative application of information technologies

1. **Key business processes** – structured, measured set of activities designed to produce a specific output for a particular customer or market

* Focused on organizational outcome (e.g., products)
* Same techniques used as requirements determination

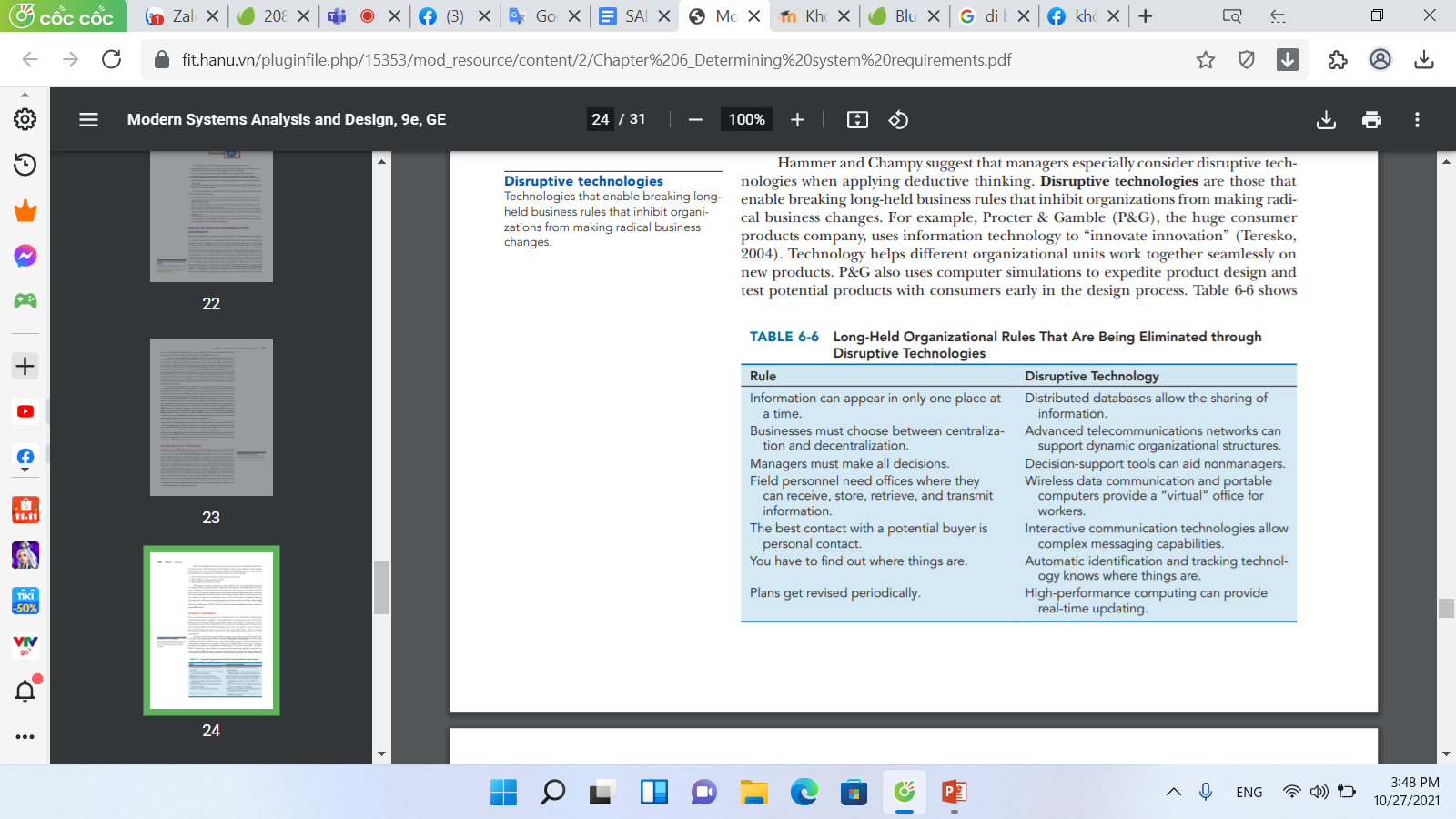
1. **Which activities need radical change?**

* Importance of activity to delivering an outcome
* Feasibility of changing the activity
* Level of dysfunction of current activity

1. Information technologies must be applied to radically improve business processes
2. **Induction** – reasoning from the specific to the general

* Managers learn power of new technologies and ways to innovate and alter how work is done

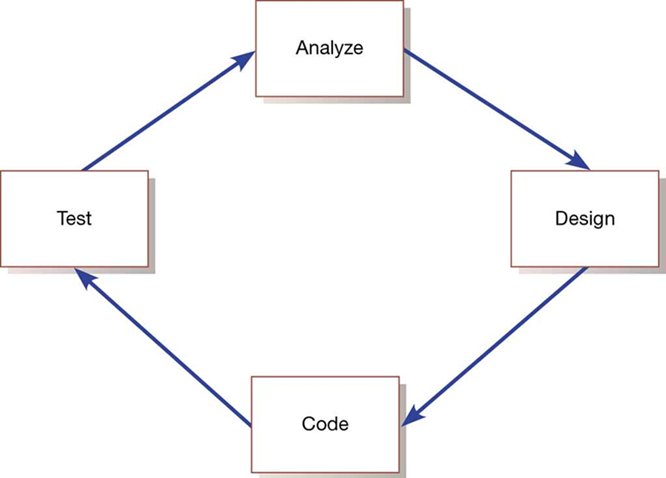
1. **Disruptive technologies** – technologies that enable breaking long-held business rules that inhibit organizations from making radical business changes
2. **Long-Held Organizational Rules That Are Being Eliminated through Disruptive Technologies**



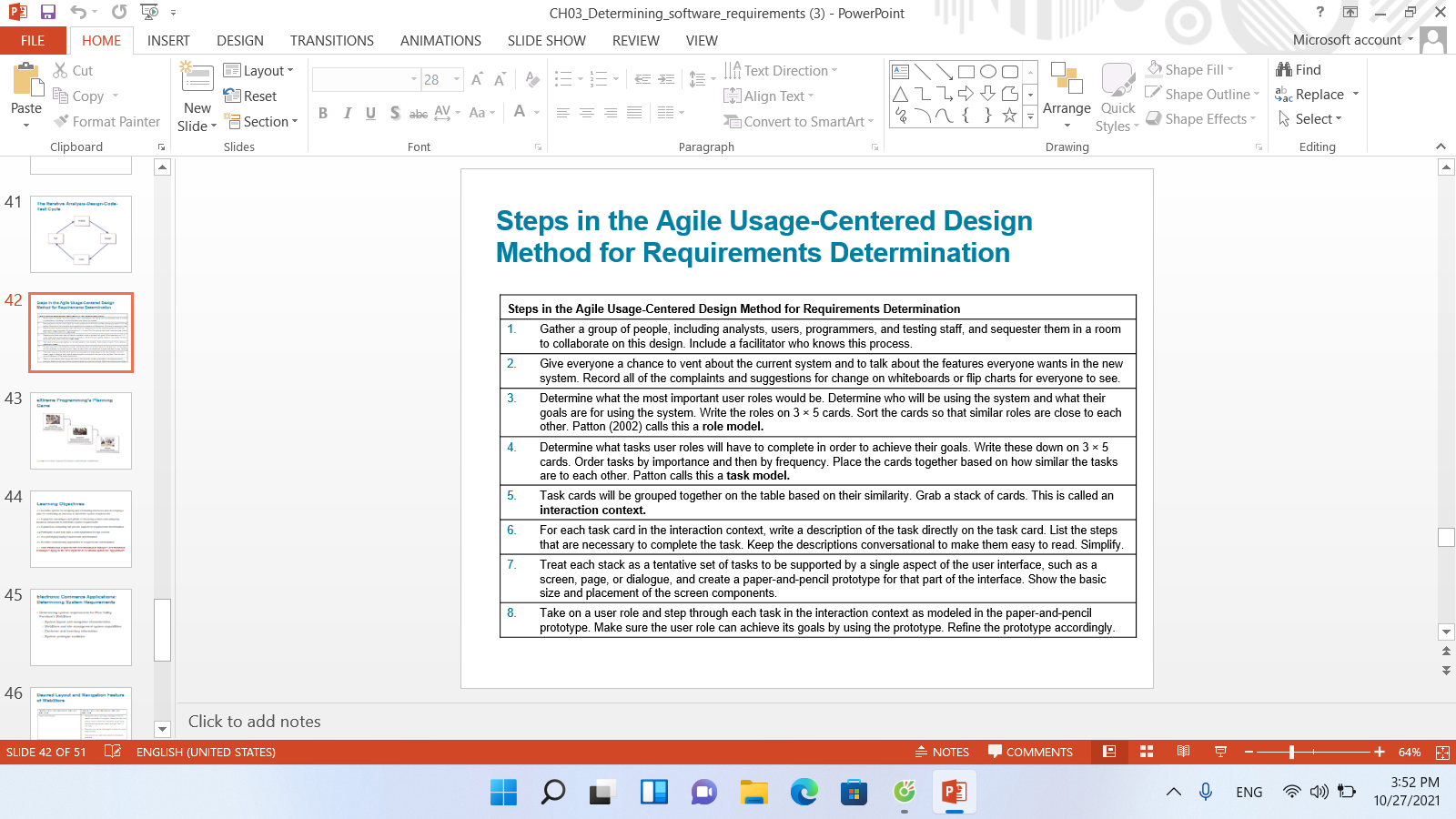
1. **Requirements Determination Using Agile Methodologies**

* Continual user involvement replaces the S D L C with iterative analyze—design—code—test cycle
* Agile usage-centered design focuses on user roles, goals, and tasks to achieve those goals
* The planning game is a stylized approach to development to maximize interaction between those who use and those who build the system

1. **The Iterative Analysis-Design-Code-Test Cycle**



1. **Steps in the Agile Usage-Centered Design Method for Requirements Determination**



**LINK QUIZLET CHAP 3** <https://quizlet.com/264193286/chapter-3-requirements-gathering-flash-cards/>

**Systems development process**

This aids an organization from moving to as-is system to to-be system

**System request**

The output from the planning phase

Provides general ideas for the to-be system, defines the project's scope, and provides the initial work plan

**Analysis Phase**

Takes the general ideas in the system request and refines them into a detailed requirements definition, functional models, structural models, and behavioral models that together form the system proposal

What system will do, who will use system, what value will it provide

**System Walkthrough**

Meeting at which the concept for the new system is presented to the users, managers, and key decision makers

**Goal of system walkthrough**

Explain the system in moderate detail so that the users, managers, and key decision makers clearly understand it, can identify needed improvements, and can make a decision about whether the project should continue

**Analysis deliverables**

What is the first step in design of a new system?

**Requirements determination**

Single most critical step of the entire system development process

Changes can be easily made at this stage

**Purpose of requirements determination**

Turn the very high-level explanation of the business requirements stated in the system request into a more precise list of requirements that can be used as inputs for models

**Requirement**

A statement of what the system must do or what characteristic it must have

**Business Requirements**

These focus on the needs of the business user

**System Requirements**

Requirements in design are written from the developer's perspective

**Functional Requirement**

Relates directly to a process a system has to perform or data it needs to contain

What will the system useful to the user? Why will they be attracted to this system?

Display of data, and manipulate that data

Requirements that state that a system must have the ability to search for available inventory or report expenses

**Nonfunctional requirements**

Behavioral properties that the system must have, such as performance and usability

Ability to access the system using a web browser

Easy to read for older users

Secure from hackers

**Nonfunctional Requirements**

Used primarily in design when decisions are made about the user interface, the hardware and software, and the system's underlying physical architecture

**Operational, performance, security, cultural and political**

What kind of nonfunctional requirements are there?

**Operational**

This nonfunctional requirement deals with the physical and technical environment which the system will operate

**Performance**

This nonfunctional requirement deals with the speed, capacity, and reliability of the system

**Security**

This nonfunctional requirement deals with who has authorized access to the system under what circumstance

**Cultural and Political**

This nonfunctional requirement deals with legal requirements and cultural factors

**Globalization**

If the necessary operational environments do not exist for a mobile solution to be developed, it is important to adapt the solution to the local environment

Not reasonable to develop a high technological based solution in an area with no power of communication infrastructure

**Manual Systems**

These systems with user interfaces need to be further kept in mind when designing them.

Think of colors of the UI, these could mean different things in different cultures

**Requirements Definition**

Straightforward text report that simply lists the functional and nonfunctional requirements in an outline format

This document describes to the analysts exactly what the system needs to end up doing

**Ranked from high to low or label with the version that they will be released with**

How are some business requirements prioritized?

**Define the scope of the system**

Most important purpose of the requirements definition

**Both business people and analysts work together**

What is the most effective approach to determining business requirements?

Business process automation

Means leaving the basic way in which the organization operates unchanged and using computer technology to do some of the work

Based on small change that improves process efficiency

Spend long time understanding current system

**Business process improvement**

Efficiency and effectiveness

Less time understanding the current system

Creates process improvements that lead to better effectiveness

Changes to the way the organization operates in order to take advantage of new opportunities offered by technology or to copy what competitors are doing

**Business process reengineering**

Means changing the fundamental way in which the organization operates—"obliterating" the current way of doing business and making major changes to take advantage of new ideas and new technology

Most amount of change-complete change

Revamps the way things work so that the organization is transformed on some level

Spend little time on understanding current system because they identify a new way of doing business

**BPA, BPI, BPR**

Analysts can use these tools when they need to guide the users in explaining what is wanted from a system.

**Creating a requirements definition**

An iterative and ongoing process whereby the analyst collects information with requirements-gathering techniques (e.g., interviews, document analysis), critically analyzes the information to identify appropriate business requirements for the system, and adds the requirements to the requirements definition report

**Determine functional and nonfunctional requirements**

First step in creating a requirements definition

**Requirements gathering**

Second step in creating a requirements definition

Iterative process

Small batches of requirements can be identified and implemented incrementally

**Verify, change and complete the list**

Last step in creating a requirements definition

**Management of requirements**

Hardest part of managing a project

**No access to correct set of users, requirements are inadequate, unknowable at the beginning, verifying and validating**

What problems can arise with regard to requirements determination

Agile development struggles with this

**Understanding current system, identifying improvements, develop requirements for new system**

Three steps of analysis

**RAD or agile development**

When is understanding of the current system skipped

**Waterfall and parallel development**

When is understanding the current system a vital step

**Collect information**

Requirements gathering techniques are use to

**Requirements analysis strategies**

This drives the kind of information that is gathered and how it is ultimately analyzed

**Change the system will change the organization**

The choice of analysis technique to be used is based on the amount of

Problem analysis, Root Cause Analysis

Two popular BPA techniques

**Problem analysis**

Asking the users and managers to identify problems with the current system and suggest problem solution

The most straightforward (and probably the most commonly used) requirements-analysis technique

Small and incremental changes

Provide a new report that currently does not exist

**Root Cause Analysis**

The analyst starts by having the users generate a list of problems with the current system and then prioritize the problems in order of importance

Focus on the cause, not the solution

Each problem is investigated

Always challenge the obvious

**Duration analysis, activity-based costing, informal benchmarking**

Three popular BPI activities

**Duration Analysis**

Detailed examination of the amount of time it takes to perform each process in the current as-is system

Each sub-process is clocked and they are added up to the total time

Bad if total time is more than sum of parts time

Associated with BPI

**Process integration**

Changing the fundamental process so that fewer people work on the input, which often requires changing the processes and retraining staff to perform a wider range of duties

Solution to duration analysis

**Process parallelization**

Changing the process so that all the individual steps are performed at the same time

Solution to duration analysis

**Activity based costing**

The analysts identify the costs associated with each of the basic functional steps or processes, identify the most costly processes, and focus their improvement efforts on them

Associated with BPI

**Informal Benchmarking**

Studying how other organizations perform a business process in order to learn how your organization can do something better

Associated with BPI

**Outcome analysis, technology analysis, activity elimination**

Three BPR activities

**Outcome analysis**

Focuses on understanding the fundamental outcomes that provide value to customers

What does the customer want in the end?

Think carefully about what the organization's products and services enable the customers to do—and what they could enable the customer to do.

An insurance company provides payment for fixing care, what if the company fixed the car or contracted the car to an authorized body shop

**Technology analysis**

Develop a list of important and interesting technologies. Then the group systematically identifies how every technology could be applied to the business process and identifies how the business would benefit.

**Activity elimination**

The analysts and managers work together to identify how the organization could eliminate each activity in the business process, how the function could operate without it, and what effects are likely to occur

**BPR**

What strategy would be appropriate for potential business value?

**BPA**

What strategy would be appropriate for project cost?

**BPR**

What is the most risky strategy with failure of a system

**Interviews**

Most commonly used requirements-gathering technique

Need to talk to managers of the system and the users of the system

Iterative process starting with senior managers, midlevel managers, and users

**Selecting interviewees, designing interview questions, preparing for the interview, conducting the interview, and postinterview follow-up**

Five parts to interview

**Interview Schedule**

Listing all the people who will be interviewed, when, and for what purpose

**Closed ended questions**

Questions that require a specific answer, used to capture specific, factual information

Used when wanting precise information

How many request do you process per day?

**Open ended questions**

Questions that leave room for elaboration on the part of the interviewee

Designed to gather rich information and give the interviewee more control over the information that is revealed during the interview

What do you think about the current systems?

**Probing question**

Follow-up questions that ask for more information or examples

Used whenever the interviewer is not satisfied with his/her understanding of the interviewee's answer, and needs more explanation before moving on to another topic.

Why? Can you give me an example?

**Unstructured Interviews**

Interviews that seek broad and roughly defined information

**Structured interviews**

Takes place when the business process is understood and very specific information is needed

Specific sets of questions are developed before the interviews

**Top down**

The interviewer starts with broad, general issues and gradually works toward more-specific ones

Enables the interviewee to raise a set of big-picture issues before becoming enmeshed in details, so the interviewer is less likely to miss important issues

**Bottom up**

The interviewer starts with very specific questions and moves to broad questions

**Prepare for interview**

Have a general interview plan listing the questions to be asked in the appropriate order, should anticipate possible answers and provide follow-up with them, and should identify segues between related topics

Prepare the interviewee as well, let them prepare for the interview

**How to start an interview**

Explanation of why the interviewer is there and why he or she has chosen to interview the person

**How to end an interview**

Briefly explain what will happen, he or she should reassure the interviewee that his or her time was well spent and very helpful to the project.

**Interview Report**

Describes the information from the interview

**JAD**

Structured process in which ten to twenty users meet together under the direction of a facilitator skilled in JAD techniques.

They discuss opinions and topics

**Problems with JAD**

Sometimes people are reluctant to challenge the opinions of others (particularly their boss), a few people often dominate the discussion, and not everyone participates

**Electronic JAD**

Each participant uses special software on a networked computer to send anonymous ideas and opinions to everyone else

Select participants, Design JAD session, Prepare, Conduct, Follow up

Five steps of JAD

**Select Participants**

This is done the same way as interview, take people who have different views of the system and problems

**Design a JAD session**

Designed and structured using the same principles as interviews

Must be carefully planned

Use top down

**Prepare for JAD session**

Prepare the people you are using, let the people know what is expected to be understood after the session

**Conducting JAD**

Follow a formal agenda, follow schedule, respect other's opinions, accepting disagreement

**Stick to agenda, help group learn jargon and technology, record information in public area**

JAD Facilitators jobs

**JAD post session report**

Describes the information from the JAD and is used and given to managers

**Questionnaire**

Set of written questions used to obtain information from individuals

Used when information and opinion is needed from a large group of people

**Select participants, design questionnaire, send questionnaire, follow up**

Four steps of questionnaire

**Selecting questionnaire participants**

Select a sample, or subset, of people who are representative of an entire group

**Designing questionnaire**

Developing good questions is critical for questionnaires. Questions on questionnaires must be very clearly written and leave little room for misunderstanding, so closed-ended questions tend to be most commonly used

Decide how these answers will be analyzed and used

**Opinion questions**

Ask respondents the extent to which they agree or disagree (e.g., Are network problems common?

**Factual questions**

Seek more precise values (e.g., How often does a network problem occur: once an hour, once a day, once a week?

**Administering the questionnaire**

Getting the questionnaire completed and sent back.

Tell them why and how there information will be used for this questionnaire

**Questionnaire follow up**

Develop a questionnaire report soon after the questionnaire deadline. This ensures that the analysis process proceeds in a timely fashion and that respondents who requested copies of the results receive them promptly.

**Document analysis**

Use this requirement gathering technique to understand the as-is system

Review documentation from the project team while system was being created

Helpful documents: paper reports, memorandums, policy manuals, user-training manuals, organization charts, forms, and, of course, the user interface with the existing system

**Informal system**

Forms or reports that are never used should probably be eliminated. Likewise, boxes or questions on forms that are never filled in (or are used for other purposes) should be rethought

Differences in how the system should be used and how it is really used

**Observation**

The act of watching processes being performed

Enables the analyst to see the reality of a situation, rather than listening to others describe it in interviews or JAD sessions

Often used to supplement interview information

**JAD and interviews**

Depth of information

**Questionnaire and document analysis**

Breadth of information

**JAD**

Integration of information

**Concept maps**

Represent meaningful relationships between concepts

Useful for focusing individuals on the small number of key ideas on which they should concentrate

**System Proposal**

Brings together into a single comprehensive document the material created during planning and analysis

Includes an executive summary, the system request, the workplan, the feasibility analysis, the requirements definition, and the evolving models that describe the new system. The evolving models include functional models, structural models, and behavioral models.

**Executive summary**

Provides all critical information in a very concise form. It can be thought of as a summary of the complete proposal. Its purpose is to allow a busy executive to quickly read through it and determine which parts of the proposal he or she needs to go through more thoroughly

**Story card**

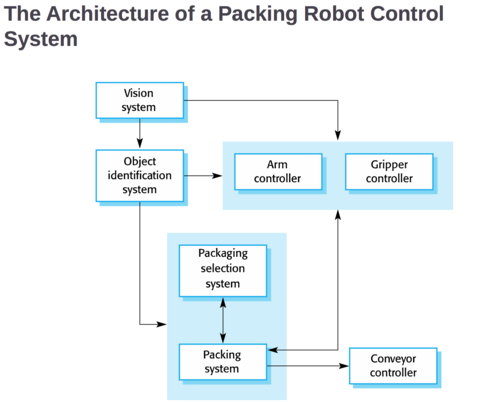
Typically an index card with a single requirement (functional or nonfunctional) written on it

It is discussed to determine the amount of effort it will take to implement it

**Chap 4: Designing architecture**

1. **What is Architectural Design:**

Architectural design is concerned with understanding how a software system should be organized and designing the overall structure of that system. Architectural design is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

****

1. **informal design:**

when a product is made by the designer without the use of a plan

1. **formal design:**

involves the complete preparation of a set of working drawings

1. **"form follows function":**

any architectural form should have an intended practice purpose and should perform a function

1. **functionalism:**

quality of being useful (serving a purpose other than adding beauty or aesthetic value)

1. **organic design**

all materials, functions, forms, and surroundings are completely coordinated and harmony with nature

1. **eclectic design**

mixed textures, time periods, styles, colors

1. **trends:**

general developments

1. **fads:**

temporary popular fashions

1. **BASIC ELEMENTS OF DESIGN**

line

form

space

color

light and shadow

texture and materials

1. **LINE:**

encloses space and provides the outer line or contour of forms

they can produce a sense of movement or produce a greater sense of length or height

gives a sense of stability

1. **vertical lines:**

create height by using strength and alertness

1. **horizontal lines:**

emphasize width by expanding the perception of space

1. **diagonal lines:**

restlessness or transition

1. **curved lines:**

soft, graceful, flowing movements

1. **Form:**

lines joined together produce form and create space

1. **circles and ovals:**

create a feeling of completeness

1. **principles of design:**

balance

rhythm

repetition

emphasis

subordination

proportion

unity

variety

opposition

transition

1. **Bungalows changed architecture forever because they were:**

cost effective

more practical

new technology

moved across the country

1. **Design is the process of?:**

Figuring out how to implement all of the customer's requirements

1. **Early design decisions address how to implement the individual units, and later design decisions address the system's architecture:**

False, it's reversed.

Early design decisions address the system's architecture. Later design decisions address how to implement the individual units

1. **Is design intellectually challenging?**

Yes. Many possibilities, nonfunctional design goals, eternal factors.

1. **Cloning:**

Borrowing design/code in its entirety, with minor adjustments

1. **Reference models**

Generic architectures that suggest how to decompose the system

1. **T/F: For most problems there is no reference model**

True

1. **Architectural Styles**

Generic solutions of software architectures. Focusing on one can create problems.

1. **Give 2 examples of tools for understanding options and evaluating chosen architecture.**

-Design Patterns

-Design Convention/Idiom

-Design Principles

1. **Design Patterns**

Generic solutions for making lower-level design decisions

1. **Design Convention/Idiom**

A collection of design decisions and advice that, taken together, promotes certain design qualities

1. **Design Principles**

Descriptive characteristics of good design

1. **Designing software systems is an \_\_\_\_\_\_\_\_\_\_ process.**

Iterative

1. **What are the 6 ways to use architectural models?**

1. understand system

2. determine amount of reuse from other systems

3. provide blueprint for construction

4. reason about system evolution

5. analyze dependencies

6. support management decisions and understand risks

1. **Use top down and bottom up to do what?**

Create a hierarchy of info with increasing details

1. **What are 3 popular design methods?**

1. functional decomposition

2. feature oriented decomposition

3. object oriented design

1. **Functional decomposition**

Partitions functions or reqs into modules. Low-level design divide into subfunctions and describes which subfunctions call each other

1. **Feature-oriented decomposition**

Assigns features to modules. High-level design describes system in terms of service. Low-level design describe how each feature augment service

1. **Object-oriented decomposition**

Assigns objects to modules. High-level IDs system's object types and explains relations. Low-level details attributes and operations

1. **What are the 5 parts of the Design Process Model?**

Modelling, analysis, documentation, review, software architecture document (SAD)

1. **T/F: It is helpful to use Agile processes when there is uncertainty about requirements**

True

1. **Functional Decomposition describes?**

Which modules/subfunctions call each other

1. **For feature-oriented decomposition, the low-level design describes?**

How each feature augments the service and identifies interactions among features

1. **Object-Oriented Decomposition is characterized by?**

Assigning objects to modules

1. **In Object-Oriented Decomposition, high-level design identifies?**

The system's object types

1. **In Object-Oriented Decomposition, low-level design details?**

The objects' attributes and operations

1. **A design is modular when?**

Each activity of the system is performed by exactly one software unit, and the inputs/outputs of each unit are well-defined

1. **A software unit is well-defined when?**

Its interface accurately and precisely specifies the unit's externally visible behavior

1. **What is a component?**

"a self-contained piece of software with a well-defined set of interfaces" that can be developed, bought, and sold as a distinct entity

1. **Give 3 common types of architectural views.**

-Dependencies

-Generalization

-Work-Assignment

1. **Dependencies View**

Shows dependence among software units. Good for planning and assessing impact of making design change

1. **Generalization View**

Software units are generalizations, like Obj-O inheritance. Good for abstract software units

1. **Work-Assignment View**

Decomposes the system into work tasks for teams. Good for manager planning and allocating resources

1. **List the 5 architectural styles**

-Pipes-and-Filter

-Client-Server

-Peer-to-Peer

-Publish-Subscribe

-Repositories

1. **Architectural styles**

take advantage of common features by applying generalized patterns

1. **In Pipes-And-Filter, the pipes are \_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ and the filter is the \_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_**

Streams of data; the transformation of the data

1. **T/F filters can be easily reused in other systems and are simple to resu**

True

1. **3 drawbacks of pipe-and-filter**

-Encourages batch processing

-Not good for handling interactive application

-Duplication in filters functions

1. **The 2 types of components in Client-Server Style**

-server components who offer services

-client components who access server components using a request/reply protocol

1. **Callback (client-server)**

an executable function a client may send a server, which subsequently calls under circumstances

1. **Peer-to-peer**

Each component acts as its own process and acts as both client and server

1. **T/F In P2P style, a component can initiate a request to any server component**

False, a component can initiate a request to any peer component

1. **Pros of P2P (3)**

Scales up well, increased system capabilities, highly tolerant of failures

1. **In Publish-Subscribe style, components interact by?**

Broadcasting and reacting to events (implicit invocation)

1. **How does publish-subscribe work?**

Component expresses interest by subscribing. When another component publishes event, subs notified.

1. **Registering (publish-subscribe)**

when a subscribing component associates one of its procedures with each event of interest

1. **Pros and cons of publish-subscribe**

Pro: strong support for evolution and customization; easy to reuse components in other systems.

Cons: difficult to test and need shared repository to share persistent data

1. **2 components in Repositories**

-has a central data store

-A collection of components that operate on it

1. **3 interaction methods of Repositories**

-Traditional database

-Blackboad

-Knowledge Sources

1. **Traditional database**

Transactions trigger process executions

1. **Blackboard**

Central store controls triggering process

1. **Knowledge sources**

Info about system's execution that triggers execution of individual data accessor

1. **T/F Openness is the major advantage of repositories**

True. Available to many programmers but data format must be acceptable to all components

1. **T/F For combining architectural styles, you usually use one**

False. You combine with different styles at different layers or mix styles to model different component interaction. Need documentation

1. **To support specific quality attribute, 7 tactics are utilized:**

- Modifiability

- Performance

- Security

- Reliability

- Robustness

- Usability

- Business goals

1. **Modifiability**

Design easy to change. In/direct affected

1. **When a change is made, directly-affected units...**

Change in responsibilities to accommodate the change

1. **When a change is made, indirectly-affected units**

Don't change. They revise their implementation but not their responsibilities

1. **To minimize the number of software units affected by change, focus on \_\_\_\_\_\_\_\_\_\_\_\_ anticipated changes**

Clustering

1. **What are the 3 changes under modifiability?**

Anticipate expected changes, cohesion, and generality, coupling, interfaces, multiple interfaces

1. **Anticipate expected changes (mod)**

ID design most likely to change and encapsulate,

1. **Cohesion (mod)**

Keep software units cohesive to increase chance responsibilities confined to few units assigned to those responsibilities

1. **Generality (mod)**

The more general the units, the more likely change accommodate unit inputs instead of whole unit

1. **Coupling (mod)**

Lowering reduce likelihood that change one unit will ripple effect

1. **Interfaces (mod)**

If unit interacts with other units only through interfaces, change no spread beyond unit's boundary while interface constant

1. **Multiple interfaces (mod)**

Modded unit providing new data or service can offer new interface without changing unit's existing interface

1. **Performance**

Describes constraints on system speed and capacity

1. **3 perfomance attributes**

-Response time (how fast it responds to requests)

-Throughput (how many requests can be processed in a minute)

-Load (How many users can be supported before the other two suffer)

1. **Ways to improve performance**

-Good resource-allocation scheduling

-improve utilization of resources

-reduce demand for resources

-FIFO, explicit priority

1. **The two key characteristics relevant to security**

Immunity and Resilience

1. **Immunity (security)**

Ability to thwart an attempted attack

Minimize exploits

1. **Resilience (security)**

Ability to recover quickly and easily from an attack

Segment functionality to contain attack

1. **A system is \_\_\_\_\_\_\_\_ if it correctly performs its required functions under assumed conditions**

Reliable

1. **T/F A FAULT is the result of human error, while a FAILURE is an observable departure of a system from its required behavior**

True

1. **T/F Failures are seen by system developers, faults are seen by users and customers**

False, it is switched

1. **A system is \_\_\_\_\_\_ if it includes mechanisms for accommodating or recovering from problems in the environment or in another unit**

Robust

1. **Mutual suspicion is where each software unit does what?**

Assumes the other units contain faults

1. **Examples of recovery tactics (robust)**

- Rollback to checkpoint

- Abort transaction

- Backup

- Reduced service

- Trigger exception

1. **Usability**

The ease of a user to operate the system

1. **T/F UI's should be combined into various software units**

False, it should be in its own separate software unit

1. **Business goals**

Quality attributes the system is expected exhibit

1. **Build vs Buy (business)**

Save dev time and money, more reliable, components create constraints and vulnerable to suppliers

1. **Initial dev vs. maintenance cost (business)**

Save money by making moddable, increased complexity delay release and lose market share

1. **New vs. known tech (business)**

Acquire expertise cost money and delays product, learn new tech or hire someone

1. **Give 6 techniques to evaluate the design.**

- Measuring design quality

- Safety analysis

- Security analysis

- Trade-off analysis

- Cost-benefit analysis

- Prototyping

1. **T/F Fault tree analysis traces backward through a design**

True

1. **Fault trees are used to?**

Determine which faults to correct/avoid/tolerate

1. **Data-flow graph depicts?**

The transfer of data from one process to another

1. **Control-flow graph depicts?**

Possible transfer of control among software units

1. **T/F Once fault tree constructed don't have to search for weaknesses.**

False. You do have to search

1. **Rules for forming cut-set trees (safety analysis)**

- Assign top node to match LOGIC GATE

- Working top-down, expand (OR, AND gate)

- Go until visit all events

- If find fault, correct it, mod to prevent and detect fault and recover

1. **T/F There is a professional duty to explore design alternatives**

True

1. **Cost-Benefit Analysis**

Widely used business tool for estimating and comparing the costs and benefits of a proposed change

1. **4 architectural designs to implement OS,MD**

shared data, abstract data/data module, implicit invocation, pipe and filter

1. **T/F A Cost-Benefit Analysis contrasts financial benefits with expected man-hours**

False, financial benefits with financial costs

1. **Payback period**

length of time before accumulating benefits recover costs of implementation

1. **System arch is vital to overall dev and serves as basis on decisions for:**

Design, QA, project management

1. **System Architecture Design includes (6):**

- System Overview

- Views

- Software Units

- Analysis Data and Results

- Design Rationale

- Definitions/glossary/acronyms

1. **Ways of evaluating SAD quality (2)**

Validation (design satisfies all requirements) and Verification (design aheres to good design principles)

1. **Give 8 people involved in design validation review**

- Analyst who defined requirements

- System architect

- Program designer for project

- System tester

- System maintainer

- Moderator

- Recorder

- Other interested developers not involved in the project

1. **Passive review process (verification)**

Reading the documentation and looking for problems

1. **Active review process (verification)**

Exercise the design document in ways developers will use it in practice

1. **The product family's collection of reusable assets are stored where?**

In a code asset base

1. **T/F Products in a product line are developed sequentially down the line**

False, their simultaneous development is planned from the beginning

1. **The cost of developing the common product line must what?**

Be more than offset by the savings expected

1. **Product lines are based on the \_\_\_\_\_\_\_\_ among projects, and the best way to \_\_\_\_\_ \_\_\_\_**

Commonalities,Exploit them

1. **Candidate elements in a core asset base (7)**

- Reqs

- Software architecture

- Models and analysis results

- Software units

- Testing

- Project planning

- Team organization

1. **Advantages of Product-line architecture**

Promote planned modifiability

1. **A key contributor to the product-line success if having a \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_.**

Product-line mindset(It's backwards capable)

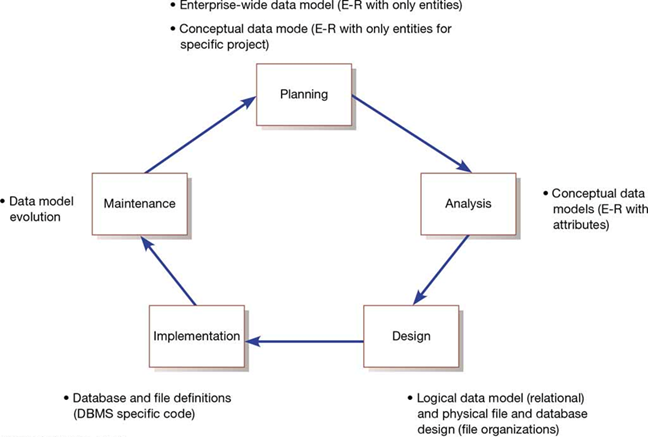
**Chap 5 : Designing Databases**

1. **Database Design**

File and database design occurs in two steps:

* Develop a logical database model, which describes data using notation that corresponds to a data organization used by a database management system
* Relational database model
* Prescribe the technical specifications for computer files and databases in which to store the data
* Physical database design provides specifications
* Logical and physical database design in parallel with other system design steps

1. **The process of Database Design**



* **Normalization**: a way to build a data model that has the properties of simplicity, nonredundancy and minimal maintenance.
* **Four key steps** in **logical database** modeling and design:
* Develop a logical data model for each known user interface for the application using normalization principles
* Combine normalized data requirements from all user interfaces into one consolidated logical database model (view integration)
* Translate the conceptual E-R data model for the application into normalized data requirements
* Compare the consolidated logical database design with the translated E-R model and produce one final logical database model for the application
* Key **physical database** design decisions:
* Choosing a storage format (data type) for each attribute from the logical database model
* Grouping attributes from the logical database model into physical records
* Arranging related records in secondary memory (hard disks and magnetic tapes) so that records can be stored, retrieved and updated rapidly
* Selecting media and structures for storing data to make access more efficient

1. **Deliverables and Outcomes**

* **Primary Key:** attribute whose value is unique across all occurrences of a relation
* Physical database design converts relations into database tables: Programmers and database analysts code the definitions of the database using Structured Query Language (SQL)

1. **The relational database model**

* **Relational database model**: data represented as a set of related tables or relations
* **Relation**: named, two-dimensional table of data. Each relation consists of a set of named columns and an arbitrary number of unnamed rows.
* Relations have **several properties** that distinguish them from non relational tables:
* Entries in cells are simple
* Entries in columns are from the same set of values
* Each row is unique
* The sequence of columns can be interchanged without changing the meaning or use of the relation
* The rows may be interchanged or stored in any sequence
* **Well-structured relation**: relation that contains a minimum amount of redundancy and that allows users to insert, modify, and delete the rows without error or inconsistencies; also known as a table

1. **Rules of Normalization:**

* **First Normal Form (1NF)**: Has no multivalued attributes, unique rows, and all relations are in 1NF
* **Second Normal Form (2NF)**: Each nonprimary key attribute is identified by the whole key (referred to as a full functional dependency)
* **Third Normal Form (3NF)**: Nonprimary key attributes do not depend on each other (referred to as a transitive dependency)

1. **Functional Dependence and Primary Keys**

* **Foreign key:** attribute that appears as a nonprimary key attribute in one relation and as a primary key attribute (or part of a primary key) in another relation
* **Referential integrity:** rule that states that either each foreign key value must match a primary key value in another relation or the foreign key value must be null (i.e., have no value)

1. **Represent entities**

* Each regular entity is transformed into a relation
* The identifier of the entity type becomes the primary key of the corresponding relation
* The primary key must satisfy the following two conditions
* The value of the key must uniquely identify every row in the relation
* The key should be non redundant
* The entity type label is translated into a relation name

1. **Representing relationships**

* **Binary 1:N Relationship** – represented by adding the primary key attribute (or attributes) of the entity on the one side of the relationship as a foreign key in the relation that is on the many side of the relationship
* **Binary or Unary 1:1 relationship** is represented by any of the following:
* Add the primary key of A as a foreign key of B
* Add the primary key of B as a foreign key of A
* Both of the above
* **Binary and Higher-Degree M:N Relationships** – represented by creating another relations, include the primary keys of all relations into the new one as a primary key
* Becomes a composite key
* Any non-key attributes associated with the M:N relationship are included in the new relation
* Unary **1:N Relationship** (also called recursive relationships):
* Is modeled as a relation
* Primary key of that relation is the same as for the entity type
* Foreign key is added to the relation that references the primary key values
* Unary **M:N Relationship** is modeled as one relation, then:
* Create a separate relation to represent the M:N relationship
* The primary key of the new relation is a composite key of two attributes that both take their values from the same primary key
* Any attribute associated with the relationship is included as a nonkey attribute in this new relation
* **Recursive foreign key:** foreign key in a relation that references the primary key values of that same relation

1. **Merging relations**

* Is the last step in the logical database design
* Purpose is to remove redundant relations
* **Synonym**: two different names used for the same attribute. When merging, get agreement from users on a single, standard name
* **Homonym**: single attribute name that is used for two or more different attributes. Resolved by creating a new descriptive name
* **Dependencies between nonkeys occurs** when two 3NF relations are merged to form a single relation
* If a transitive dependency exists such as Major → Advisor, you need to normalize to remove the transitive dependency

1. **Physical File and Database Design**

* **Designing physical files/databases** requires the following information:
* Normalized relations, including volume estimates
* Definitions of each attribute
* Descriptions of where and when data are used: entered, retrieved, deleted, and updated (including frequencies)
* Expectations or requirements for response time and data integrity
* Descriptions of the technologies used for implementing the files and database so that the range of required decisions and choices for each is known
* **Field**: smallest unit of named application data recognized by system software
* An attribute from a relation is now recognized as a field in a database
* **Data type**: coding scheme recognized by system software for representing organizational data
* **Selecting a data type** requires **balancing four objectives**:
* Minimize storage space
* Represent all possible values of the field
* Improve data integrity of the field
* Support all data manipulations desired on the field
* **Calculated field**: field that can be derived from other database fields. Also known as a computed field or a derived field.
* **Default value**: value a field will assume unless an explicit value is entered for that field
* **Range control**: limits values (numeric or alpha-numeric data) that can be entered into a field
* **Referential integrity**: constraint specifying that the value (or existence) of an attribute in one relation depends on the value (or existence) of the same attribute in another relation
* **Null value**: special field value, distinct from zero, blank, or any other value, that indicates that the value for the field is missing or otherwise unknown

1. **Designing Physical Table**

* **Physical table**: named set of rows and columns that specifies the fields in each row of the table
* **Denormalization:** process of splitting or combining normalized relations into physical tables based on affinity of use of rows and fields
* Partitioning is the capability to split a table into separate sections. Partitioning types include:
* **Range partitioning:** partitions are defined by nonoverlapping ranges of values for a specified attribute
* **Hash partitioning:** a table row is assigned to a partition by an algorithm and then maps the specified attribute value to a partition
* **Composite partitioning:** combines range and hash partitioning by first segregating data by ranges on the designated attribute, and then within each of these partitions
* **Three common situations** where denormalization is used are:
* Two entities with a one-to-one relationship
* A many-to-many relationship (associative entity) with nonkey attributes
* Reference data
* **Physical file:** named set of table rows stored in a contiguous section of secondary memory
* **File organization:** technique for physically arranging the records of a file
* Objectives for choosing file organization include:
* Fast data retrieval
* High throughput for processing transactions
* Efficient use of storage space
* Protection from failures or data loss
* Minimizing need for reorganization
* Accommodating growth
* Security from unauthorized use
* **Pointer**: field of data that can be used to locate a related field or row of data
* Three basic families of file organization:
* **Sequential file organization** – file organization in which rows in a file are stored in sequence according to a primary key value
* **Indexed file organization** – file organization in which rows are stored either sequentially or nonsequentially, and an index is created that allows software to locate individual rows
* **Hashed file organization** – file organization in which the address of each row is determined using an algorithm
* **Index:** table used to determine the location of rows in a file that satisfy some condition
* **Secondary key:** represents one or a combination of fields for which more than one row may have the same combination of values
* Allows an index to point to more than one record

1. **Indexed file organization**

* Advantage is allowing for both random and sequential processing
* Disadvantages include:
* Extra space required to store indexes
* Extra time necessary to access and maintain indexes
* Guidelines for choosing indexes include:
* Specify a unique index for the primary key of each table
* Specify an index for foreign keys
* Specify an index for nonkey fields that are referenced in qualification, sorting and grouping commands for the purpose of retrieving data

1. **Designing Controls for Files**

* Techniques for file restoration:
* Periodically making a backup copy of a file
* Storing a copy of each change to a file in a transaction log or audit trail
* Storing a copy of each row before or after it is changed
* Means of building data security into a file:
* Coding, or encrypting, the data in the file
* Requiring data file users to identify themselves by entering usernames and passwords
* Prohibiting users from directly manipulating any data in the file by forcing users to work with a copy (real or virtual)

The first step in representing entities using the relational model is to determine which identifier will be used as the key. FALSE

Relations should always be normalized to the highest degree possible. FALSE

If a weak entity is ID-dependent but not existence-dependent, it can be represented using the same techniques as a strong entity. FALSE

The key of the parent entity becomes part of the key of an ID-dependent entity. TRUE

From a pragmatic standpoint, the only important rule of normalization is that the determinant of every functional dependency must be a candidate key. TRUE

An entity needs to be examined according to normalization criteria before creating a table from it in the relational database design. FALSE

When creating a table in the relational database design from an entity in the extended E-R model, the attributes of the entity become the rows of the table. FALSE

By default, the identifier of the entity becomes the foreign key of the corresponding table. FALSE

The ideal primary key is short, numeric, and fixed. TRUE

A surrogate key is appropriate when the primary key of a table contains a lengthy text field. TRUE

One of the important properties of an attribute is whether or not it is required. TRUE

The technique for representing E-R relationships in the relational model is dependent on the minimum cardinality. FALSE

For a 1:1 relationship, the key of each table should be placed in the other table as the foreign key. FALSE

Relationships that are 1:1 do not require referential integrity constraints. FALSE

In certain circumstances, there may be a preference as to which table in a 1:1 relationship contains the foreign key. TRUE

When applied to 1:N relationships, the term "parent" refers to the many sides of the relationship since a child may have many parents. FALSE

To represent a 1:N relationship in the relational model, the key of the entity on the one side of the relationship is placed as a foreign key in the entity on the many sides of the relationship. TRUE

To represent a 1:N relationship in the relational model, the key of either entity may be placed as a foreign key in the other entity. FALSE

In the relational model, many-to-many relationships cannot be directly represented by relations the way 1:1 and 1:N relationships can. TRUE

To represent a M:N relationship in the relational model, an intersection relation is created to represent the relationship itself. TRUE

The key for an intersection relation is always the combination of the keys of the parent entities. TRUE

All recursive relationships are 1:1. FALSE

Recursive relationships can be represented in the relational model using the same techniques that are used for binary relationships. TRUE

The first step in transforming an extended E-R model into a relational database design is to \_\_\_\_\_\_\_\_. create a table for each entity

Each attribute of an entity becomes a(n) \_\_\_\_\_\_\_\_ of a table. column

The identifier of the entity becomes the \_\_\_\_\_\_\_\_ of the corresponding table. primary key

The ideal primary key is \_\_\_\_\_\_\_\_. short, numeric and fixed

A surrogate key should be considered when \_\_\_\_\_\_\_\_. the key contains a lengthy text field

Which of the following is not true about surrogate keys? They are nonunique within a table.

One of the important properties of a column is whether or not it is \_\_\_\_\_\_\_\_. required

In a relational database design, all relationships are expressed by \_\_\_\_\_\_\_\_. creating a foreign key

Which of the following would be a reason to denormalize a relation? Improve performance

Which of the following is true about representing a weak entity with the relational model?

If the weak entity is ID-dependent, the key of the parent entity must be part of the key of the weak entity.

Which of the following is true when representing a 1:1 binary relationship using the relational model? The key of either entity is placed in the other as a foreign key.

Given the tables

TABLE\_A (Attribute1, Attribute2, Attribute3)

TABLE\_B (Attribute4, Attribute5, Attribute6)

as shown in the figure below, which of the following would display the correct placement of foreign keys in the relational model? TABLE\_A (Attribute1, Attribute2, Attribute3)

TABLE \_B (Attribute4, Attribute5, Attribute6, Attribute1)

Which of the following is the correct technique for representing a 1:N relationship in the relational model? The key of the entity on the one side is placed into the relation for the entity on the many side.

Given the tables

PRODUCT (ProductID, Description, Cost)

SUPPLIER (SupplierID, ContactName, PhoneNumber)

as shown in the figure below, which of the following would represent the correct placement of foreign keys? PRODUCT (ProductID, Description, Cost, SupplierID)

SUPPLIER (SupplierID, ContactName, PhoneNumber)

Which of the following is the correct technique for representing a M:N relationship using the relational model? An intersection relation is created, and the keys of both parent entities are placed as a composite key in the intersection relation.

Given the tables

PRODUCT (ProductID, Description, Cost)

SUPPLIER (SupplierID, ContactName, PhoneNumber)

as shown in the figure below, which of the following would represent the correct placement of foreign keys? PRODUCT (ProductID, Description, Cost)

SUPPLIER (SupplierID, ContactName, PhoneNumber)

PRODUCT\_SUPPLIER (ProductID, SupplierID)

In many-to-many relationships in a relational database design, \_\_\_\_\_\_\_\_. the keys of both tables are placed in a third table and are joined into a composite key

In many-to-many relationships in a relational database design, \_\_\_\_\_\_\_\_. the intersection table is ID-dependent on both of the parents; the minimum cardinality from the intersection table to the parents is always M

In relational database design, ID-dependent entities are used to \_\_\_\_\_\_\_\_. represent N:M relationships

When transforming an E-R data model into a relational database design, the key of the parent entity should be placed as part of the primary key into the child entity \_\_\_\_\_\_\_\_. when the child entity is ID-dependent

When transforming an ID-dependent E-R data model relationship into a relational database design and the child entity is designed to use a surrogate key, then \_\_\_\_\_\_\_\_. the relationship changes to a non-ID-dependent relationship

What relationship pattern is illustrated in the following schema?

PRODUCT (ProductID, Description)

SUPPLIER (SupplierID, ContactName, PhoneNumber)

PRODUCT\_SUPPLIER (ProductID, SupplierID, Cost)

ProductID in PRODUCT\_SUPPLIER must exist in ProductID in PRODUCT

SupplierID in PRODUCT\_SUPPLIER must exist in SupplierID in PRODUCT

Association relationship

What relationship pattern is illustrated in the following schema?

VEHICLE (VehicleID, Cost)

CAR (VehicleID, NumberOfSeats)

TRUCK (VehicleID, CargoCapacity)

VehicleID in CAR must exist in VehicleID in VEHICLE

VehicleID in TRUCK must exist in VehicleID in VEHICLE

Supertype/subtype relationship

Which of the following is not true about representing subtypes in a relational database design? All of the attributes of the supertype are added to the subtype relations.

What relationship pattern is illustrated in the following schema?

EMPLOYEE (EmployeeID, OfficePhone, Manager)

Manager in EMPLOYEE must exist in EmployeeID in EMPLOYEE

Recursive relationship

Which of the following is not true of recursive relationships? Even when the relationship is 1:N, a new table must be defined to represent the relationship.

The first step of database design is to define a table for each \_\_\_\_\_\_\_\_. ENTITY

Once a table has been defined, it should be examined according to \_\_\_\_\_\_\_\_ criteria. NORMALIZATION

There are cases where it is possible to normalize a table too far, in which case there may be a need for \_\_\_\_\_\_\_\_. DENORMALIZATION

To normalize a relation, the determinant of every functional dependency should be a(n) \_\_\_\_\_\_\_\_. CANDIDATE KEY

To represent a many-to-many relationship in the relational model, a(n) \_\_\_\_\_\_\_\_ table is used. INTERSECTION

For a(n) \_\_\_\_\_\_\_\_ weak entity, it is necessary to add the key of the parent entity to the weak entity's relation so that this added attribute becomes part of the weak entity's key. ID-DEPENDENT

A(n) \_\_\_\_\_\_\_\_ is a relationship among entities of the same class. RECURSIVE RELATIONSHIP

**Chap 6: Designing Interfaces**

1. **Designing Interface with Forms and Reports**

* **Forms:** business document that contains some predefined data and may include some areas where additional data are to be filled in. An instance of a form is typically based on one database record.
* Usually in a stylized format and not a simple row and column look
* **Reports:** business document that contains only predefined data; it is a passive document used solely for reading or viewing. A report typically contains data from many unrelated records or transactions
* Reports typically contain data from many unrelated records or transactions

1. **Common types of Business Reports**

| **Report Name** | **Description** |
| --- | --- |
| Scheduled Reports | Reports produced at predefined intervals—daily, weekly, or monthly—to support the routine informational needs of an organization. |
| Key-Indicator Reports | Reports that provide a summary of critical information on a recurring basis. |
| Exception Reports | Reports that highlight data that are out of the normal operating range. |
| Drill-Down Reports | Reports that provide details behind the summary values on a key-indicator or exception report. |
| Ad-hoc Reports | Unplanned information requests in which information is gathered to support a nonroutine decision. |

1. **The process of designing interfaces**

* A user-centered activity that follows a prototyping approach:
* Understanding the target audience and their needs
* Collect initial requirements
* Structure and refine this information into an initial prototype
* Users then evaluate the prototype
* Make any needed refinements until users are satisfied and accept the prototype
* **Paper prototype:** series of mock screens that can be used to test content, look, and feel, as well as the task flow and other usability factors
* Focus is on the design (content, layout, flow)
* **Wireframe:** simple design to show the placement of information elements on a screen and the space needed for each element
* Allows users to get a sense and feel of a design
* A **coding sheet** is an “old” tool for designing forms and reports in text-based format
* A **wireframe** is an input screen roughed out on a sheet of paper
* A **data input screen** designed in Visual Basic allows a variety of font sizes, colors, and highlighting

1. **Deliverables and outcomes**

* Design specifications are the major deliverables and contain three sections:
* **Narrative overview**: Characterizes, users, tasks, system, and environmental factors
* **Sample design:** Image of the form is more thoroughly tested and assessed
* **Testing and usability assessment:** Provides all testing and assessment information; Determine usability

**General Guidelines for the Design of Interfaces**

•**Meaningful Titles:**

–Clear and specific titles describing content and use of form or report

–Revision date or code to distinguish a form or report from prior versions

–Current date, which identifies when the form or report was generated

–Valid date, which identifies on what date (or time) the data in the form or report were accurate

•**Meaningful Information:**

–Only needed information should be displayed

–Information should be provided in a manner that is usable without modification

•**Balance the Layout:**

–Information should be balanced on the screen or page

–Adequate spacing and margins should be used

–All data and entry fields should be clearly labeled

•**Design an Easy Navigation System:**

–Clearly show how to move forward and backward

–Clearly show where you are (e.g., page 1 of 3)

–Notify user when on the last page of a multipaged sequence

**Guidelines for Displaying Text**

| Case | Display text in mixed uppercase and lowercase and use conventional punctuation. |
| --- | --- |
| Spacing | Use double spacing if space permits. If not, place a blank line between  paragraphs. |
| Justification | Left-justify text and leave a ragged-right margin. |
| Hyphenation | Do not hyphenate words between lines. |
| Abbreviations | Use abbreviations and acronyms only when they are widely understood by users and are significantly shorter than the full text. |

**Contrasting the Display**

* Poorly-Designed Help Screen with Many Violations of the General Guidelines
* An Improved Design for a Help Screen

**General Guidelines for Displaying Tables and Lists**

* **Use meaningful labels:**
* All columns and rows should have meaningful labels
* Labels should be separated from other information by using highlighting
* Redisplay labels when the data extend beyond a single screen or page
* **Formatting columns, rows, text:**
* Sort in a meaningful order (e.g., ascending, descending, or alphabetic)
* Place a blank line between every five rows in long columns
* Similar information displayed in multiple columns should be sorted vertically (i.e., read from top to bottom, not left to right)
* Columns should have at least two spaces between them
* Allow white space on printed reports for user to write notes
* Use a single typeface, except for emphasis
* Use same family of typefaces within and across displays and reports
* Avoid overly fancy fonts
* **Formatting numeric, textual, and alphanumeric date:**
* Right-justify numeric data and align columns by decimal points or other delimiter
* Left-justify textual data. Use short line length, usually 30–40 characters per line (this is what newspapers use, and it is easier to speed-read).
* Break long sequences of alphanumeric data into small groups of three to four characters each
* **General Guidelines for Displaying Tables and Lists**

**Use meaningful labels:**

* All columns and rows should have meaningful labels
* Labels should be separated from other information by using highlighting
* Redisplay labels when the data extend beyond a single screen or page

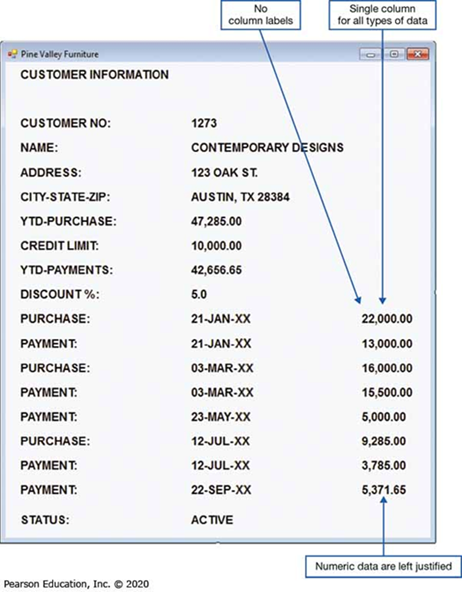
**Formatting columns, rows, text:**

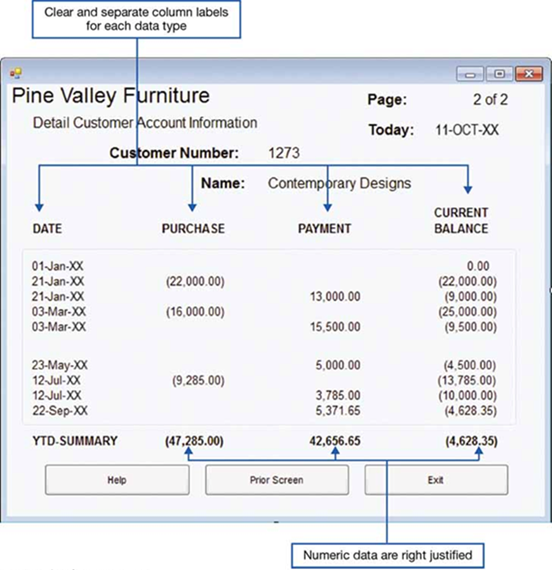
* Sort in a meaningful order (e.g., ascending, descending, or alphabetic)
* Place a blank line between every five rows in long columns
* Similar information displayed in multiple columns should be sorted vertically (i.e., read from top to bottom, not left to right)
* Columns should have at least two spaces between them
* Allow white space on printed reports for user to write notes
* Use a single typeface, except for emphasis
* Use same family of typefaces within and across displays and reports
* Avoid overly fancy fonts

**Formatting numeric, textual, and alphanumeric date:**

* **Right-justify numeric data** and align columns by decimal points or other delimiter
* **Left-justify textual data.** Use short line length, usually 30–40 characters per line (this is what newspapers use, and it is easier to speed-read).
* Break long sequences of **alphanumeric data** into small groups of three to four characters each

**Contrasting the Display of Tables and Lists (a) Poorly- Designed Form**

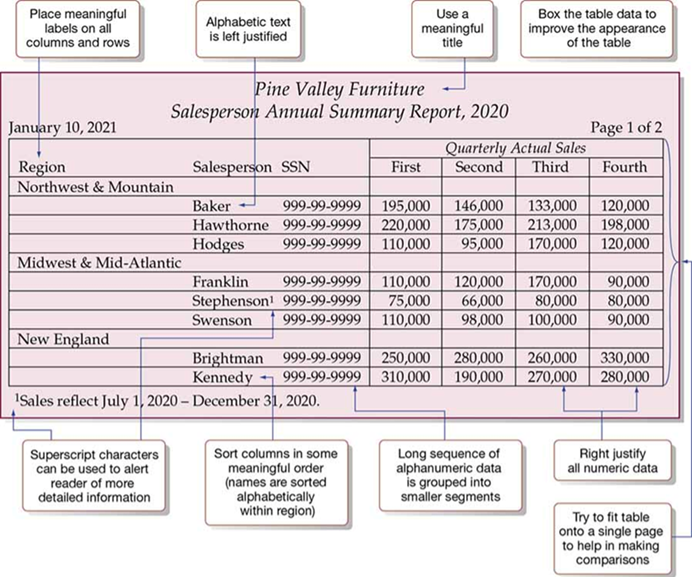
****

****

**Guidelines for Selecting Tables Versus Graphs**

* **Tables**
* Reading individual data values
* **Graphs**
* Providing a quick summary of data
* Detecting trends over time
* Comparing points and patterns of different variables
* Forecasting activities
* Reporting vast amounts of information when relatively simple impressions are to be drawn

**Tabular Report Illustrating Numerous Design Guidelines (Pine Valley Furniture)**



**Graphs for Comparison (a) Line Graph (b) Bar Graph**

**Paper versus Electronic Reports**

* Laser and ink-jet printers can print reports that look like display screen
* Same guidelines apply
* High-speed impact printers are faster, but less user-friendly in appearance
* Good for large batch reports
* The coding sheet from figure 10-2 may be useful for these types of reports

**Assessing Usability**

* **Usability –** overall evaluation of how a system performs in supporting a particular user for a particular task
* **Three characteristics of usability:**

1. Speed – Can you complete a task efficiently?
2. Accuracy – Does the system provide what you expect?
3. Satisfaction – Do you like using the system

**General Design Guidelines for Usability of Forms and Reports**

| Usability Factor | Guidelines for Achievement of Usability |
| --- | --- |
| Consistency | Consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs. Consistent response time each time a function is performed. |
| Organization | Formatting should be designed with an understanding of the task being performed and the intended user. Text and data should be aligned and sorted for efficient navigation and entry. Entry of data should be avoided where possible (e.g., computing rather than entering totals). |
| Clarity | Outputs should be self-explanatory and not require users to remember information from prior outputs in order to complete tasks. Labels should be extensively used, and all scales and units of measure should be clearly indicated |
| Format | Information format should be consistent between entry and display. Format should distinguish each piece of data and highlight, not bury, important data. Special symbols, such as decimal places, dollar signs, and ± signs, should be used as appropriate. |
| Flexibility | Information should be viewed and retrieved in a manner most convenient to the user. For example, users should be given options for the sequence in which to enter or view data and for use of shortcut keystrokes, and the system should remember where the user stopped during the last use of the system. |

**Characteristics for Consideration When Designing Forms and Reports**

| Characteristic | Consideration for Form and Report Design |
| --- | --- |
| User | Issues related to experience, skills, motivation, education, and personality should be considered. |
| Task | Tasks differ in amount of information that must be obtained from or provided to the user. Task demands such as time pressure, cost of errors, and work duration (fatigue) will influence usability. |
| System | The platform on which the system is constructed will influence interaction styles and devices. |
| Environment | Social issues such as the users’ status and role should be considered in addition to environmental concerns such as lighting, sound, task interruptions, temperature, and humidity. The creation of usable forms and reports may necessitate changes in the users’ physical work facilities. |

**Measures of Usability**

**Methods to assess usability:**

**Learnability –** usability dimension concerned with how difficult it is for the user to perform a task for the first time

**Efficiency –** usability dimension concerned with how quickly users can perform tasks once they know how to perform them

**Error rate –** usability dimension concerned with how many errors a user might encounter and how easy it is to recover from those errors

**Memorability—**How easy is it to remember how to accomplish a task when revisiting the system after some period of time?

**Satisfaction and aesthetics—**How enjoyable is the system’s visual appeal and how enjoyable is the system to use?

**Common Errors When Designing the Layout of Web Pages**

| Error | Recommendation |
| --- | --- |
| Nonstandard Use of G U I Widgets | Make sure that when using standard design items, they behave in accordance with major interface design standards. For example, the rules for radio buttons state that they are used to select one item among a set of items, that is, not confirmed until “OK’d” by a user. In many Web sites selecting radio buttons is used as both selection and action. |
| Anything That Looks Like Advertising | Because research on Web traffic has shown that many users have learned to stop paying attention to Web advertisements, make sure that you avoid designing any legitimate information in a manner that resembles advertising (e.g., banners, animations, pop-ups). |
| Bleeding-Edge Technology | Make sure that users don’t need the latest browsers or plug-ins to view your site. |
| Scrolling Test and Looping Animations | Avoid scrolling text and animations because they are both hard to read and users often equate such content with advertising. |
| Nonstandard Link Colors | Avoid using nonstandard colors to show links and for showing links that users have already used; nonstandard colors will confuse the user and reduce ease of use. |
| Outdated Information | Make sure your site is continuously updated so that users “feel” that the site is regularly maintained and updated.  Outdated content is a sure way to lose credibility |
| Slow Download Times | Avoid using large images, lots of images, unnecessary animations, or other time-consuming content that will slow the downloading time of a page. |
| Fixed-Formatted Text | Avoid fixed-formatted text that requires users to scroll horizontally to view content or links. |
| Displaying Long Lists as Long Pages | Avoid requiring users to scroll down a page to view information, especially navigational controls. Manage information by showing only N items at a time, using multiple pages, or by using a scrolling container within the window. |

**Designing Forms and Reports at P V F**

Guidelines established at Pine Valley Furniture (P V F) include:

* Use lightweight graphics
* Establish forms and data integrity rules
* Use stylesheet-based H T M L

**Lightweight graphics –** small, simple images that allow a Web page to be displayed more quickly

Forms and data integrity rules:

All forms that request information should be clearly labeled and provide adequate room for input

Specific fields requiring specific information must provide a clear example

Forms must designate which fields are optional, required, and which have a range of values

**Stylesheet-Based H T M L –** Web design approach that separates content from the way in which it is formatted and presented, making ongoing maintenance easier and site-wide consistency much higher

Allows content of a Web page to remain separate from the way it is formatted

Facilitates site-wide consistency

**QUIZLET**

**mobile-first design**

Designing Web site pages for mobile phones before designing them for larger devices: tablets, laptops, desktops, etc.

**ripple-through effect**

a situation in which changing one item in a project affects - and necessitates the changing of - several other dependent items

**assets**

all of the content (text, images, audio, video, scripts, etc.) that is included in a designed object (Web page, app or game screen, etc.)

**CSS shorthand property**

a CSS property that you can use to specify multiple related properties at the same time

**HTML element box**

The invisible box in which every HTML element resides. CSS lets you control the appearance of HTML element boxes and their contents.

**What are the steps that you should take to create the first page of your site?**

- Revisit and revise the Web site vision statement, strategy, and specification.

- Revisit and revise the word list and mood board to establish the look and feel of your site.

- Gather all the existing assets (text, images, audio, multimedia, etc.) and create all the new assets you need to make the page.

- Build the actual Web page (HTML document). This page will serve as a default page template for your site.

**What are the three most commonly used site structures?**

Hierarchical — Pages are organized into parent and child (sub)sections.

Sequential — Pages are organized into a specific sequence.

Web-like — Pages are connected via a Web of links.

**positional awareness**

An understanding on the part of the user of the position of the current page within the structure of the site.

**What are some techniques you can use to ensure users have strong and accurate positional awareness when browsing your site?**

- include a descriptive heading (or title) on each page

- assign different colors to the main site sections

- assign different background images to different site sections

- include breadcrumb trails

- include a site map

three-click rule

you should try to ensure that users can get the desired information on a Web site in

**three-clicks or less**

obviously, it isn't the end of the world if a user has to click 4 or 5 times, but you should make an effect to minimize the number of clicks a user needs to make to get what they want

**The first step in creating an effective navigation system is to organize your site content in a \_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_\_\_\_ manner.**

logical, usable, consistent

**primary navigation elements**

global - can be accessed form everywhere in the site

**secondary navigation elements**

local - the navigation elements are accessible from the primary navigation element it is related to



**card sorting**

A tool for designing the information architecture of a Web site, in which participants organize topics on cards into categories that make sense to them (usability.gov).

**navigation bars**

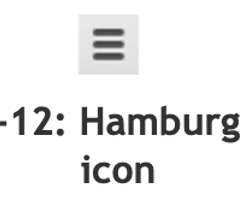
a set of text or image links laid out horizontally or vertically

**drop-down menus**

a set of child sub-items that appears (drops down) when you point the mouse to a parent item in the menu bar

**hamburger menus**

may be used to replace a navigation bar with a drop-down menu highlighted by an icon that looks like the buns and meat of a hamburger



**inline links**

the class text hyperlink embedded within a line of text

**footers**

hold the fine print of a site such as it's copyright, privacy, usage licensing, etc. Also often includes copies of the links that appear in the primary navigation bar

**tag clouds**

a cloud-like assemblage of links whose size (and sometimes color) indicates their relative importance in the site

particularly popular in blogs and social networking sites

**guided navigation**

helps users work through a potentially confusing multistep process

**software wizard**

A program that takes a user through a complex sequence of steps to accomplish a specific goal.

**mystery meat navigation**

A navigation element (link, image, image map, etc.) whose destination is unknown until the user points the mouse at it.

**pagination**

identifies the current page by its numeric position, often users previous and next controls

**What are some disadvantages to using image maps?**

- they can be difficult to navigate for people with disabilities

- large image maps can load slowly

- aren't good for responsive web design

- encourage mystery meat navigation

**What are some CARP and ERBU principals that you can use to make your navigation controls easy to find?**

contrast

proximity

emphasis

**Principals to keep in mind when creating Web navigation**

- design for users (optimize for usability for site visitors)

- make navigation controls visible and easy to find

- provide multiple navigation options

- provide clear positional awareness for users

- be consistent

- follow conventions

- user clear and descriptive names

- make navigation quick and easy

**Considerations when using drop-down menus**

they may be a good choice for larger sites, but they are overkill for smaller websites

can be difficult to use and may also be difficult for search engines to index

**Where should the most important navigation options go?**

At the beginning and end of your navigation bars and menus

reserve the middle area for less important options

**Contextual Inquiry**

A user-centered design method in which information is gathered about users by interviewing them at their place of work.

**user experience design (UXD)**

A design methodology that seeks to optimize the user experience all aspects of the user's interaction with a product or service: usability, accessibility, pleasure, etc.

**user analysis**

The collection and analysis of user data to help understand the characteristics and behaviors of these users.

**Web analytics**

The collection and analysis of Web site data to help understand the characteristics and behaviors of its users.

**What are some methods of gathering user analysis?**

- usability testing

- context interviews (interview at work while they are using the product

- diary studies (keep a diary of thoughts and actions over a period of time)

- ethnographic shadowing (stay with a user for a period of time to learn how they think and act)

- surveys

- user workshops (a group of 5 - 10 people are guided in a discussion of the product)

- feedback forms

**Which produces more accurate data: Web site usability testing or We analytics?**

Web analytics

Web analytics simply observes and records what users do when they are actually visiting the site, on their own, unimpeded by testing procedures or observers

**What are some things that Web analytics can reveal about a site's visitors?**

- conversions

- pages per visit, time per visit

- bounce rate

- exit pages

**conversation**

when a Web site visitor performs a desired action on the site: downloads a file, makes an online purchase, clicks on an ad, etc.

**metric**

the measurement of a specific characteristic of a system's performance or efficiency

**bounce rate**

The percentage of sessions in which users leave a site from their entrance page without having interacted with that page.

**exit pages**

the page from which a user leaves the site

**UI design pattern**

a reusable solution to a common UI design problem

reusable - pattern should be general enough to be applied to variations of the problem

common - if the problem isn't common, it isn't worth creating a new design pattern

Web analytics can find patterns that identify the problems that need to be solved

**Web forms are created with HTML using the \_\_\_\_\_\_\_\_\_\_\_\_ tag.**

<form>

**Data fields within a form are created with the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ HTML tag.**

<input>

**input validation**

the process of inspecting data given to a program by the user and determining if it is valid

**framework**

a set of customizable tools and techniques

Bootstrap and Foundation are both examples of Web frameworks

**Bootstrap**

an open-source framework developed by Twitter that consists of HTML, CSS, and JavaScript code designed to help build user interface components

**Foundation**

an open-source framework maintained by ZURB that is especially well-suited to responsive design.

**Less and Sass**

Style sheet languages that specify how structured documents are presented

**UI library**

A collection of pre-written scripts and code snippets that you can use to create UI elements for apps, Web sites, and applications.

JQuery UI, React, and Kendo UI are examples of UI libraries

**JQuery UI**

an open-source UI library maintained and distributed by the JQuery foundation

consists of GUI widgets, animated visual effects, and themes implemented with JQuery, CSS, and HTML

**React**

an open-sourced UI library maintained by Facebook, Instagram and a community of independent developers

written in JavaScript

enables developers to create UI elements for Web applications simply and effectively

**Kendo UI**

a for-pay JQuery-based UI library that is fast, lightweight and features 70+ UI widgets, Bootstrap support, mobile controls, and offline data processing

**Content Management System (CMS)**

an application that enables developers and end users to create, review, edit, and publish Web site content

Popular CMSs in use today are WordPress, Drupal, and Joomla

**PHP**

A server-side scripting language designed for web development but also used as a general-purpose programming language.

**MySQL**

Relational database management system (RDBMS)

**WordPress**

an open-source CMS based on PHP and MySQL that supports themes, plug-ins, mobile, and other features

**Drupal**

A free and open source Content Management System that is written in PHP.

**front-end**

the visible interface that site visitors directly interact with via UI controls

everything the users see and interact with

**back-end**

hidden from site visitors and performs the behind the scenes which enable the Web site to function properly

**Web development IDE**

An application you can use to build and publish a Web site.

**Telnet client**

a program that uses the Telnet network protocol to enables two devices on the same network to communicate with each other

**SSH client**

a program that uses the secure SSH protocol to enable two devices on the same network to communicate with each other

**Web hosting service**

A company that provides Web servers (Web hosts) on which Web site files can be stored. Also called ISP: Internet service provider.

**shared server hosting**

many different Web sites are hosted on the same server

best for Web site with modest storage, processing and traffic demands

**virtual private server hosting**

multiple Web sites are hosted on the same server, but using different virtual servers

provides root access and more control over server configuration but at a higher cost (compared to shared server hosting)

Like shared server hosting, performance can be affected by other sites on the server

**dedicated server hosting**

Web site is hosted on its own exclusive physical server

This option provides the most control, the highest speed, storage space, and bandwidth, but is also the most expensive option.

This will require more time and expertise to manage.

**cloud hosting**

a set of servers work together to host a group of Web sites

reliable, customizable and able to handle high traffic spikes

You typically won't have root access and substantial technical expertise is needed

**Summative Usability Testing**

done when the Web site is nearly complete

The goal is to ensure that the Web site works as intended and to compare the usability of the Web site to that of its competitors

we are trying to be as objective as possible rather than going by gut feelings like we might have with earlier tests

**What are some things you can test for in a summative usability test?**

- how many clicks and how much time is it taking users to do what they need to do?

- Are users struggling to figure out how to accomplish theirs tasks? (Look for patterns)

- Are users clicking the wrong buttons due to unclear labelling or a difficult to maneuver UI?

- Are the pages taking too long to load?

(There are other things you can test for... this is just intended to give you an idea)

**After administering a summative test, you should..**.

assess the results and write up a list of bugs and problems grouped into priority levels, based on scope and severity

(once that is done, you'll work through the bugs in their priority order)

**code validator**

An application that checks code to ensure it complies with the coding language's rules, syntax and recommended practices.

**What is the most trusted JavaScript validator?**

JSLint

**You should re-validate your Web site code every time \_\_\_\_\_\_\_\_\_\_\_\_\_**

you make changes and once every month or so even if you have not made any changes

**quality assurance**

an umbrella term for a set of procedures (usability testing, code validation, site optimization, etc.) that ensure a site's form and function are of high quality

**Web site operation manual**

A document that describes in detail how to operate, maintain and customize a Web site.

**maintaining the site**

Performing all the tasks necessary to keep a Web site up-to-date and in good working order, so that it displays and works correctly in current browsers and devices.

**site maintenance plan**

Document outlining scheduled dates to update and verify links, content, and functionality of the Web site.

**Search Engine Optimization (SEO)**

The practice of ensuring that site pages acquire and maintain a favorable position in search rankings

**search ranking**

the position at which a site appears in the results listing of a search engine

**Web crawler**

An automated program that gathers data from Web pages.

**SEO keywords**

Specific words in a Web page that make it possible for users to find the page with a search engine.

**keyword suggestion tool**

A tool that expands a keyword to a set of similar keywords.

**What are some keyword-based techniques for increasing your site's search engine ranking?**

- include search keywords in the page title and headings

- include a substantial amount of keyword-related content on the page

- use variations of keywords in the page text

- use the same keywords that similar Web pages use

- use long tail keywords

- use keywords that are in the site URL

**long-tail keywords**

a keyword phrase, usually of 3-5 words

**minify**

Remove all the unnecessary characters from source code without changing its functionality.

**metadata**

a type of data that provides information about other data in a Web page or site

data about data

**XML sitemap**

An XML file that contains metadata about the pages on a site.

**XML file**

a file written in XML, a language used to store, describe and share data via networks such as the Internet

**robots.txt file**

a text file that provides special instructions about a Web site to Web crawlers

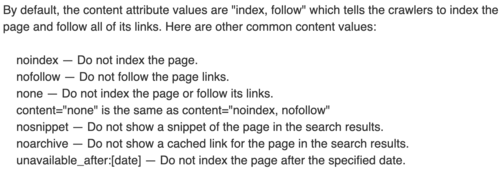
**<meta> tag**

An HTML tag that provides information (metadata) about the Web page in which it resides.

**How can you use the <meta> tag to influence web crawlers?**

<meta name="robots" content="value1, value2, ..." />

- Works with all search engine crawlers while using "googlebot" as the name only works with Google crawlers



**User Interface**

-The aggregate of means by which people—the users—interact with the system

-Provides means of input and output—the

system itself

**Physically, Perceptually, Conceptually**

The Three aspects of of User Interface

**interface, efficient,feedback,queries,productivity**

The Objectives in designing user interface:

• Match the user \_\_\_\_ to the task.

• Make the user interface \_\_\_\_\_

• Provide appropriate \_\_\_\_ to users.

• Generate usable \_\_\_\_\_.

• Improve \_\_\_\_\_ of knowledge workers.

**Physically**

Is an Aspect of User Interface that comprise of the devices the user really touches.

**Perceptually**

Is an Aspect of User Interface that consists of everything the user sees, hears, or touches.

**Conceptually**

Is an Aspect of User Interface that include everything the user knows about using the system.

**Presentation language, Action language**

What are Two main components of User Interface?

**consistency, shortcuts, feedback, error, reversal**

Eight Guidelines for Designing User Interface

•Strive for \_\_\_\_\_\_ .

•Enable frequent users to use \_\_\_\_\_\_.

•Offer informative \_\_\_\_\_\_ .

•Design dialogs to yield closure.

•Offer simple \_\_\_\_\_ handling.

•Permit easy \_\_\_\_\_ of actions.

•Support internal locus of control.

•Reduce short‐term memory load

**Strive for consistency**

It is guideline for Designing User Interface.

It is imperative to design a consistent appearing and functioning interface . The following should be consistent throughout the system

1). The way information is arranged on forms

2). The names and arrangement of menu items

3). The size and shape of items

4). The sequence followed to carry out the task

**Enable frequent users to use shortcuts.**

It is guideline for Designing User Interface.

User used to work one application for the whole day that they quickly loose patience with long menu sequences and multiple dialog boxes when they know exactly what they want to do.

**Offer informative feedback.**

It is guideline for Designing User Interface.

Each action taken by the user should result to some type of feedback from the computer so that the users would know the action was recognized.

**Design dialogs to yield closure.**

It is guideline for Designing User Interface.

Every dialog should be arranged in a clear sequence with a beginning, middle, and end.

**Offer simple error handling.**

It is guideline for Designing User Interface.

User errors are costly, involve the time needed to correct mistakes

**Permit easy reversal of actions.**

It is guideline for Designing User Interface.

Permit easy reversal of actions so users can feel that they can explore options and take action that can be canceled or reversed.

**Support internal locus of control.**

It is guideline for Designing User Interface.

Experience users need to feel that they are in charge of the system and that the system responds to their commands.

**Reduce short‐term memory load.**

It is guideline for Designing User Interface.

People can only remember about seven chunks of information at a time.

**Natural‐Language Interface**

Is a type of User Interface that

• Permits users to interact with the computer in their

everyday or natural language

• No special skills required of the user who interacts

with the computer

• Used by some Web sites

**Question‐and‐Answer Interface**

Is a type of User Interface that

• The computer displays a question to the user on the display

• The user enters an answer (via keyboard stroke or

mouse click), and the computer acts on that input

information

• Dialog box is a type of this interface

• Examples include use of wizard and Office Assistant

in MS applications

**Menu Interface**

Is a type of User Interface that

• Provides the user with an onscreen list of available selections

• Users are limited to the options displayed

• Users need to know what task should be accomplished

• Users should know which task they desire to perform

when utilized

• Can be set up to use keyboard entry, lightpen, or mouse • Consistency in design

**Menu Interface**

Is a type of User Interface that

• GUI menus are used to control PC software and have the following guidelines:

1. The main menu bar is always displayed.

2. The main menu uses single words for menu items.

Main menu options always display secondary pull‐

down menus.

3. The main menu should have secondary options

grouped into similar sets of features.

4. The drop‐down menus that display when a main

menu item is clicked often composed of more than

one word.

5. These secondary options carry out actions or display

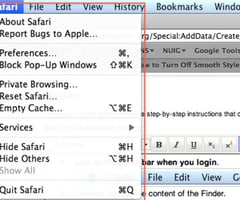
additional menu items

6. Menu items in grey (disabled) are unavailable for the

current activity.

Clicking on a GUI object with the right mouse button, an

object menu (or pop‐up menu) is displayed.



**Form‐Fill Interface**

Is a type of User Interface that

• Consists of onscreen forms or Web‐based forms

displaying fields containing data items or parameters

that need to be communicated to users

• Input/output forms

**Command‐Language Interface**

Is a type of User Interface that

• Allows the user to enter explicit statements to invoke

operations within a system

• Requires users to remember command syntax and

semantics

• Can be a burden to users since they need to

memorize names, syntax, and operations

• Experienced users prefer this type because of the

faster completion time it allows

**Graphical User Interface (GUI)**

Is a type of User Interface that

• Allows users to directly manipulate the graphical

representation on the screen via keyboard input,

joystick, or mouse

• Presents graphical icons, visual indicators, or special

graphical elements called "widgets"

• Continuous feedback on task accomplishment these

provide

• Sets a challenge because an appropriate model of

reality or a satisfactory conceptual model of the

representation should be invented

• When used in on intranets, extranets, or Web, it

requires careful planning

**title, organized, data, scroll, modify**

Effectively designed form contains the following

• A self explanatory \_\_\_ and field headings

• Has fields \_\_\_\_ into logical groupings with distinctive boundaries

• Provides default values when practical

• Displays \_\_\_\_ in appropriate field lengths

• Minimizes the need to \_\_\_ windows

• Input forms for display can be simplified when supplied with default values for fields and allow

users to \_\_\_\_ default information

**Form-Fill Interface**

These are benefits of what User Interface?

-Its benefit is that the printed version of the

filled‐in form provides excellent

documentation

-The main drawback is that experienced users

may become impatient and may want more

efficient ways to enter data

**Guidelines for Designing Dialog**

-Meaningful communication so that the computer understands what people are entering and people understand what the computer is presenting or requesting

- Minimal user action.

1.Entering codes rather than whole words on entry

screens. Codes are also entered when using a

command-language interface.

2. Only entering data that are not already stored on

files.

3. Supplying the editing characters (e.g., slashes as

date field separators

4.Using default values for fields on entry screens.

5. Designing an inquiry (or change or delete) program

so that the user needs to input only the first few

characters of a name or item description.

6. Providing keystrokes for selecting pull‐down menu

options

inputting-is often considered to be the slowest part of a computer system and a good dialog tends to minimize the number of keystrokes required.

**Stylus**

• Pointed stick that looks like a pen

• Becoming popular because of new handwriting

recognition software and PDAs

• Examples include Palm and Pocket/PC devices and



**Tablet PC**

• A notebook computer with a stylus‐sensitive display.

Is much more powerful compared to handheld computers

**Standards to consider in assessing the**

**interfaces chosen**

1. The training period necessary for users should be

acceptably short.

2. Users who are early in their training should be able

to enter commands without thinking about them or

without referring to a help menu or manual.

Keeping the interfaces consistent throughout the

application can be helpful in this regard

3. The interface should be faultless so that errors are

few and those that do occur are not occurring

because of poor design.

4. The time that users and the system need to recover

from errors should be short.

5. Occasional users should be able to study again the

system quickly.

Guidelines for Designing Dialog

**Guidelines for Designing Dialog**

-Standard operation and consistency.

1. Locating titles, date, time, and operator and feedback

messages in the same places on all displays

2. Exiting each program by the same key or menu option

3. Canceling a transaction in a consistent manner, usually

with the use of a function key (F12) on a mainframe and

the ESC key on a PC.

Guidelines for Designing Dialog

**Guidelines for Designing Dialog**

4.Getting help in a standardized manner.

5. Standardizing the colors used for all displays.

6. Standardizing the use of icons for similar operations

when using a GUI.

7. Using consistent terminology in a display screen or Web

site.

8.Providing a consistent way to navigate through the dialog.

9. Using consistent font alignment, size, and color on a

Web page.