IBM Software Group

Essentials of Visual Modeling with UML

Module 2: Principles of Visual Modeling

Rational. software

Objectives

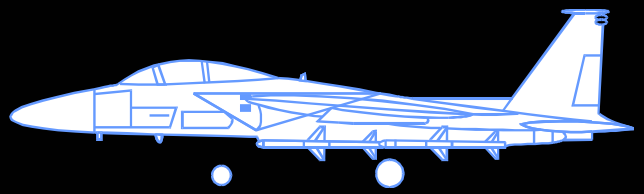
- Describe the importance of visual modeling.
- Define the four principles of visual modeling.
- Explain what the Unified Modeling Language (UML) represents.
- Define the type of process that best relates to the UML.

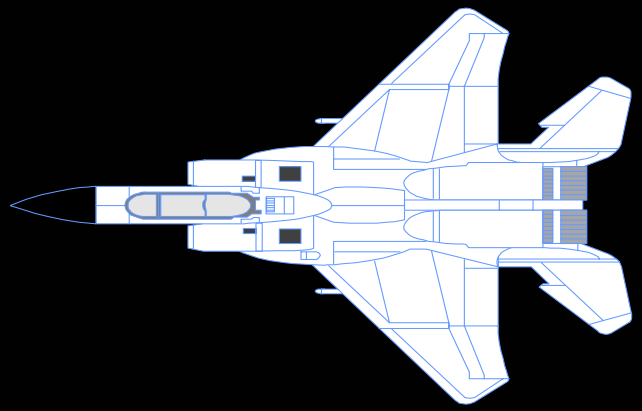
Where Are We?

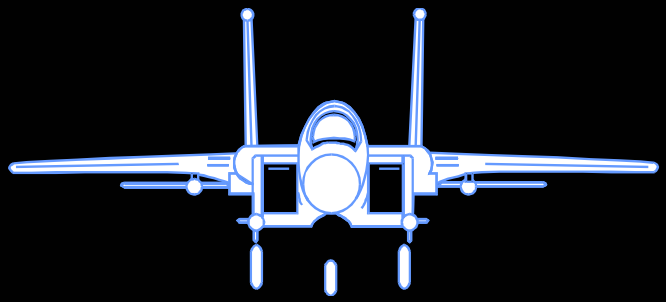
- ★ What is modeling?
  - Four principles of visual modeling
  - The UML
  - Process and visual modeling

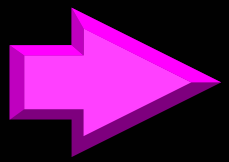
What Is a Model?

- A model is a simplification of reality.









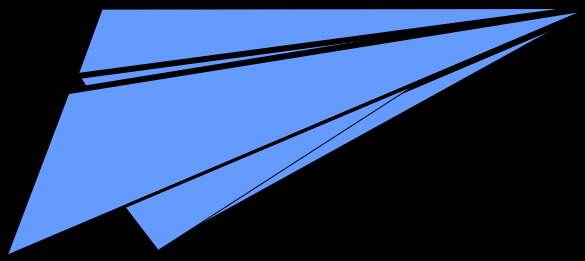
Why Model?

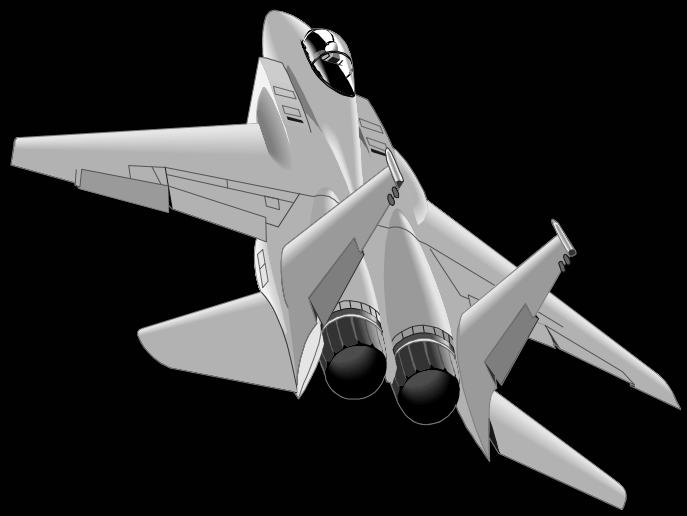
- Modeling achieves four aims:
  - Helps you to visualize a system as you want it to be.
  - Permits you to specify the structure or behavior of a system.
  - Gives you a template that guides you in constructing a system.
  - Documents the decisions you have made.
- You build models of complex systems because you cannot comprehend such a system in its entirety.
- You build models to better understand the system you are developing.

The Importance of Modeling

Less Important

More Important





Paper Airplane

Fighter Jet

Software Teams Often Do Not Model

- Many software teams build applications approaching the problem like they were building paper airplanes
  - Start coding from project requirements
  - Work longer hours and create more code
  - Lacks any planned architecture
  - Doomed to failure
- Modeling is a common thread to successful projects

Where Are We?

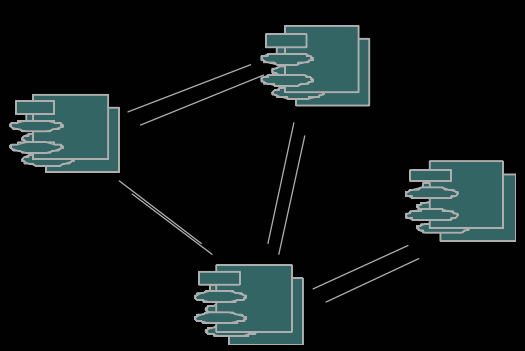
- What is modeling?
- ★ Four principles of visual modeling
- The UML
- Process and visual modeling

Four Principles of Modeling

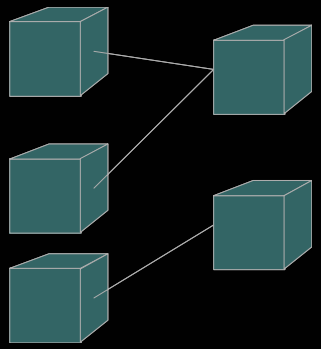
- The model you create influences how the problem is attacked.
- Every model may be expressed at different levels of precision.
- The best models are connected to reality.
- No single model is sufficient.

Principle 1: The Choice of Model Is Important

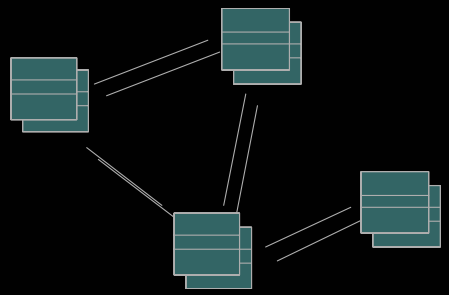
- The models you create profoundly influence how a problem is attacked and how a solution is shaped.
  - In software, the models you choose greatly affect your world view.
  - Each world view leads to a different kind of system.



Process Model



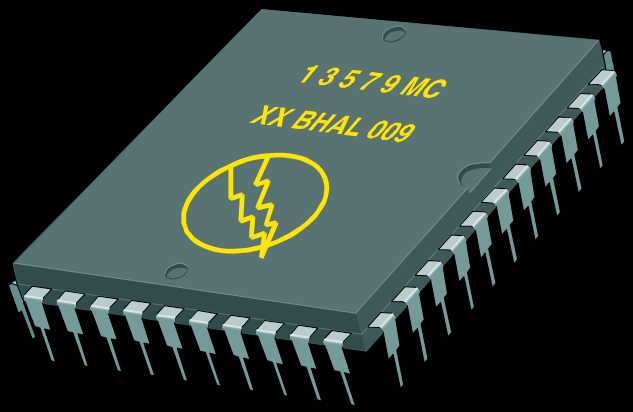
Deployment Diagram



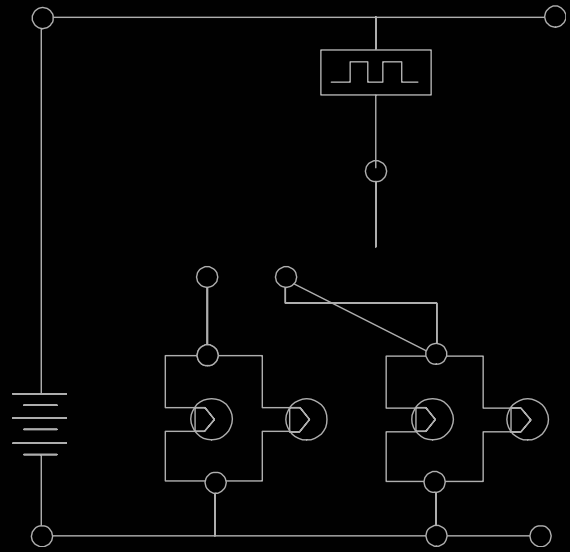
Design Model

Principle 2: Levels of Precision May Differ

- Every model may be expressed at different levels of precision.
  - The best kinds of models let you choose your degree of detail, depending on:
    - Who is viewing the model.
    - Why they need to view it.



View for Designers



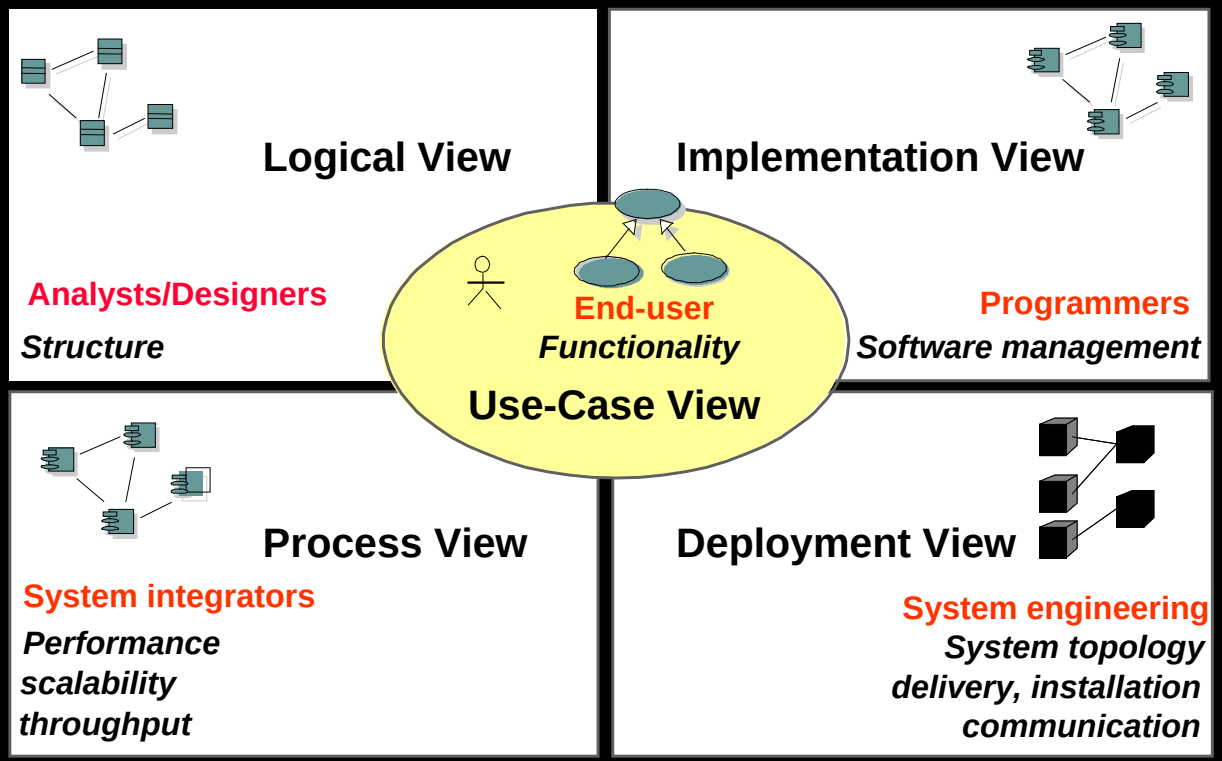
View for Customers

Principle 3: The Best Models Are Connected to Reality

- All models simplify reality.
- A good model reflects potentially fatal characteristics.

Principle 4: No Single Model Is Sufficient

- No single model is sufficient. Every non-trivial system is best approached through a small set of nearly independent models.
  - Create models that can be built and studied separately, but are still interrelated.



Where Are We?

- What is modeling?
- Four principles of visual modeling
- ★ The UML
- Process and visual modeling

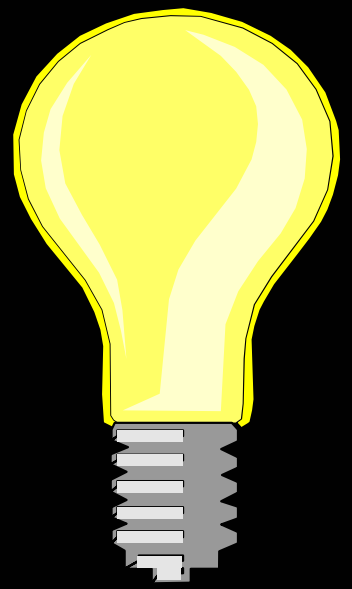
What Is the UML?

- The UML is a language for
  - Visualizing
  - Specifying
  - Constructing
  - Documenting

  the artifacts of a software-intensive system.

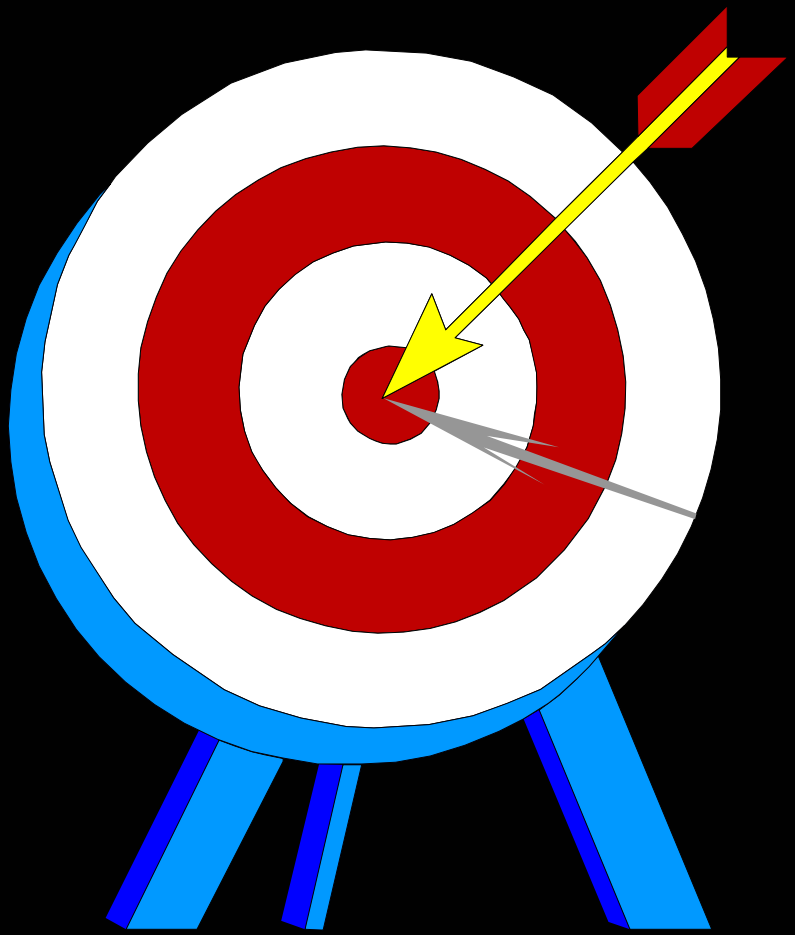
The UML Is a Language for Visualizing

- Communicating conceptual models to others is prone to error unless everyone involved speaks the same language.
- There are things about a software system you can't understand unless you build models.
- An explicit model facilitates communication.



The UML Is a Language for Specifying

- The UML builds models that are precise, unambiguous, and complete.

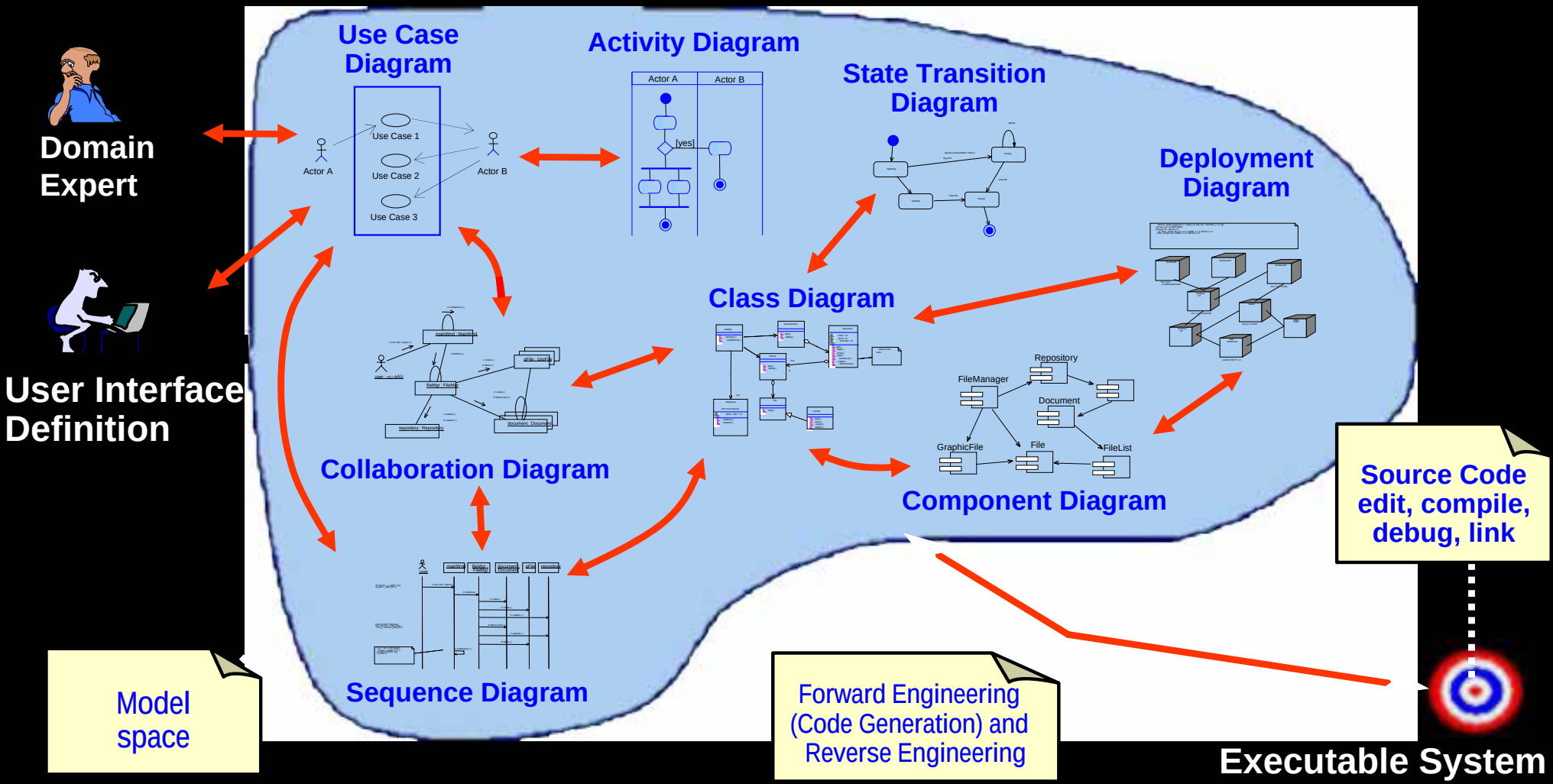


The UML Is a Language for Constructing

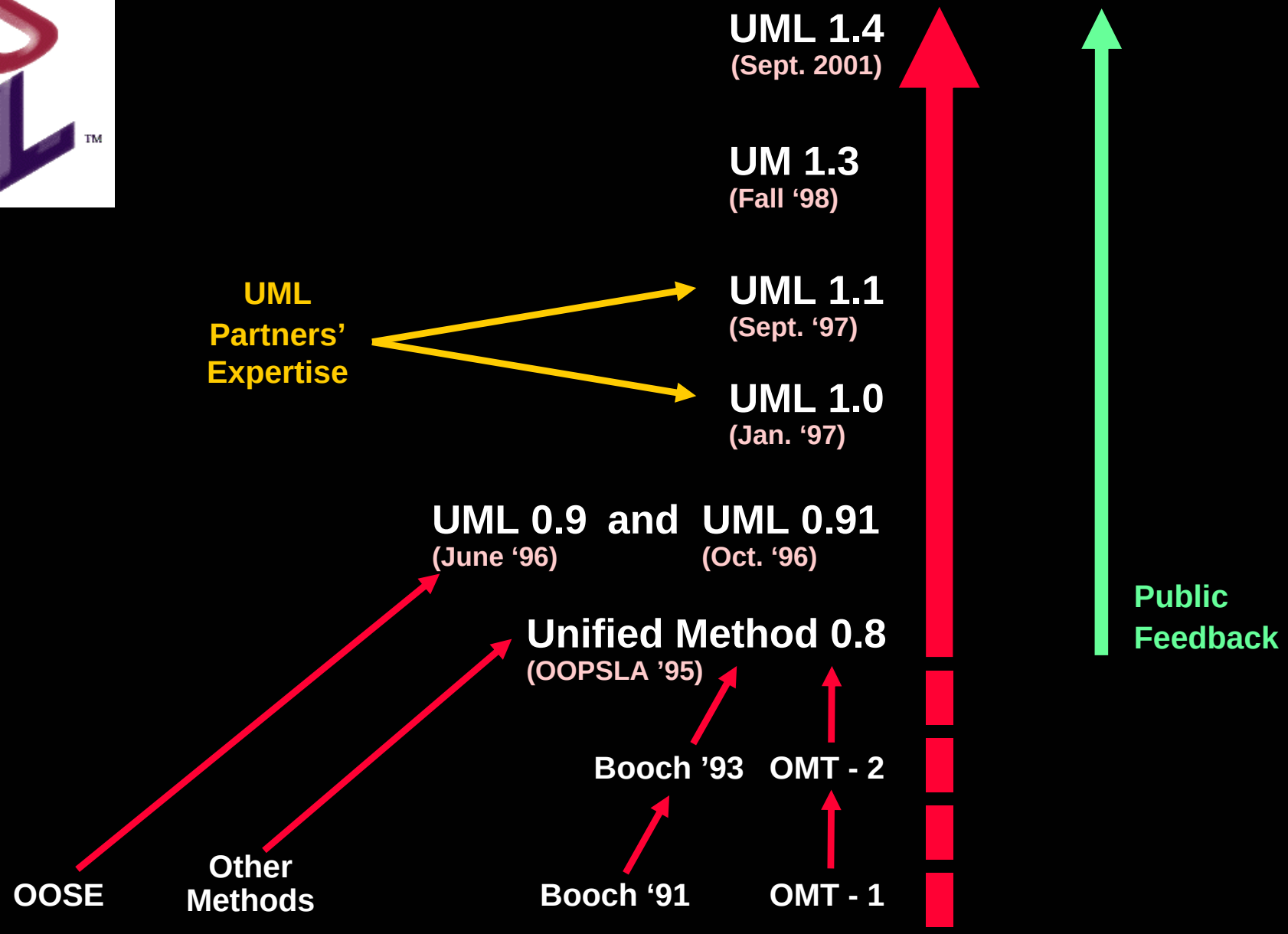
- UML models can be directly connected to a variety of programming languages.
  - Maps to Java, C++, Visual Basic, and so on
  - Tables in a RDBMS or persistent store in an OODBMS
  - Permits forward engineering
  - Permits reverse engineering

The UML Is a Language for Documenting

- The UML addresses documentation of system architecture, requirements, tests, project planning, and release management.

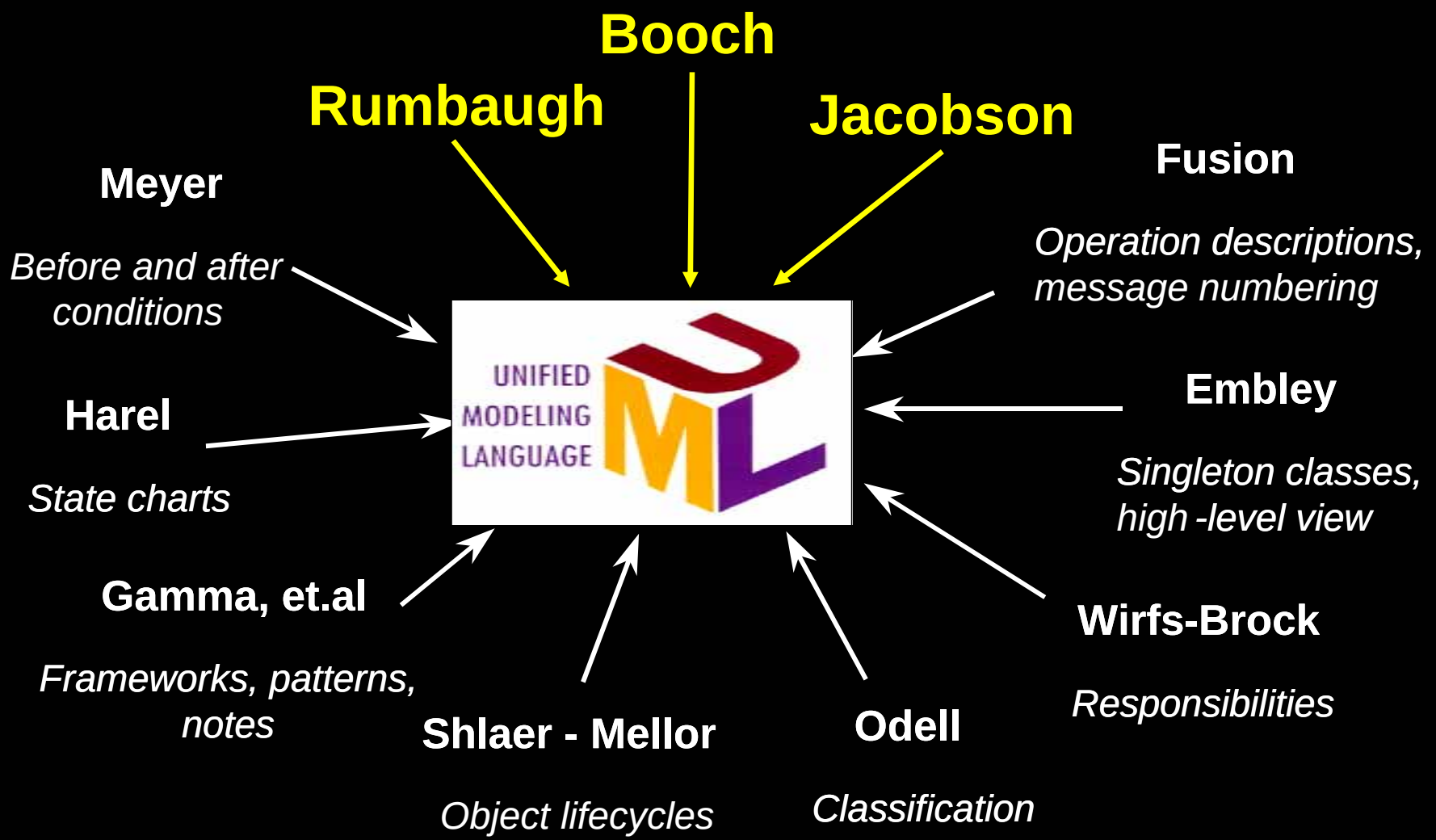


History of the UML



IBM

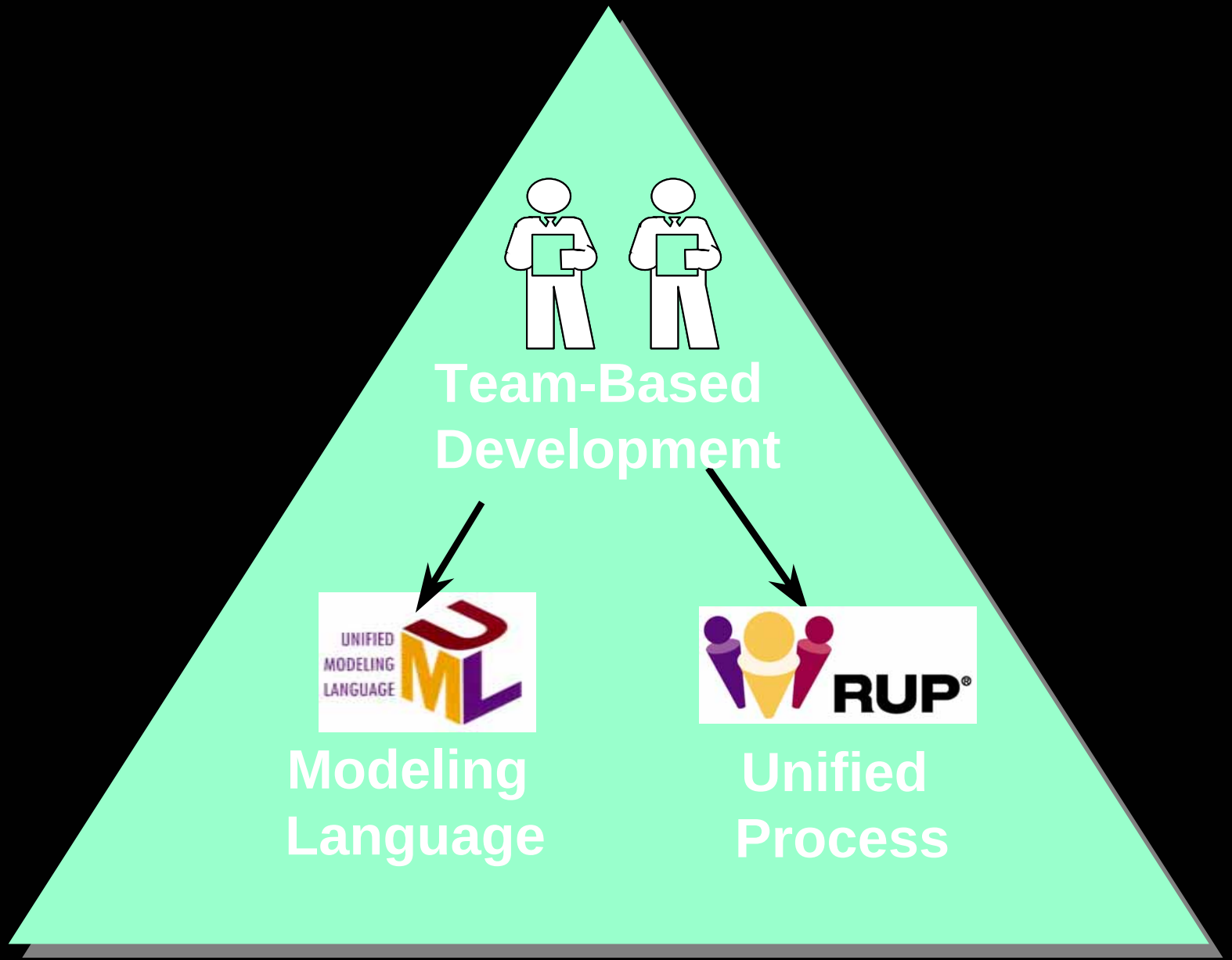
Inputs to the UML



Where Are We?

- What is modeling?
- Four principles of visual modeling
- The UML
- ★ Process and visual modeling

A Language Is Not Enough to Build a System

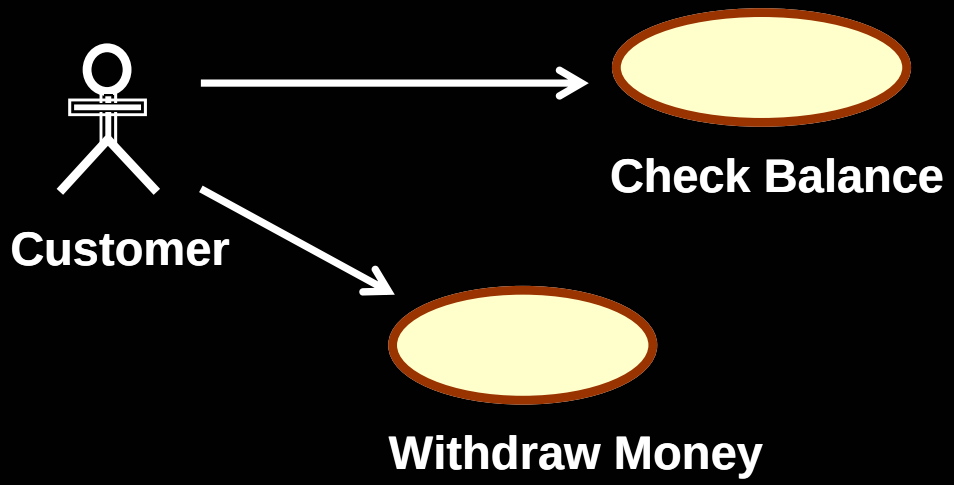


What Type of Process Most Benefits the UML?

- The UML is largely process independent. A process fully benefits from the UML when the process is:
  - Use-case driven
  - Architecture-centric
  - Iterative and incremental

A Use-Case Driven Process

- Use cases defined for a system are the basis for the entire development process.
- Benefits of use cases:
  - Concise, simple, and understandable by a wide range of stakeholders.
  - Help synchronize the content of different models.



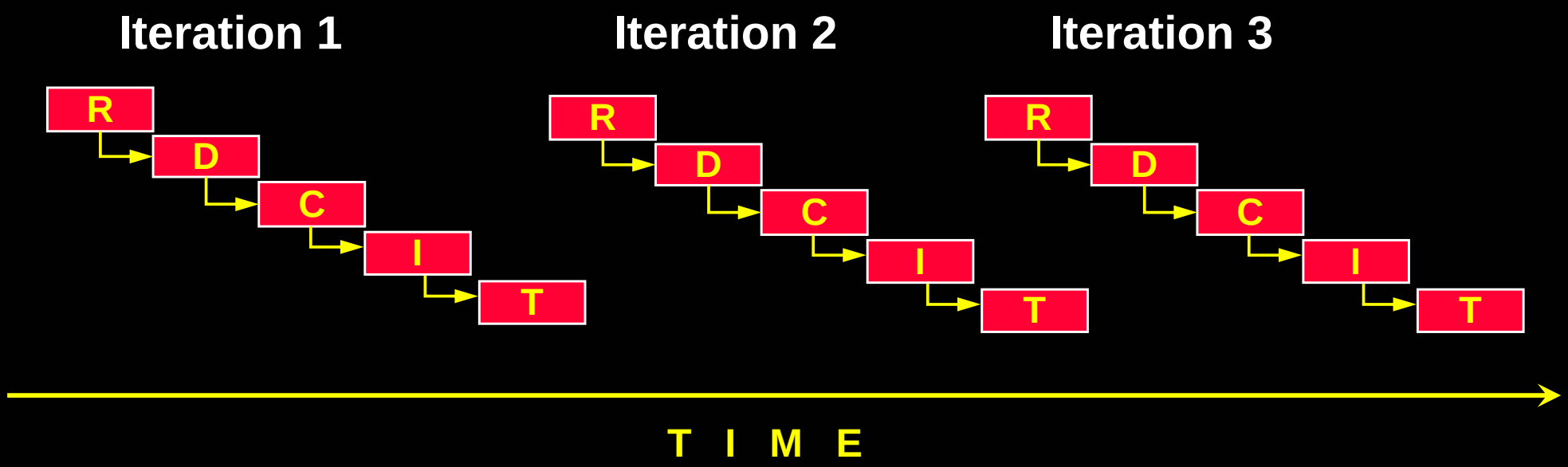
An Architecture-Centric Process

- A system's architecture is used as a primary artifact for conceptualizing, constructing, managing, and evolving the system under development.
- Benefits:
  - Intellectual control over a project to manage its complexity and to maintain system integrity.
  - Effective basis for large-scale reuse.
  - A basis for project management.
  - Assistance in component-based development.

An Iterative and Incremental Process

- Critical risks are resolved before making large investments.
- Initial iterations enable early user feedback.
- Testing and integration are continuous.
- Objective milestones focus on the short term.
- Progress is measured by assessing implementations.
- Partial implementations can be deployed.

Iterative Development



- Earliest iterations address greatest risks.
- Each iteration produces an executable release, an additional increment of the system.
- Each iteration includes integration and test.

Review

- What is a model?
- What are the four principles of modeling? Describe each one.
- What is the UML? Describe each of its four benefits.
- What process characteristics best fit the UML? Describe each characteristic.
- What is an iteration?