Business Systems

Systems Analysis

Author: Eng. Carlos Andrés Sierra, M.Sc.
carlos.andres.sierra.v@gmail.com

Lecturer
Computer Engineer
School of Engineering
Universidad Distrital Francisco José de Caldas

2024-I

Outline

1. Domain-Driven Design
2. Business Systems Analysis
3. Software Methodologies
4. Requirements Engineering

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Outline

1. Domain-Driven Design

2. Business Systems Analysis

3. Software Methodologies

4. Requirements Engineering

Basics of Domain-Driven Design I

- DDD is focusing on the **core domain** and **domain logic**, it is a way of thinking aimed at accelerating software projects that have to **deal with complicated domains**.
- The essential **terms** of DDD are *context*, *model*, *ubiquitous language*, *bounded context*, and *business logic in layers*.
- DDD is a set of **principles** and patterns that help to design a system ensuring alignment with the **real-world business needs**.

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

Basics of Domain-Driven Design I

- DDD is focusing on the **core domain** and **domain logic**, it is a way of thinking aimed at accelerating software projects that have to **deal with complicated domains**.
- The essential **terms** of DDD are *context*, *model*, *ubiquitous language*, *bounded context*, and *business logic in layers*.
- DDD is a set of **principles** and patterns that help to design a system ensuring alignment with the **real-world business needs**.

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

Basics of Domain-Driven Design I

- DDD is focusing on the **core domain** and **domain logic**, it is a way of thinking aimed at accelerating software projects that have to **deal with complicated domains**.
- The essential **terms** of DDD are *context*, *model*, *ubiquitous language*, *bounded context*, and *business logic in layers*.
- DDD is a set of **principles** and patterns that help to design a system ensuring alignment with the **real-world business needs**.

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
    - Focus on the core domain.
    - Base complex designs on models of the domain.
    - Constantly collaborate with domain experts.
    - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
    - Domain Layer.
    - Application Layer.
    - Presentation Layer.
    - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Basics of Domain-Driven Design II

Figure: Prompt: Draw a soccer coach teaching robots soccer players.

- The main principles of DDD are:
  - Focus on the core domain.
  - Base complex designs on models of the domain.
  - Constantly collaborate with domain experts.
  - Develop a knowledge-rich model.
- The business logic in layers is showed as follows:
  - Domain Layer.
  - Application Layer.
  - Presentation Layer.
  - Infrasctructure Layer.

Outline

1 Domain-Driven Design

2 Business Systems Analysis

3 Software Methodologies

4 Requirements Engineering

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more...
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more. . .
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more. . .
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more. . .
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more. . .
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more...
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Systems

- Business systems are complex systems that support the **operations and processes** of a business.
- They are designed to automate and streamline business processes, improve efficiency, and provide accurate and timely information for **decision-making**.
- Business systems can include a wide range of components, such as:
  - Enterprise resource planning (ERP) systems
  - Customer relationship management (CRM) systems
  - Supply chain management (SCM) systems
  - Business intelligence (BI) systems
  - E-commerce systems
  - And many more...
- Effective business systems analysis is crucial for understanding and improving business processes, identifying system requirements, and ensuring successful system implementation.

Business Analysis

- Business analysis is the process of identifying business needs and determining **solutions** to *business problems*.
- It involves understanding the current state of the business, analyzing requirements, and recommending improvements.
- Business analysts use various **techniques** and tools to gather and document requirements, such as `interviews`, `surveys`, and `workshops`.
- The goal of business analysis is to align business objectives with IT solutions and ensure that the resulting **systems meet** the needs of the business.
- Key activities in business analysis include:
  - Understanding business processes and workflows.
  - Identifying and documenting business requirements.
  - Analyzing and prioritizing requirements.
  - Creating business process models and diagrams.
  - Collaborating with stakeholders to validate requirements.

Business Analysis

- Business analysis is the process of identifying business needs and determining **solutions** to *business problems*.
- It involves understanding the current state of the business, analyzing requirements, and recommending improvements.
- Business analysts use various **techniques** and tools to gather and document requirements, such as `interviews`, `surveys`, and `workshops`.
- The goal of business analysis is to align business objectives with IT solutions and ensure that the resulting **systems meet** the needs of the business.
- Key activities in business analysis include:
  - Understanding business processes and workflows
  - Identifying and documenting business requirements
  - Analyzing and prioritizing requirements
  - Creating business process models and diagrams
  - Collaborating with stakeholders to validate requirements

Business Analysis

- Business analysis is the process of identifying business needs and determining **solutions** to *business problems*.
- It involves understanding the current state of the business, analyzing requirements, and recommending improvements.
- Business analysts use various **techniques** and tools to gather and document requirements, such as `interviews`, `surveys`, and `workshops`.
- The goal of business analysis is to align business objectives with IT solutions and ensure that the resulting **systems meet** the needs of the business.
- Key activities in business analysis include:
  - Understanding business processes and workflows
  - Identifying and documenting business requirements
  - Analyzing and prioritizing requirements
  - Creating business process models and diagrams
  - Collaborating with stakeholders to validate requirements

Business Analysis

- Business analysis is the process of identifying business needs and determining **solutions** to *business problems*.
- It involves understanding the current state of the business, analyzing requirements, and recommending improvements.
- Business analysts use various **techniques** and tools to gather and document requirements, such as `interviews`, `surveys`, and `workshops`.
- The goal of business analysis is to align business objectives with IT solutions and ensure that the resulting **systems meet** the needs of the business.
- Key activities in business analysis include:
  - Understanding business processes and workflows
  - Identifying and documenting business requirements
  - Analyzing and prioritizing requirements
  - Creating business process models and diagrams
  - Collaborating with stakeholders to validate requirements

Business Analysis

- Business analysis is the process of identifying business needs and determining **solutions** to *business problems*.
- It involves understanding the current state of the business, analyzing requirements, and recommending improvements.
- Business analysts use various **techniques** and tools to gather and document requirements, such as `interviews`, `surveys`, and `workshops`.
- The goal of business analysis is to align business objectives with IT solutions and ensure that the resulting **systems meet** the needs of the business.
- Key activities in business analysis include:
  - Understanding business processes and workflows
  - Identifying and documenting business requirements
  - Analyzing and prioritizing requirements
  - Creating business process models and diagrams
  - Collaborating with stakeholders to validate requirements

IT Project Management

- IT Project Management is the process of planning, organizing, and controlling the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
    - Project initiation and planning
    - Project execution and control
    - Project closure and evaluation
- Common IT Project Management methodologies include:
    - Waterfall
    - Agile
    - Scrum
    - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning`, `organizing`, `and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

IT Project Management

- IT Project Management is the process of `planning, organizing, and controlling` the resources and activities required to complete an **IT project**.
- It involves defining project goals, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
  - Project initiation and planning
  - Project execution and control
  - Project closure and evaluation
- Common IT Project Management methodologies include:
  - Waterfall
  - Agile
  - Scrum
  - Kanban

Outline

1. Domain-Driven Design

2. Business Systems Analysis

3. Software Methodologies

4. Requirements Engineering

Software Methodologies

- Methodologies are a **set of steps** to complete tasks or to perform processes.
- They are next level of algorithms, more complex, more detailed.

Traditional Methodologies

- At the beginning, methodologies to solve software problems were **big**, tons of steps, documentation of decisions taken, and looking for a lot of explanations for everything.
- Some problems required *old school methodologies* to be solved. However, there are **just a few** cases of them.
- Big methodologies required a lot of resources, as `humans`, `knowledge`, `time` and `money`. Sadly, in real-world, you rarely have all those things to solve problems.

Agile Methodologies

- Agile methodologies were created and developed for **small technology companies** unconsciously around thirty years ago.
- Some technology companies are tricky: start with `small teams`, with a `few of money`, but with big `potential growth`.
- The term startups group this kind of companies. If you want to develop a product with small teams and no so much budget, you need to think smart and think fast.
- Agile methodologies focus on **final product** more than in **processes** and **documentation**.
- It means, have a good leadership, a good team culture, a good learning curve, share knowledge, make the client a strong part of the process, and have quickly new versions of the product.

Agile Methodologies

- Agile methodologies were created and developed for **small technology companies** unconsciously around thirty years ago.
- Some technology companies are tricky: start with small teams, with a few of money, but with big potential growth.
- The term startups group this kind of companies. If you want to develop a product with small teams and no so much budget, you need to think smart and think fast.
- Agile methodologies focus on **final product** more than in **processes** and **documentation**.
- It means, have a good leadership, a good team culture, a good learning curve, share knowledge, make the client a strong part of the process, and have quickly new versions of the product.

Agile Methodologies

- Agile methodologies were created and developed for **small technology companies** unconsciously around thirty years ago.
- Some technology companies are tricky: start with `small teams`, with a `few of money`, but with big `potential growth`.
- The term startups group this kind of companies. If you want to develop a product with small teams and no so much budget, you need to think smart and think fast.
- Agile methodologies focus on **final product** more than in **processes** and **documentation**.
- It means, have a good leadership, a good team culture, a good learning curve, share knowledge, make the client a strong part of the process, and have quickly new versions of the product.

Agile Methodologies

- Agile methodologies were created and developed for **small technology companies** unconsciously around thirty years ago.
- Some technology companies are tricky: start with `small teams`, with a `few of money`, but with big `potential growth`.
- The term startups group this kind of companies. If you want to develop a product with small teams and no so much budget, you need to think smart and think fast.
- Agile methodologies focus on **final product** more than in **processes** and **documentation**.
- It means, have a good leadership, a good team culture, a good learning curve, share knowledge, make the client a strong part of the process, and have quickly new versions of the product.

Agile Methodologies

- Agile methodologies were created and developed for **small technology companies** unconsciously around thirty years ago.
- Some technology companies are tricky: start with `small teams`, with a `few of money`, but with big `potential growth`.
- The term startups group this kind of companies. If you want to develop a product with small teams and no so much budget, you need to think smart and think fast.
- Agile methodologies focus on **final product** more than in **processes** and **documentation**.
- It means, have a good leadership, a good team culture, a good learning curve, share knowledge, make the client a strong part of the process, and have quickly new versions of the product.

Study Case: Scrum

Outline

1. Domain-Driven Design
2. Business Systems Analysis
3. Software Methodologies
4. Requirements Engineering

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
    - Eliciting requirements from stakeholders
    - Analyzing and documenting requirements
    - Validating requirements with stakeholders
    - Managing changes to requirements

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
  - Eliciting requirements from stakeholders
  - Analyzing and documenting requirements
  - Validating requirements with stakeholders
  - Managing changes to requirements

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
  - Eliciting requirements from stakeholders
  - Analyzing and documenting requirements
  - Validating requirements with stakeholders
  - Managing changes to requirements

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
  - Eliciting requirements from stakeholders
  - Analyzing and documenting requirements
  - Validating requirements with stakeholders
  - Managing changes to requirements

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
    - Eliciting requirements from stakeholders
    - Analyzing and documenting requirements
    - Validating requirements with stakeholders
    - Managing changes to requirements

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
  - Eliciting requirements from stakeholders
  - Analyzing and documenting requirements
  - Validating requirements with stakeholders
  - Managing changes to requirements

Requirements Engineering I

- Requirements engineering is the process of **gathering, documenting, and managing** the requirements for a software project.
- It involves understanding the needs of the **stakeholders**, defining the **scope** of the project, and creating a **shared understanding** of the requirements.
- Requirements engineering is critical to the success of a software project, as it helps to **ensure** that the resulting system meets the needs of the **users** and **stakeholders**.
- Key activities in requirements engineering include:
  - Eliciting requirements from stakeholders
  - Analyzing and documenting requirements
  - Validating requirements with stakeholders
  - Managing changes to requirements

Requirements Engineering II

- Requirements engineering is the set of **processes and tools** to gather a project requirements in the best possible way.
- It is normal the clients do not know what they want. They are confused, or with **wrong expectations**, even with **bad understanding** of the problem.
- Gather right information means make the right questions. At the same time, it is important to understand **business domain**, define business rules, and create the `right shared vocabulary`.
- Always be honest, understand client's expectations, and define processes align with those **expectations**.
- Effective requirements engineering requires good communication with stakeholders, a clear understanding of the business domain, and the ability to translate business needs into technical requirements.

Requirements Engineering II

- Requirements engineering is the set of **processes and tools** to gather a project requirements in the best possible way.
- It is normal the clients do not know what they want. They are confused, or with **wrong expectations**, even with **bad understanding** of the problem.
- Gather right information means make the right questions. At the same time, it is important to understand **business domain**, define business rules, and create the `right shared vocabulary`.
- Always be honest, understand client's expectations, and define processes align with those **expectations**.
- Effective requirements engineering requires good communication with stakeholders, a clear understanding of the business domain, and the ability to translate business needs into technical requirements.

Requirements Engineering II

- Requirements engineering is the set of **processes and tools** to gather a project requirements in the best possible way.
- It is normal the clients do not know what they want. They are confused, or with **wrong expectations**, even with **bad understanding** of the problem.
- Gather right information means make the right questions. At the same time, it is important to understand **business domain**, define business rules, and create the `right shared vocabulary`.
- Always be honest, understand client's expectations, and define processes align with those **expectations**.
- Effective requirements engineering requires good communication with stakeholders, a clear understanding of the business domain, and the ability to translate business needs into technical requirements.

Requirements Engineering II

- Requirements engineering is the set of **processes and tools** to gather a project requirements in the best possible way.
- It is normal the clients do not know what they want. They are confused, or with **wrong expectations**, even with **bad understanding** of the problem.
- Gather right information means make the right questions. At the same time, it is important to understand **business domain**, define business rules, and create the `right shared vocabulary`.
- Always be honest, understand client's expectations, and define processes align with those **expectations**.
- Effective requirements engineering requires good communication with stakeholders, a clear understanding of the business domain, and the ability to translate business needs into technical requirements.

Requirements Engineering II

- Requirements engineering is the set of **processes and tools** to gather a project requirements in the best possible way.
- It is normal the clients do not know what they want. They are confused, or with **wrong expectations**, even with **bad understanding** of the problem.
- Gather right information means make the right questions. At the same time, it is important to understand **business domain**, define business rules, and create the `right shared vocabulary`.
- Always be honest, understand client's expectations, and define processes align with those **expectations**.
- Effective requirements engineering requires good communication with stakeholders, a clear understanding of the business domain, and the ability to translate business needs into technical requirements.

Requirements Engineering III

- The classical theory talks about **two kind** of requirements: functional and non-functional. However, they are described in a very technical way.
- There are strategies to get information for clientes: interviews, brainstorming sessions, analysis current client process documentation. All depends of the nature of the problem, and the impact of that one into the organization.
- Sometimes a coffee conversation, a game with lego pieces, some theory of games, could bring you more useful information about the problem and the expected solution.

Requirements Engineering III

- The classical theory talks about **two kind** of requirements: functional and non-functional. However, they are described in a very technical way.
- There are strategies to get information for clientes: `interviews`, `brainstorming sessions`, analysis current `client process documentation`. All depends of the nature of the problem, and the impact of that one into the organization.
- Sometimes a coffee conversation, a game with lego pieces, some theory of games, could bring you more useful information about the `problem and the expected solution`.

Requirements Engineering III

- The classical theory talks about **two kind** of requirements: functional and non-functional. However, they are described in a very technical way.
- There are strategies to get information for clientes: `interviews`, `brainstorming sessions`, analysis current `client process documentation`. All depends of the nature of the problem, and the impact of that one into the organization.
- Sometimes a coffee conversation, a game with lego pieces, some theory of games, could bring you more useful information about the `problem and the expected solution`.

User Stories

- To communicate in the same vocabulary with the client **decreases** the **misunderstandings**, avoid confusion.
- Nowadays user stories are a wide accepted approach, a the idea is to define requirements as a **client story**, it means write the requirement in the **client vocabulary**.
- Once you have the user story, you could **split it into technical tasks** to complete it. It increases communication capabilities, and better validation stages.

User Stories

- To communicate in the same vocabulary with the client **decreases** the **misunderstandings**, avoid confusion.
- Nowadays user stories are a wide accepted approach, a the idea is to define requirements as a **client story**, it means write the requirement in the **client vocabulary**.
- Once you have the user story, you could **split it into technical tasks** to complete it. It increases communication capabilities, and better validation stages.

User Stories

- To communicate in the same vocabulary with the client **decreases** the **misunderstandings**, avoid confusion.
- Nowadays user stories are a wide accepted approach, a the idea is to define requirements as a **client story**, it means write the requirement in the **client vocabulary**.
- Once you have the user story, you could **split it into technical tasks** to complete it. It increases communication capabilities, and better validation stages.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more.
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more...
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more. . .
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more. . .
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more. . .
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more. . .
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more. . .
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Testing and Requirements Validation

- Testing is the process of **evaluating** a system or application to **ensure** that it meets the **requirements** and **expectations** of the stakeholders.
- Testing is an essential part of the software development process, as it helps to **identify defects** and **improve** the quality of the software.
- There are many different types of testing, including:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing
  - And many more...
- Testing should be planned and executed throughout the software development process, from the **requirements** phase to the **deployment** phase.

Outline

1 Domain-Driven Design

2 Business Systems Analysis

3 Software Methodologies

4 Requirements Engineering

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Thanks!

Questions?

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