



BHARATHIDASAN UNIVERSITY
Tiruchirappalli- 620024,
Tamil Nadu, India

Programme: M.A.,HUMAN RESOURCE MANAGEMENT

**Course Title : HUMAN RESOURCE INFORMATION RESOURCE
SYSTEMS**

Course Code :22HRMEC6

UNIT - I

Introduction to Information Systems

Dr. T. KUMUTHAVALLI

Associate Professor

Department of Lifelong Learning

Establishing the Framework

Understanding Information Systems

Information systems are the backbone of modern organizations, encompassing hardware, software, data, people, and processes to manage and process information.

The Core Components

These systems serve diverse functions including collecting, storing, processing, and disseminating information, enabling organizations to operate efficiently and make informed decisions.

	<p>Value Lizomrens desale of fitully suppliers post in trade.</p> <p>Put peccdively rant wsh outovers pty and prcive regorence.</p>	<p>Channes Dwon do he im peel'outing and b-or partwts to pinnlly.</p> <p>Lesidgn or colbit deais and notere thgy oflocting coocroted opnleatng.</p>	<p>Tomenats Accorpect on fele colbitons ereveal popnes.</p>	<p>Capuctts</p> <ul style="list-style-type: none"> • Bothe now ortony cond l procy will no lser esaffitiane carryon pictuel <p>Betuties</p> <ul style="list-style-type: none"> • Inccrationly pccess with gr'chers. • Dkmeted plantng arealtng or coous. 	<p>Crume rall omolyutiant wrtt vol colng protectecl onwl</p>

Business Models and Information Systems

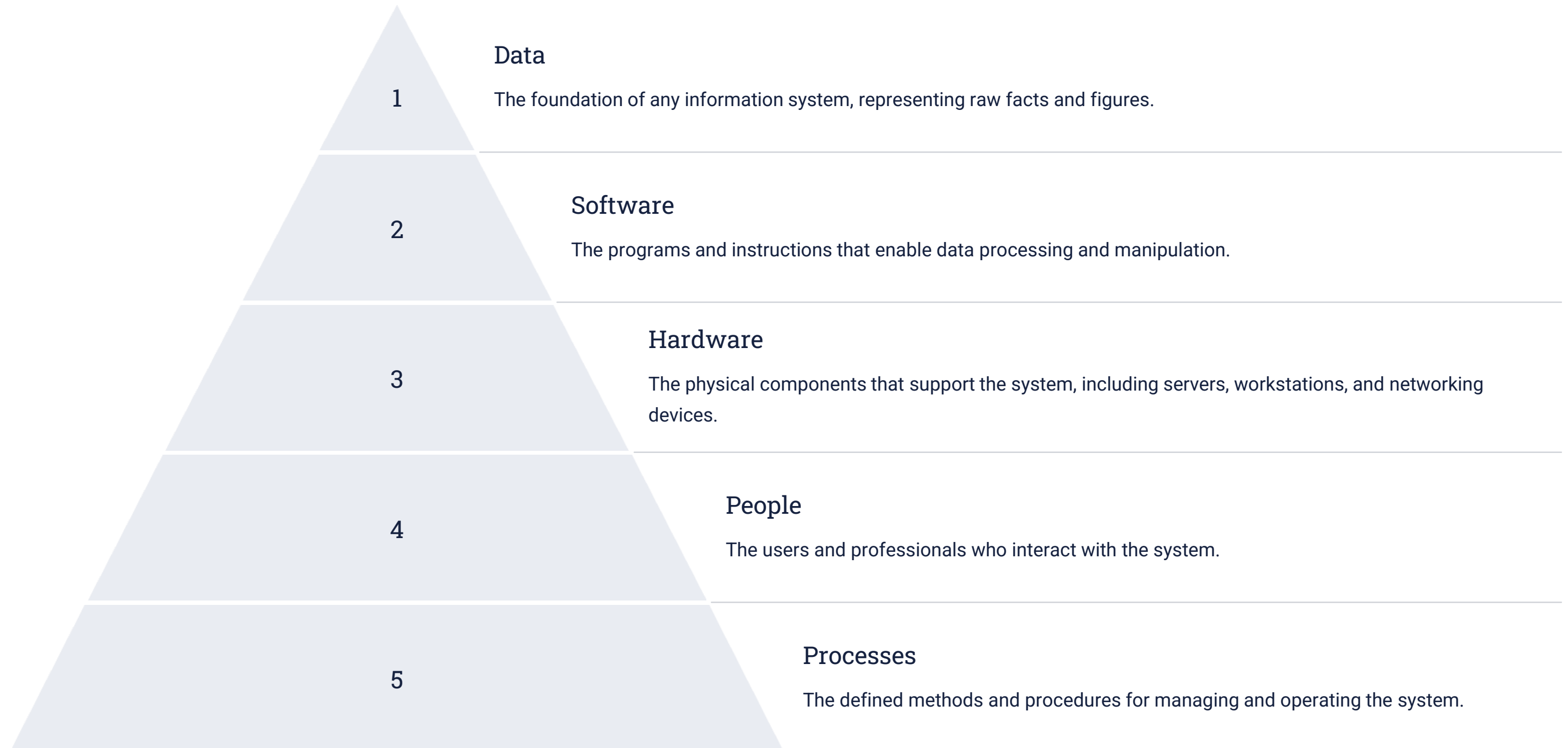
Business Model Canvas

The Business Model Canvas provides a structured framework for understanding how organizations create, deliver, and capture value.

Value Proposition Design

Information systems play a crucial role in supporting various aspects of the business model, including customer relationships, value propositions, and revenue streams.

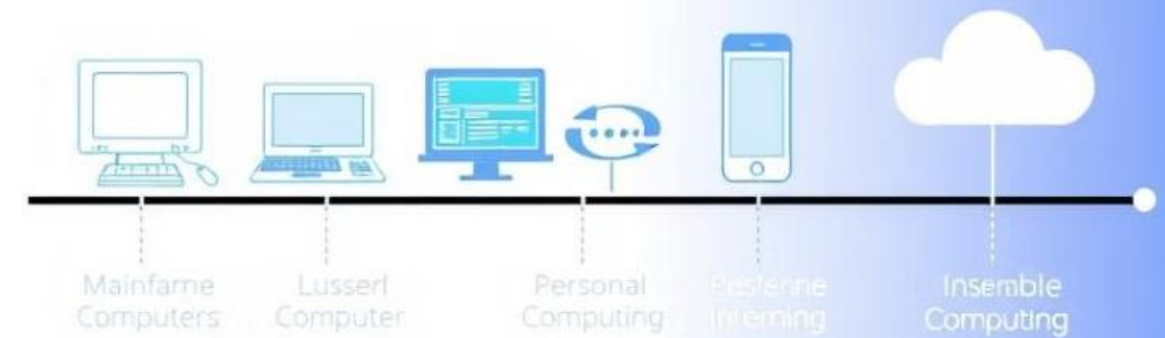
Information System Architecture



Evolution of Information Systems

- 1 Early information systems were primarily manual and paper-based, limiting efficiency and access to information.
- 2 The rise of computers in the 1960s and 1970s led to the development of automated systems, transforming data processing and management.
- 3 The advent of the internet in the 1990s ushered in a new era of interconnectedness, enabling real-time access and global communication.
- 4 Today, information systems are increasingly interconnected, relying on cloud computing, artificial intelligence, and data analytics to drive innovation and growth.

Evolution of Information Systems



System Development: Building Information Systems



1 Requirements Gathering

Defining the needs and expectations of the system users and stakeholders.

2 System Design

Creating a blueprint for the system, specifying its components, interactions, and functionality.

3 Implementation

Building the system using programming languages, databases, and other technologies.

4 Testing and Deployment

Rigorously evaluating the system's functionality and performance before releasing it to users.

Modern Information Systems: Shaping the Digital World



Cloud Computing

On-demand access to computing resources over the internet, enhancing scalability and flexibility.



Artificial Intelligence

Developing systems that can learn, reason, and solve problems, automating tasks and improving efficiency.



Data Analytics

Extracting insights and patterns from data, enabling better decision-making and predictive analysis.



Mobile Technologies

Accessing information and services anytime, anywhere, through mobile devices.



System Development Life Cycle (SDLC)

1

Planning

Defining the project scope, objectives, and resources.

2

Analysis

Gathering and analyzing user requirements to understand the system's needs.

3

Design

Developing the system architecture, data models, and interfaces.

4

Implementation

Building and coding the system using programming languages and tools.

5

Testing

Validating the system's functionality and performance before deployment.



Structured Methodologies: Designing Efficient Systems

1

Structured Programming

A disciplined approach to coding, emphasizing modularity, readability, and maintainability.

2

Data Modeling

Defining the structure and relationships of data within the system, ensuring data integrity and consistency.

3

Process Modeling

Visualizing and documenting the system's workflows and business processes.

Designing Computer-Based Methods:

Computer-based methods have revolutionized HR management, offering efficiency, accuracy, and valuable insights.

Here's a breakdown of key design considerations:

1. Core HR Functions:

Recruitment:

- **Applicant Tracking Systems (ATS):** Streamline job postings, applications, and screening.
- **Video Interviewing:** Conduct initial interviews remotely.
- **Skills Testing:** Automate assessments for specific roles.

Onboarding:

- **Digital Onboarding Portals:** Provide access to documents, training, and company culture.
- **Automated Welcome Messages:** Personalize new hire experiences.

Payroll and Benefits:

- **Payroll Processing Software:** Ensure timely and accurate payments.
- **Benefit Enrollment Systems:** Allow employees to choose and manage benefits.

Performance Management:

- **Performance Review Software:** Facilitate goal setting, reviews, and feedback.
- **360-Degree Feedback Tools:** Gather input from multiple sources.

Learning and Development:

- **Learning Management Systems (LMS):** Deliver training courses, track progress, and manage certifications.
- **Virtual Training Platforms:** Offer online courses and webinars.

2. Key Design Principles:

- **User-Centric Design:** Focus on the needs and preferences of HR professionals and employees.
- **Data Security:** Implement robust security measures to protect sensitive information.
- **Integration:** Ensure seamless integration with other HR systems and business applications.
- **Scalability:** Design systems that can grow with the organization's needs.
- **Mobile Accessibility:** Provide access to HR functions through mobile devices.
- **Data Analytics:** Leverage data to make informed decisions and improve HR practices.

3. Implementation Steps:

- **Needs Assessment:** Identify specific HR challenges and goals.
- **System Selection:** Choose the right HRIS software or build a custom solution.
- **Data Migration:** Transfer existing HR data into the new system.
- **User Training:** Educate HR staff and employees on how to use the system.
- **Testing and Rollout:** Conduct thorough testing before full implementation.
- **Ongoing Support and Maintenance:** Provide regular updates and technical support.

4. Benefits of Computer-Based HRIS:

- **Increased Efficiency:** Automate time-consuming tasks.
- **Improved Accuracy:** Reduce errors and ensure data consistency.
- **Enhanced Decision-Making:** Gain valuable insights through data analysis.
- **Better Employee Engagement:** Provide self-service options and improve communication.

5. Emerging Trends:

- **Artificial Intelligence (AI) and Machine Learning:** Automate tasks, personalize experiences, and predict future needs.
- **Cloud Computing:** Enhance scalability, flexibility, and cost-effectiveness.
- **Blockchain Technology:** Improve data security and transparency.
- **Virtual Reality (VR) and Augmented Reality (AR):** Enhance training and development experiences.

By carefully considering these factors, organizations can design and implement effective computer-based HRIS solutions that drive efficiency, improve decision-making, and enhance the overall employee experience.

Designing Computer-Based Procedures:

Computer-based procedures are step-by-step instructions designed to guide users through specific tasks or processes using computer systems. They are essential for ensuring consistency, accuracy, and efficiency in various fields, from software development to business operations.

Key Principles for Effective Design

1. User-Centric Approach:

1. **Understand Your Audience:** Tailor the procedures to the specific knowledge and experience level of your target users.
2. **Logical Flow:** Organize the steps in a logical sequence that is easy to follow.
3. **Visual Aids:** Incorporate diagrams, flowcharts, and screenshots to enhance understanding.

2. Comprehensive and Detailed Instructions:

- 1. Step-by-Step Guidance:** Break down complex tasks into smaller, manageable steps.
- 2. Decision Points:** Clearly indicate branching paths or alternative actions based on specific conditions.
- 3. Error Handling:** Provide instructions for common errors or unexpected situations.
- 4. Testing and Validation:** Thoroughly test the procedures to ensure they are accurate and effective.

3. Accessibility and Maintainability:

- 1. Consistent Formatting:** Use a consistent style and format for easy readability.
- 2. Version Control:** Maintain a history of changes to track updates and revisions.
- 3. Accessibility Features:** Consider the needs of users with disabilities by providing alternative formats or assistive technologies.

Tools and Techniques

- **Documentation Software:** Tools like Microsoft Word, Google Docs, or specialized documentation software can help create and format procedures.
- **Diagramming Tools:** Tools like Lucidchart, Draw.io, or Visio can be used to create flowcharts and diagrams.
- **Screen Recording Software:** Tools like Camtasia or Snagit can be used to create video tutorials or capture screenshots.
- **Version Control Systems:** Tools like Git can be used to track changes and collaborate on procedures.

Designing Structured Programs:

Structured programming is a programming paradigm that emphasizes the disciplined use of control flow constructs to improve the clarity, quality, and maintainability of computer programs. It is based on three fundamental control flow constructs:

- **Sequence:** A linear execution of statements, one after another.
- **Selection:** Choosing between different paths based on a condition (e.g., if-else statements).
- **Repetition:** Repeating a block of code multiple times (e.g., while and for loops).

Key Principles of Structured Programming

1. **Top-Down Design:** Breaking down a large problem into smaller, more manageable subproblems.
2. **Modularity:** Dividing a program into smaller, independent modules or functions.
3. **Abstraction:** Hiding unnecessary details and focusing on the essential aspects of a problem.
4. **Data Encapsulation:** Protecting data from unauthorized access or modification.

Steps in Designing a Structured Program

1. **Problem Definition:** Clearly define the problem to be solved.
2. **Algorithm Design:** Develop a step-by-step procedure to solve the problem.
3. **Modularization:** Divide the algorithm into smaller, independent modules.
4. **Coding:** Implement the modules using a structured programming language.
5. **Testing and Debugging:** Test the program thoroughly to identify and fix errors.
6. **Maintenance:** Maintain and update the program as needed.

Steps in Designing a Structured Program

1. **Problem Definition:** Clearly define the problem to be solved.
2. **Algorithm Design:** Develop a step-by-step procedure to solve the problem.
3. **Modularization:** Divide the algorithm into smaller, independent modules.
4. **Coding:** Implement the modules using a structured programming language.
5. **Testing and Debugging:** Test the program thoroughly to identify and fix errors.
6. **Maintenance:** Maintain and update the program as needed.

