The importance of

*Modeling*

*Unified Modeling Language* as a standard language to analyze, design and document software intensive solutions
A model is a simplification of reality. We build models so we can better understand the system we are developing.

**Aims achieved throughout modeling**

- Model help us to visualize a system as it is or as we want it to be
- Models permit us to specify the structure or behavior of a system
- Models gives us a template that guides us in constructing a system
- Models document the decision we have made
Introducing the UML (1/2)

**UML is a Language**
A modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system.

**UML is a Language for Visualizing**
Some things are best modeled textually; other are best modeled graphically. In all interesting system, there are structures that transcend what can be represented in a programming language.

**UML is a Language for Specifying**
Specifying means building models that are precise, unambiguous and complete. The UML addresses the specification of all the important analysis, design and implementation decision.

**UML is a Language for Constructing**
The UML is not a visual programming language, but its models can be directly connected to a variety of programming language (Java, C++, ...)

The **UML** is a language for
- *Visualizing*
- *Specifying*
- *Constructing*
- *Documenting*
the artifact of a software-intensive system.
The UML is a language for
- Visualizing
- Specifying
- Constructing
- Documenting
the artifact of a software-intensive system.

A software solution is made up by all sort of artifacts in addition to raw executable code. These artifacts include:

- Requirements
- Architecture
- Design
- Source code
- Project Plans
- Tests
- Prototypes
- Releases
The UML is not limited to modeling software. It is expressive enough to model non software system, such as workflow in the legal system, the structure and behavior of a patient healthcare system and the design of hardware.
UML history

“The three amigos”

Grady Booch
Ivar Jacobson
James Rumbaugh
Available UML tools
What next...

**UML** is a *language*, its is *not a method nor a developing tool.*

After the formalization of the solution, each developer can decide to implement the solution using any available tool: **JDeveloper, JBuilder, Visual Café, Jamba, Visual C++, etc…**

**Integrated tools allow to design and implement within the same suite.**
The entire picture

Project planning

Project delivery

Project programming

Analysing

Defining

Designing

Building & Implementing

Testing

PwC Methods

Language

Implementation tools

Design tools

Design tools

PwC Methods

Language