

ROBUST SYSTEMS DESIGN

Systems Analysis & Design

Author: Eng. Carlos Andrés Sierra, M.Sc.
cavirguezs@udistrital.edu.co

Full-time Adjunct Professor
Computer Engineering Program
School of Engineering
Universidad Distrital Francisco José de Caldas

2025-III



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Outline

- 1 Concepts Generation & Selection
- 2 Quality Guidelines in Systems Design
- 3 Systems Architectures



Outline

- 1 Concepts Generation & Selection
- 2 Quality Guidelines in Systems Design
- 3 Systems Architectures



Concepts Generation

- **Concepts generation** is the **process** of **creating ideas** for a **system** that **meet** the **needs** of its **users**.
- It involves **brainstorming**, **research**, and **analysis** to generate **innovative ideas** for a **system**.
- It is a **creative process** that **encourages innovation** and **creativity** in the **design** of a **system**.



Concepts Generation

- **Concepts generation** is the process of creating ideas for a system that meet the needs of its users.
- It involves brainstorming, research, and analysis to generate innovative ideas for a system.
- It is a creative process that encourages innovation and creativity in the design of a system.



Concepts Generation

- **Concepts generation** is the process of creating ideas for a system that meet the needs of its users.
- It involves brainstorming, research, and analysis to generate innovative ideas for a system.
- It is a creative process that encourages innovation and creativity in the design of a system.



Innovation and Creativity

- **Innovation** is the process of creating new ideas and solutions that improve the performance of a system.
- **Creativity** is the ability to generate original and innovative ideas that solve problems and meet the needs of users.
- They are important for ensuring that a system is robust, efficient, and effective.



Innovation and Creativity

- **Innovation** is the process of creating new ideas and solutions that improve the performance of a system.
- **Creativity** is the ability to generate original and innovative ideas that solve problems and meet the needs of users.
- They are important for ensuring that a system is robust, efficient, and effective.



Innovation and Creativity

- **Innovation** is the process of creating new ideas and solutions that improve the performance of a system.
- **Creativity** is the ability to generate original and innovative ideas that solve problems and meet the needs of users.
- They are important for ensuring that a system is robust, efficient, and effective.



Is this Innovation & Creativity?

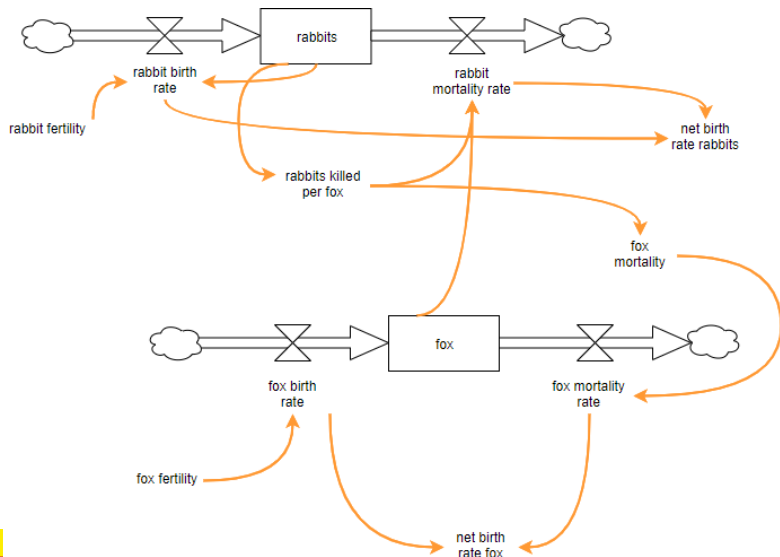


Concepts Selection

- **Concepts selection** is the process of evaluating and choosing the best ideas for a system.
- It involves analysis, comparison, and evaluation of concepts to determine which ones are the most feasible and effective.



Stock and Flow Diagram



Outline

- 1 Concepts Generation & Selection
- 2 Quality Guidelines in Systems Design
- 3 Systems Architectures



Quality Guidelines

- **Quality guidelines** are *principles* that *guide* the **design** of a **system** to *ensure* that it **meets** the **needs** of its **users**.
- They include **reliability**, **scalability**, **maintainability**, and **usability guidelines**.
- They are *important* for *ensuring* that a **system** is **robust**, **efficient**, and **effective**.



Reliability Guidelines

- **Reliability guidelines** are *principles* that *guide* the **design** of a **system** to *ensure* that it is **reliable** and **dependable**.
- They include **fault-tolerance**, **redundancy**, and **error-handling guidelines**.
- They are *important* for *ensuring* that a **system** is **robust** and **resilient** to **failures**.



Scalability Guidelines

- **Scalability guidelines** are *principles* that *guide* the **design** of a **system** to *ensure* that it is **scalable** and **flexible**.
- They include **modularity**, **extensibility**, and **performance guidelines**.
- They are *important* for *ensuring* that a **system** can **grow** and **adapt** to **changing requirements**.



Maintainability Guidelines

- **Maintainability guidelines** are *principles* that *guide* the **design** of a **system** to *ensure* that it is **easy** to **maintain** and **update**.
- They include **modularity**, **documentation**, and **versioning guidelines**.
- They are *important* for *ensuring* that a **system** can be **easily maintained** and **updated** by its **developers**.



Quality Standards

- **Quality standards** are **benchmarks** that *define* the **level** of **quality** that a **system** must **meet**.
- They include **ISO 9000**, **CMMI**, and **Six Sigma** standards.



ISO 9000

- **ISO 9000** is a quality standard that *defines* the **requirements** for a **quality management system**.
- It is *designed* to help **organizations** ensure that they **meet** the **needs** of their **customers** and **stakeholders**.
- It is *based* on a number of quality management principles, including **customer focus**, **leadership**, and **continuous improvement**.



ISO 27001

- **ISO 27001** is a quality standard that *defines* the **requirements** for an **information security management system**.
- It is *designed* to help **organizations** protect their **information** and **ensure** that it is **secure** and **confidential**.
- It is *based* on a number of information security management principles, including **risk assessment**, **security policies**, and **incident response**.



CMMI

- **CMMI** is a quality standard that *defines* the **requirements** for a mature software development process.
- It is *designed* to help organizations improve their software development processes and deliver high-quality products to their customers.
- It is *based* on a number of best practices for software development, including requirements management, project planning, and process monitoring.



Six Sigma

- **Six Sigma** is a quality standard that *defines* the **requirements** for a **process** that is *capable* of **producing high-quality products**.
- It is *designed* to help **organizations** improve their **processes** and **reduce defects** in their **products** and **services**.
- It is *based* on a number of quality management principles, including **data-driven decision-making**, **process improvement**, and **customer focus**.



Outline

- 1 Concepts Generation & Selection
- 2 Quality Guidelines in Systems Design
- 3 Systems Architectures



What is a System Architecture?

- A **system architecture** is the **structure** of a **system** that **defines** its **components**, **interactions**, and **relationships**.
- A **system architecture** is the **blueprint** of a **system** that **guides** its **development** and **implementation**.
- A **system architecture** is the **foundation** of a **system** that **ensures** that it **meets** the **needs** of its **users**.



What is a System Architecture?

- A **system architecture** is the **structure** of a **system** that **defines** its **components**, **interactions**, and **relationships**.
- A **system architecture** is the **blueprint** of a **system** that **guides** its **development** and **implementation**.
- A **system architecture** is the foundation of a system that ensures that it **meets** the **needs** of its **users**.



What is a System Architecture?

- A **system architecture** is the **structure** of a **system** that **defines** its **components**, **interactions**, and **relationships**.
- A **system architecture** is the **blueprint** of a **system** that **guides** its **development** and **implementation**.
- A **system architecture** is the **foundation** of a **system** that **ensures** that it **meets** the **needs** of its **users**.



Types of System Architectures

- There are several types of system architectures that are used in systems development.
- They include monolithic, client-server, peer-to-peer, and distributed architectures.
- Each type of architecture has its own advantages and disadvantages that depend on the specific requirements of the system.



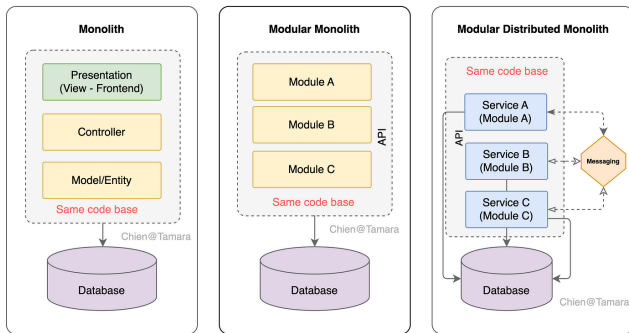
Types of System Architectures

- There are several types of system architectures that are used in systems development.
- They include monolithic, client-server, peer-to-peer, and distributed architectures.
- Each type of architecture has its own advantages and disadvantages that depend on the specific requirements of the system.



Monolithic System Architecture

- A **monolithic system architecture** is a **single-tier architecture** that *consists* of a **single unit** that *performs* all the **functions** of the **system**.
- It is **simple**, **easy** to **develop**, and **maintain**, but it is **not scalable** and **flexible**. It is *used* for **small systems** that **do not require high performance** or **reliability**.



Client-Server System Architecture

- A **client-server system architecture** is a **two-tier architecture** that *consists* of a **client** and a **server** that *communicate* with each other over a **network**.
- It is **scalable**, **flexible**, and **efficient**, but it is *complex* and *difficult* to **develop** and **maintain**. It is *used* in **medium** to **large systems** that require high performance and **reliability**.



Peer-to-Peer System Architecture

- A **peer-to-peer system architecture** is a **two-tier architecture** that *consists* of a **network** of **peers** that *communicate* with each other directly.
- It is **scalable**, **flexible**, and **efficient**, but it is *complex* and *difficult* to **develop** and **maintain**. It is *used* in **medium** to **large systems** that require **high performance** and **reliability**.



Distributed System Architecture

- A **distributed system architecture** is a **multi-tier architecture** that *consists* of a **network** of **nodes** that *communicate* with each other over a **network**.
- It is **scalable**, **flexible**, and **efficient**, but it is **complex** and **difficult** to **develop** and **maintain**. It is *used* in **large systems** that **require** high **performance** and **reliability**.



Outline

- 1 Concepts Generation & Selection
- 2 Quality Guidelines in Systems Design
- 3 Systems Architectures



Thanks!

Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/systems-analysis>

