

# BUSINESS SYSTEMS

## Systems Analysis

Author: Eng. Carlos Andrés Sierra, M.Sc.  
[cavirguezs@udistrital.edu.co](mailto:cavirguezs@udistrital.edu.co)

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

2024-III



# Outline

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



# 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 *entity, 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**.

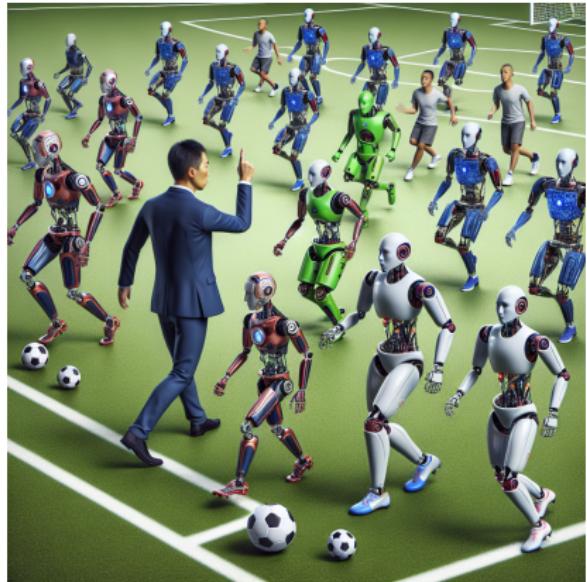


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



# 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**.

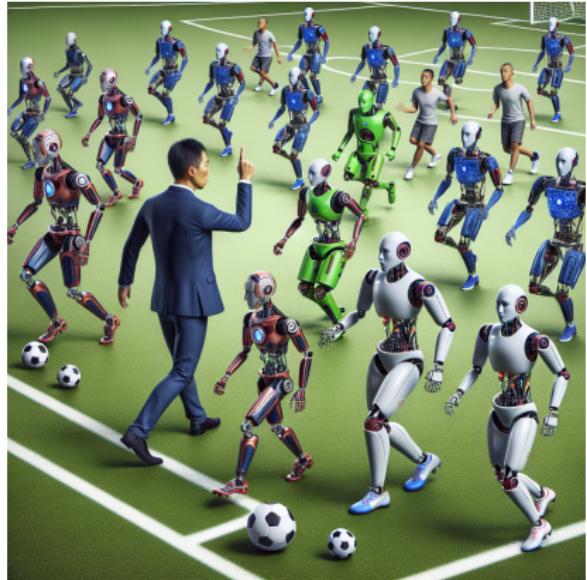


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



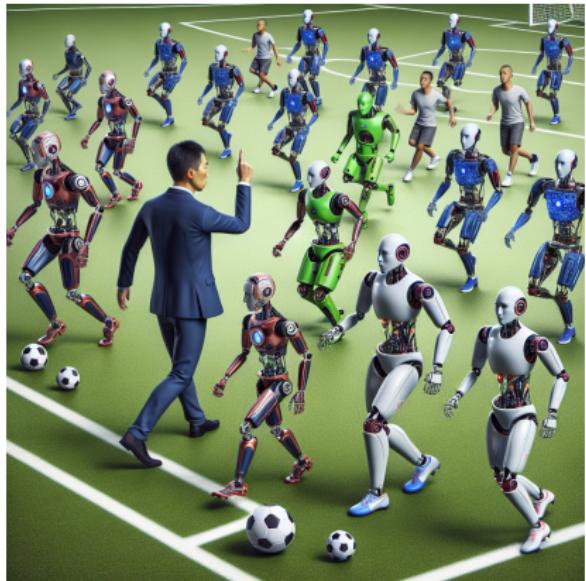
# Basics of Domain-Driven Design I

- 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:



# Basics of Domain-Driven Design II

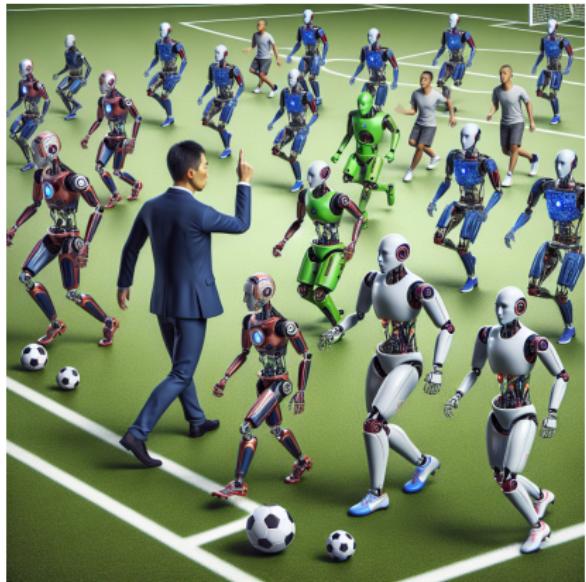


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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:



# Basics of Domain-Driven Design II

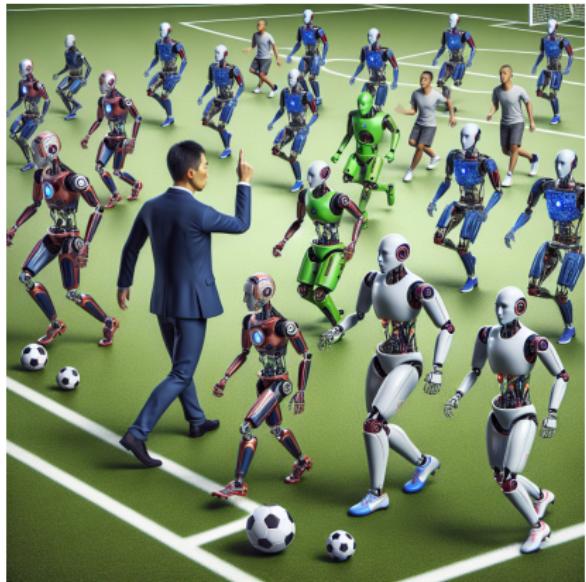


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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:

# Basics of Domain-Driven Design II

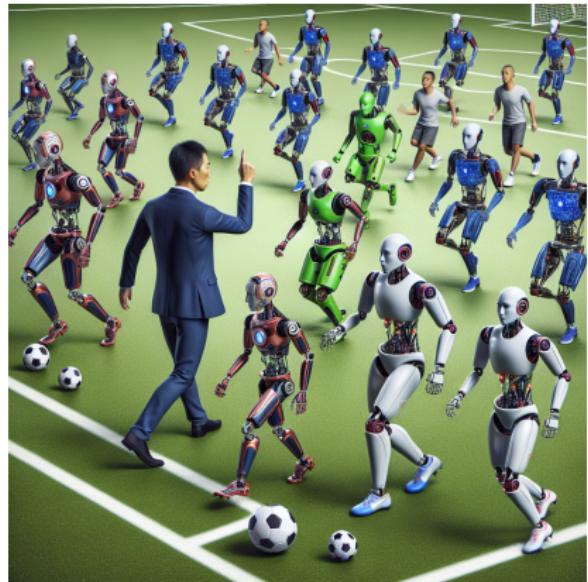


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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:

# Basics of Domain-Driven Design II

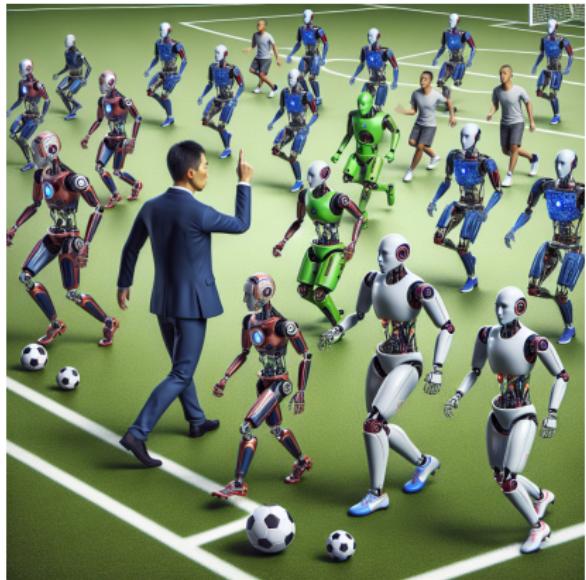


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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:



# Basics of Domain-Driven Design II

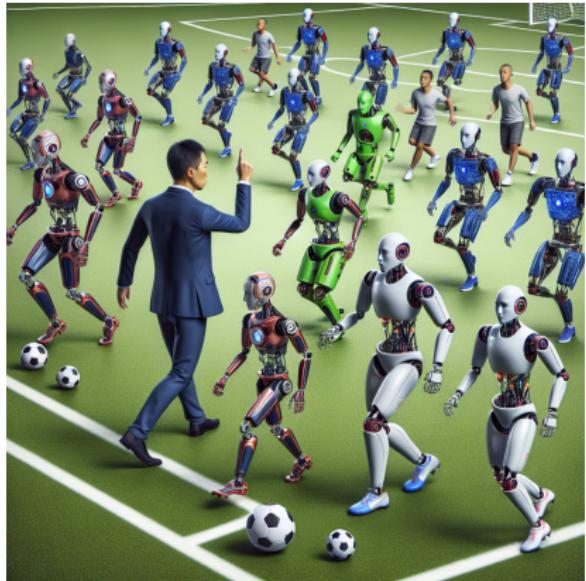


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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

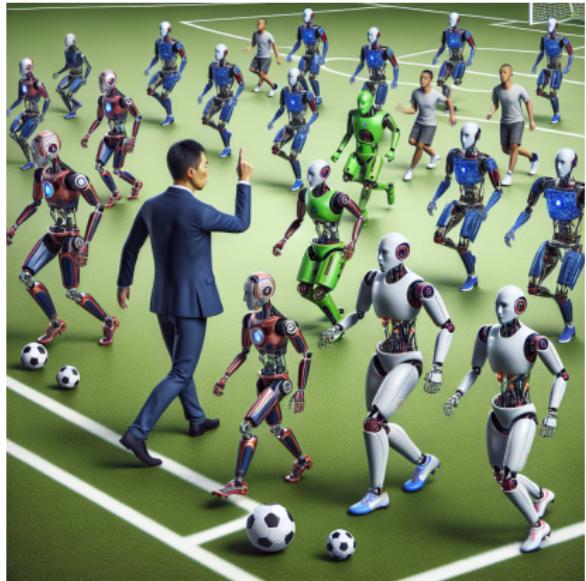


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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** *↳ business rules*
  - **Presentation Layer**
  - **Infrastructure Layer.**

# Basics of Domain-Driven Design II

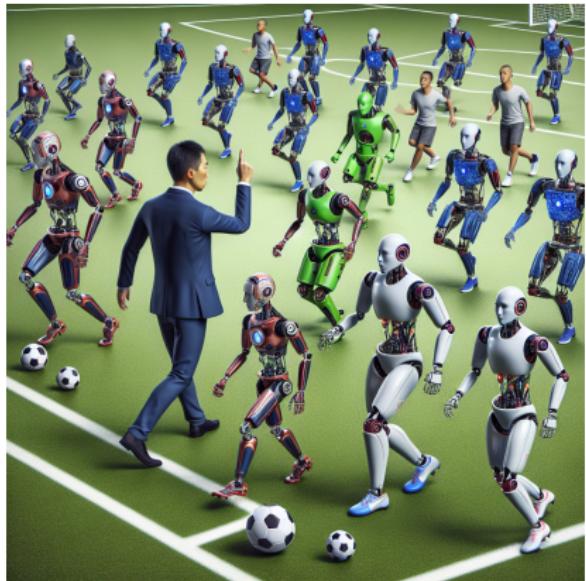


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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.**
  - **Presen~~tation~~ Layer**
  - **Infrasctructure Layer.**



# Basics of Domain-Driven Design II

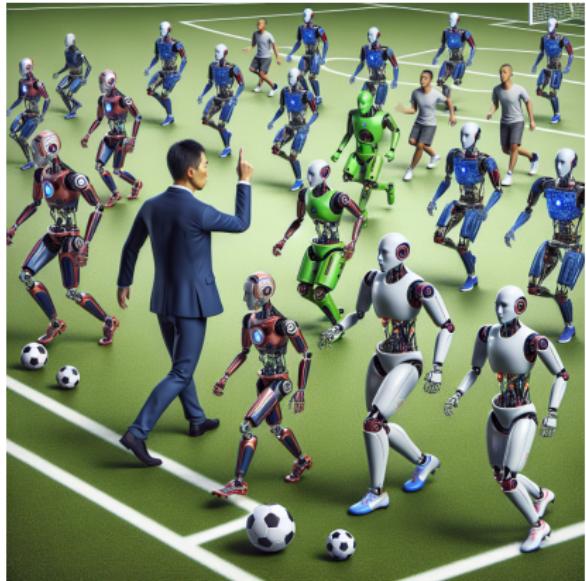


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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

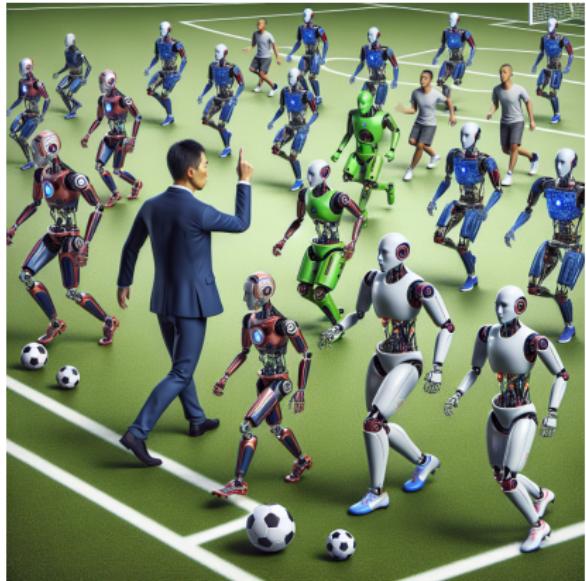
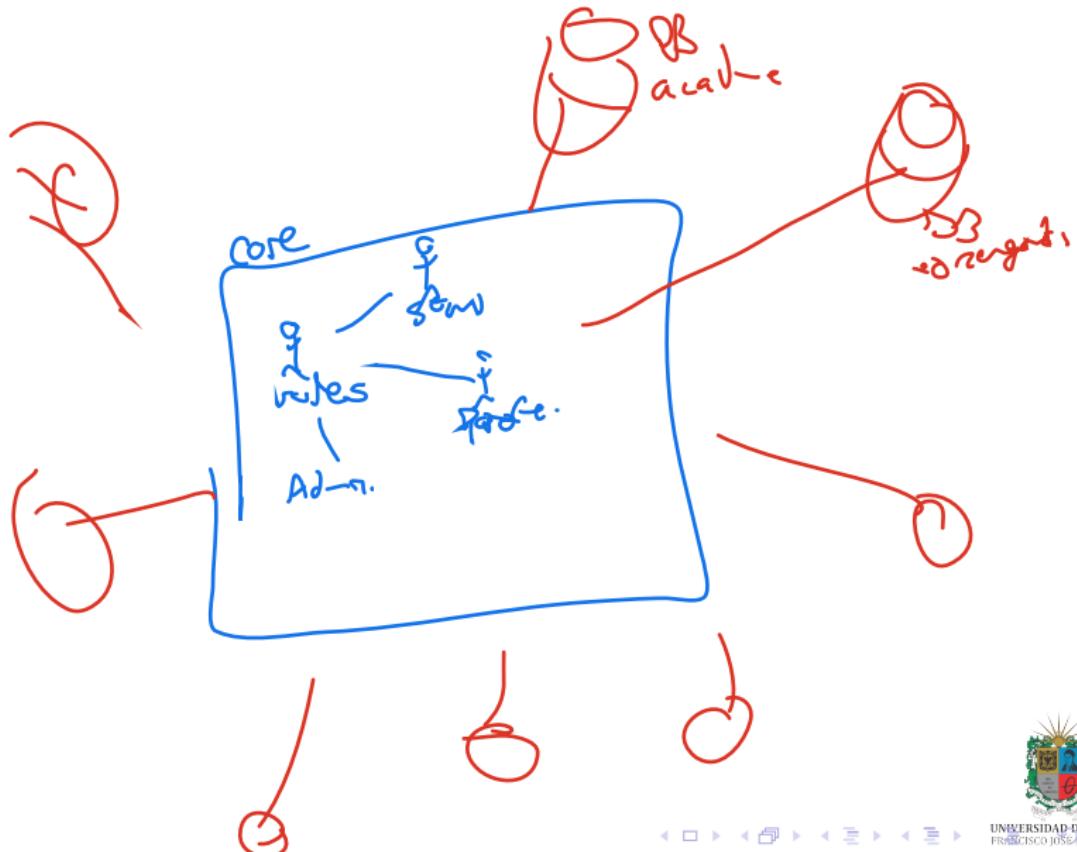


Figure: Prompt: Draw a soccer coach teaching robots soccer players.



- 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**.

# Case of Study: DDD for Condor



# 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
  - Financial management systems
  - Human resources management systems
  - Manufacturing execution systems (MES)
  - Quality management systems
- 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
  - Financial management systems
  - Human resources management systems
  - Manufacturing execution systems
  - Supply chain management systems
- 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.



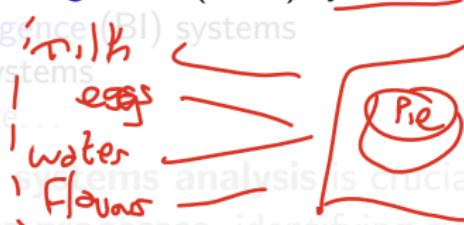
# Business Systems

- **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.



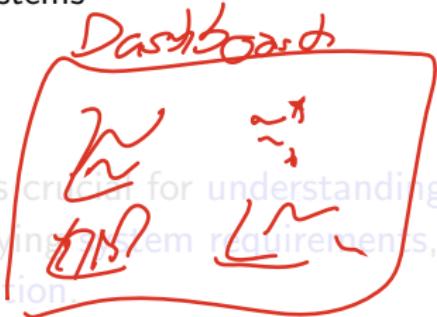
# Business Systems

- **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 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 user 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:

• Gathering requirements  
• Analyzing requirements  
• Designing solutions  
• Implementing solutions  
• Evaluating solutions



# 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.  
*systems analysis*
- 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:  
*softw*  
*work*



# 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 business 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 business 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 business requirements
  - Creating business process models and diagrams
  - Collaborating with stakeholders to validate requirements



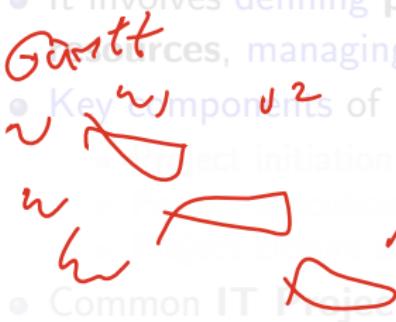
# Business Models Examples

## Understanding Business Models Through Flywheels



# 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 scope, creating a project plan, allocating resources, managing risks, and monitoring project progress.
- Key components of IT Project Management include:
- Common IT Project Management methodologies include:



# 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  
Planning

↓  
*Ciclical*

Project execution

↓  
*Flexibay*  
*Agile Methodologies*

- Common IT Project Management methodologies include:



# 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:

LICEN~~CE~~

~~→ \$ other~~

- Common Project management methodologies include:



# 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
  - PRINCE2
  - PERT
  - Gantt chart



# IT Project Management

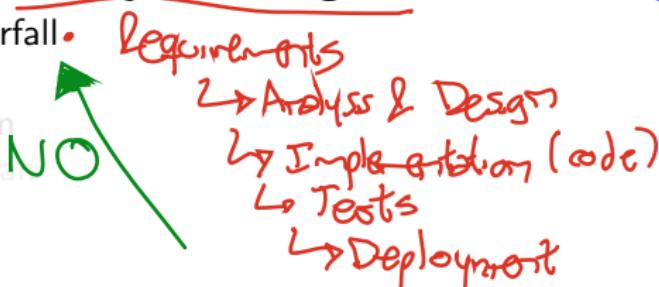
- **IT Project Management** is the **process** of planning, organizing, and controlling the resources and activities required to complete an **IT project**.
- It involves **defining project goals**, **creating a project plan**, **allocating resources**, **managing risks**, and **monitoring project progress**.
- 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

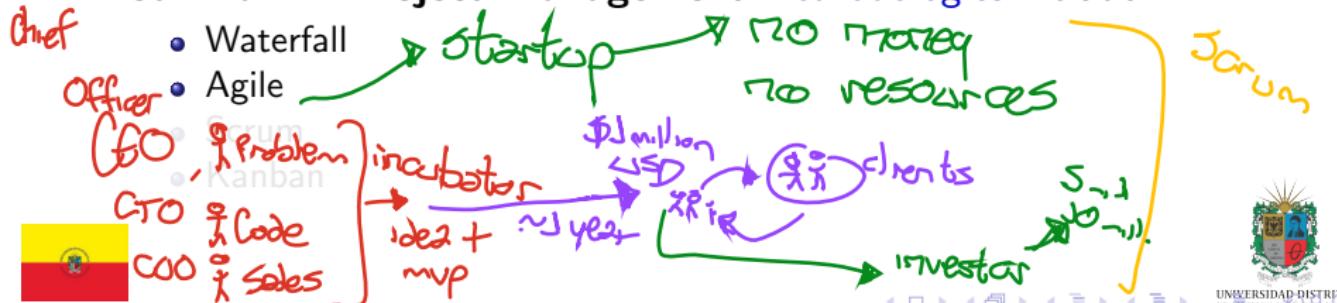
- **IT Project Management** is the **process** of planning, organizing, and controlling the resources and activities required to complete an **IT project**.
- It involves **defining project goals**, **creating a project plan**, **allocating resources**, **managing risks**, and **monitoring project progress**.
- 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:



# 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



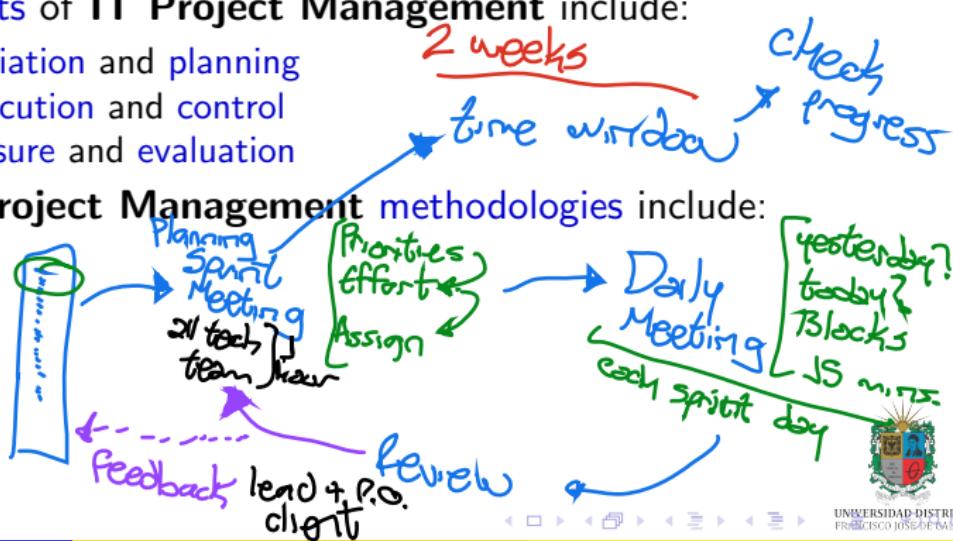
# IT Project Management

*Product Owner  $\Rightarrow$  Client interactions*

- **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

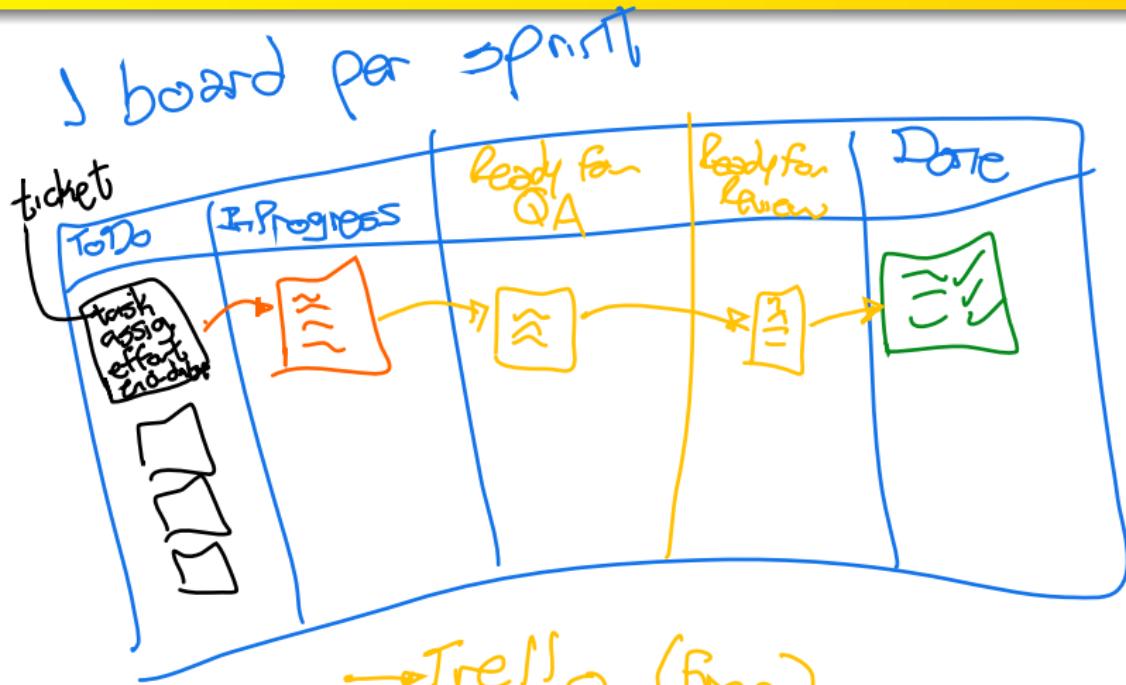
*Backlog  
(requirements)*



*Team S-JO*



# Case of Study: Example of a KANBAN Board



→ Trello (Free)

→ Jira



# Outline

1 Domain-Driven Design

2 Business Systems Analysis

•

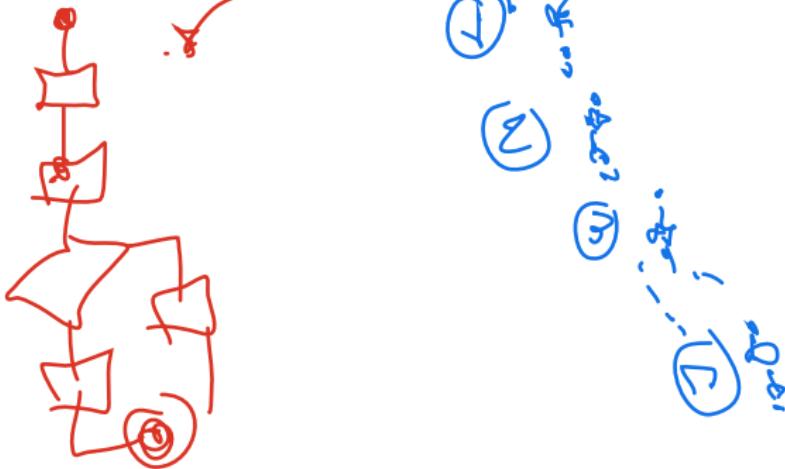
3 Software Methodologies

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.

Practical

9 notes  
S ~ - ↓



# Agile Methodologies

*Agile Mgr.Fest*

- 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 team no so much budget, you need to think smart and think fast.  
*systems analysis*
- 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.

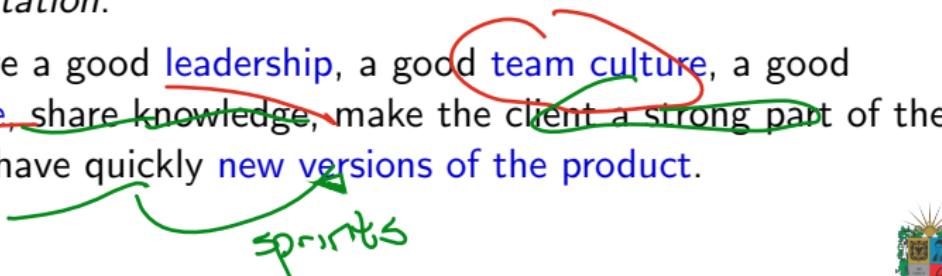


# Agile Methodologies

- 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 code a strong part of the process, and have quickly new versions of the product.

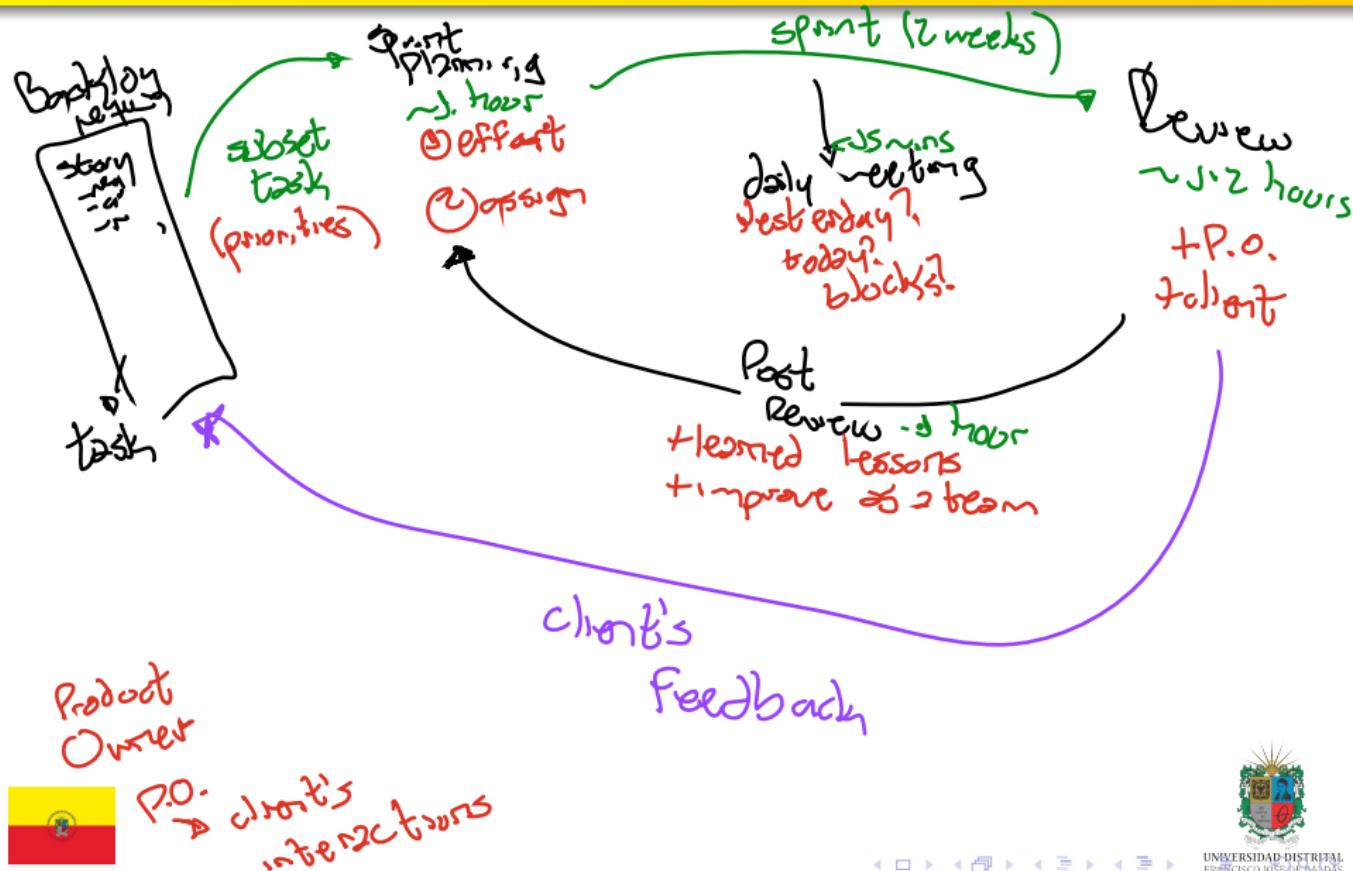


# Agile Methodologies

- 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.
- 



# Case of Study: Example of a SCRUM Workflow



# 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.  
*(1) 1 2 3* **Pr. Priorities**

- 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.  
*each one!* **technical tasks**

- Key activities in requirements engineering include:

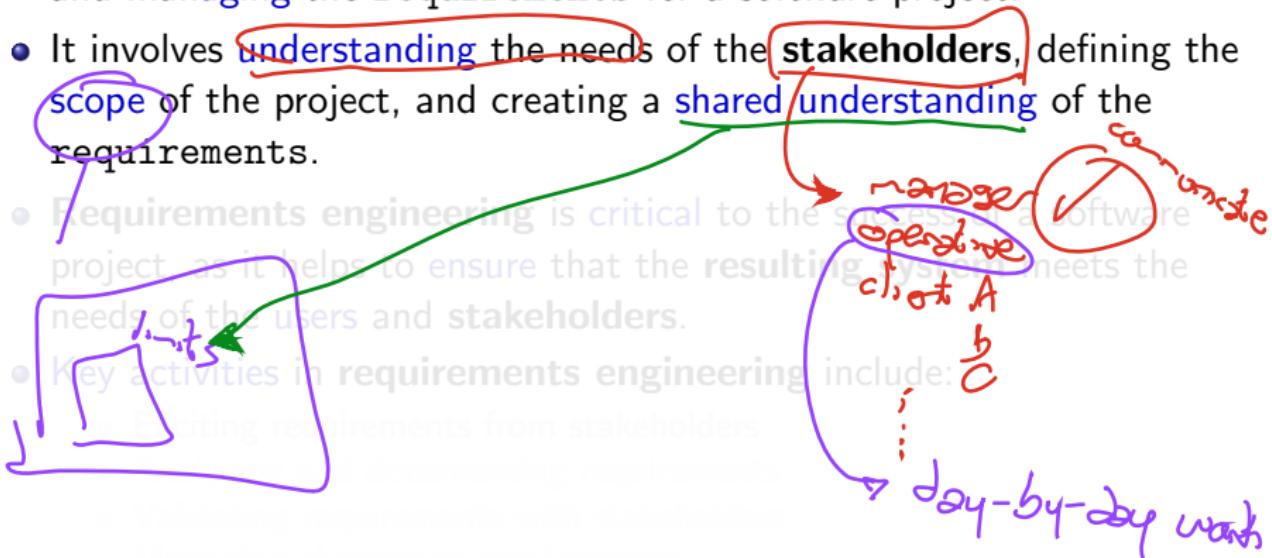
*1  
2  
3  
4  
5  
6  
7  
8  
9  
10*

**technical**



# 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:
  - Identify requirements from stakeholders
  - Analyze requirements
  - Specify requirements
  - Validate requirements
  - Manage requirements



# Requirements Engineering I

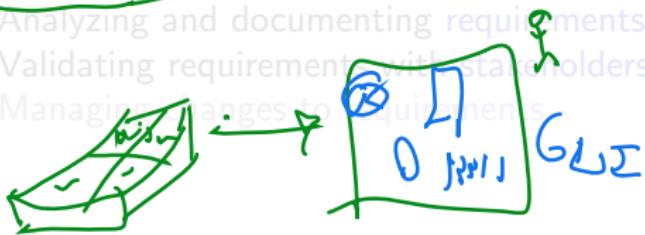
- Requirements engineering is the process of gathering, documenting, and managing the requirements for a software project.
- It involves understanding the needs of the stakeholders, defining the scope of the project, and creating a shared understanding of the requirements.
- Requirements engineering 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



# 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



bridge

# 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

- 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** aligned 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

- 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** aligned 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.



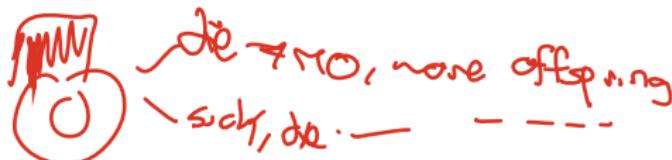
# Requirements Engineering II

- 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** aligned 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

- 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** aligned 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 gather information from clients: interviews, brainstorming, questionnaires, etc. This current client processes documentation. All depends of the nature of the problem, and the impact of that one into the organization.  
**User**
- Sometimes a coffee conversation, a game with lego pieces, the **direct interaction** with the client could bring you more useful information about the problem and the expected solution.  
**Impact client**  
**Not fit**



# 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 gather **information** from **clients**: interviews, brainstorming sessions, analysis current client processes documentation. All depends of the **nature of the problem**, and the **impact** of that one into the **organization**.
- Sometimes a **Chaos Theory**, 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 gather **information** from **clients**: interviews, brainstorming sessions, analysis current client processes 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 widely accepted approach, where 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 widely **accepted approach**, where 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
- He As a "role", want "\_\_\_", so what for validation stages.*



# User Stories

- To **communicate** in the **same vocabulary** with the client decreases the **misunderstandings**, avoid confusion.
- Nowadays **user stories** are a widely **accepted approach**, where 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 test~~ **design**
  - ~~integration test~~ **code**
  - ~~acceptance test~~ **test**
  - ~~final test~~ **tech**
- 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
  - Regression testing
  - Performance testing
  - Security testing
- 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 → *Factory / methods*
  - Integration testing
  - System testing
  - Acceptance testing → *input X output Z*
  - 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.
- Components/modules*



# 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.
- do -> requirements / non-functionalities*



# 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 → *functional - user accept*
  - 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



# Thanks!

## Questions?



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

