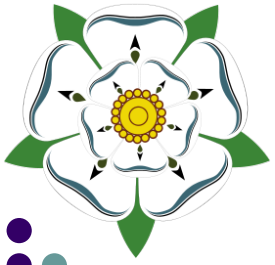


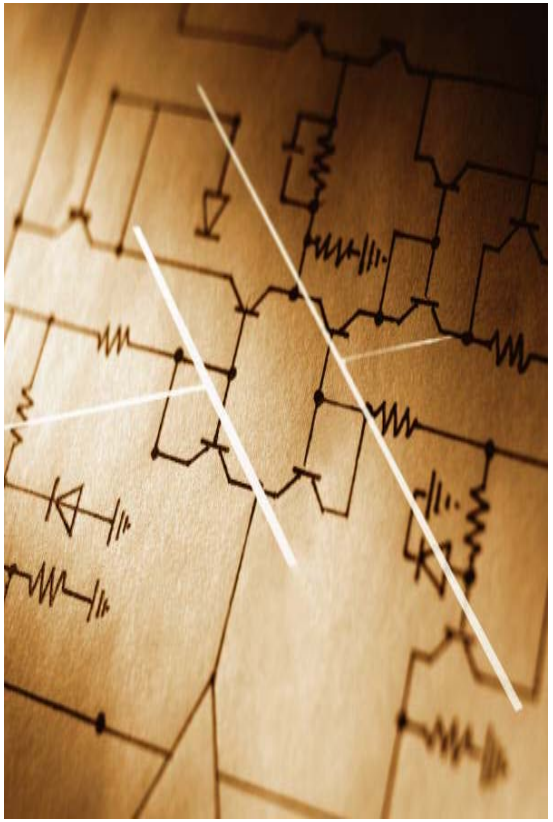
Modeling Overview

"... as we know, there are known knowns;
there are things we know we know.
We also know there are known unknowns;
that is to say we know there are some things we do not know.
But there are also unknown unknowns
- the ones we don't know we don't know."

Donald Rumsfeld

What is a Model?

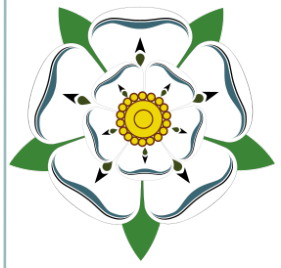
- A model is essentially a picture or representation of something of interest
- Models are used to illustrate a concept or idea that is typically difficult to describe using the written word
- Sometimes called a schematic or a blueprint



Why Model?

- A model helps us communicate ideas or concepts in a relatively precise and unambiguous way
- A model supplements the written word when words fail to convey the meaning
- A Picture is worth a thousand words!

The IT Model



- An abstract but formal representation of entities including their properties, relationships and the operations that can be performed on them
- A visual way of depicting a business, its rules, the use of systems, applications, system architectures, and interactions within the systems

When to Model?

- Models can be created at any point in the development life cycle
- Models are most effective when new concepts or ideas are being evolved
- Models always precede the implementation of the concepts that they illustrate
  - Pictures before code
  - Pictures after code have little value

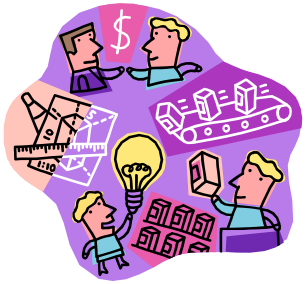
Requirements to Design?

Requirements

Design

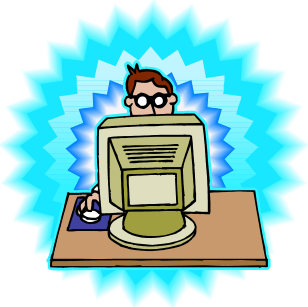
The Analysis Process

Reality

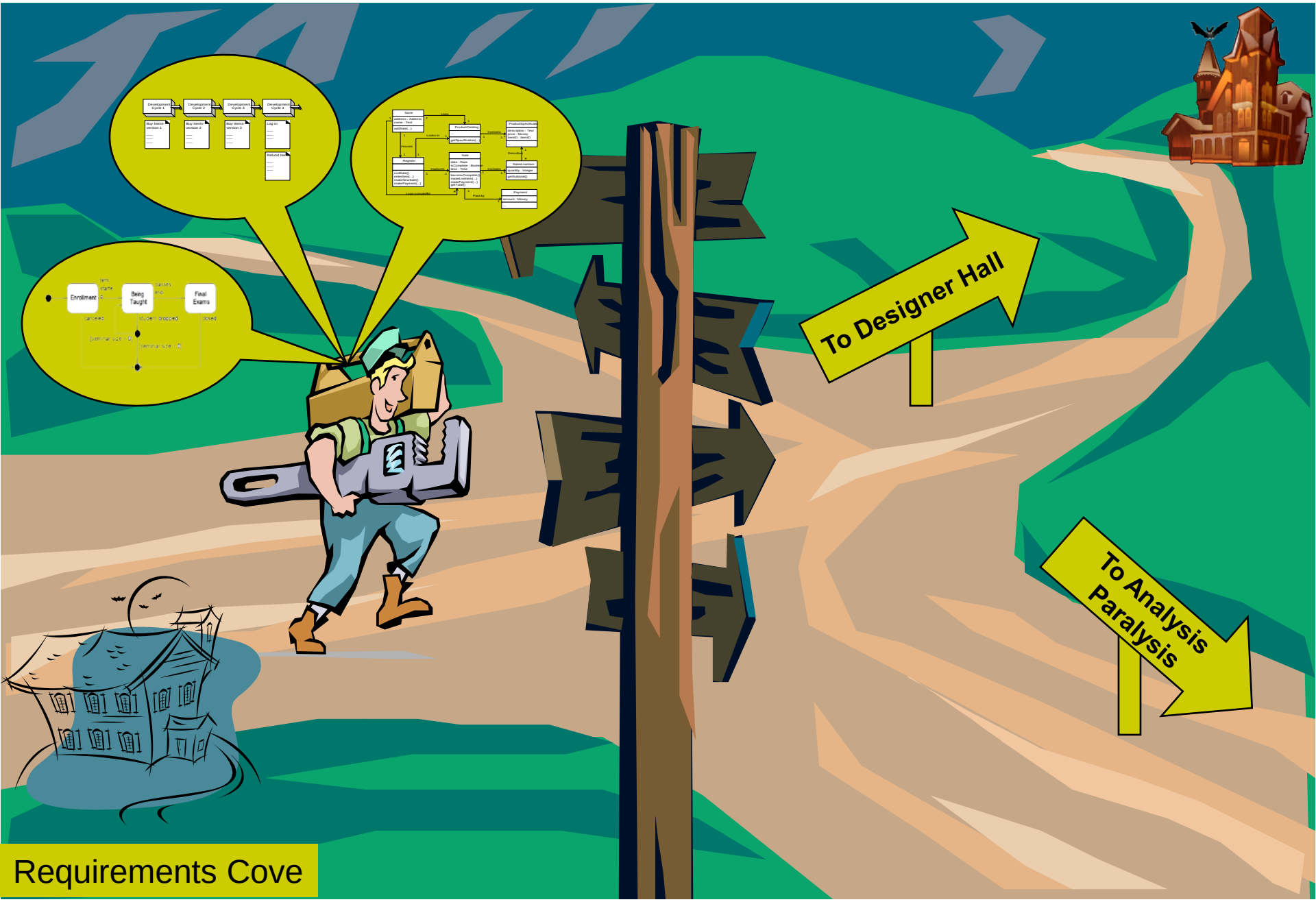


Model of Reality

Design

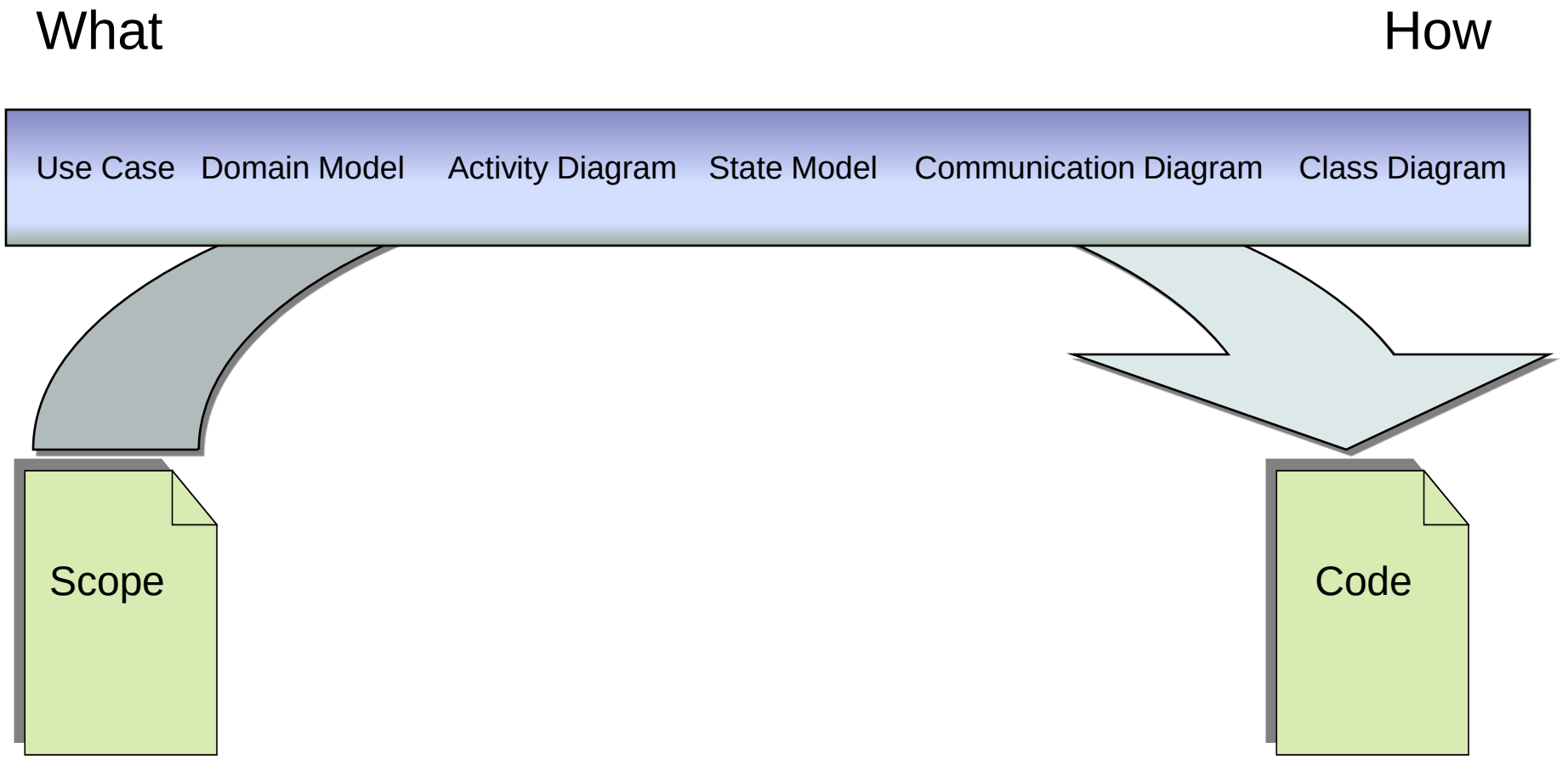


Code

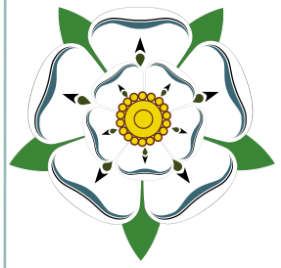


Requirements Cove

The Analysis Tools

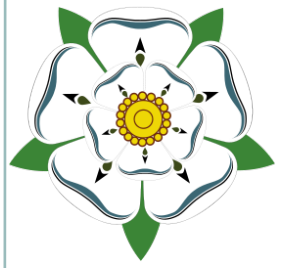


Model Persistence



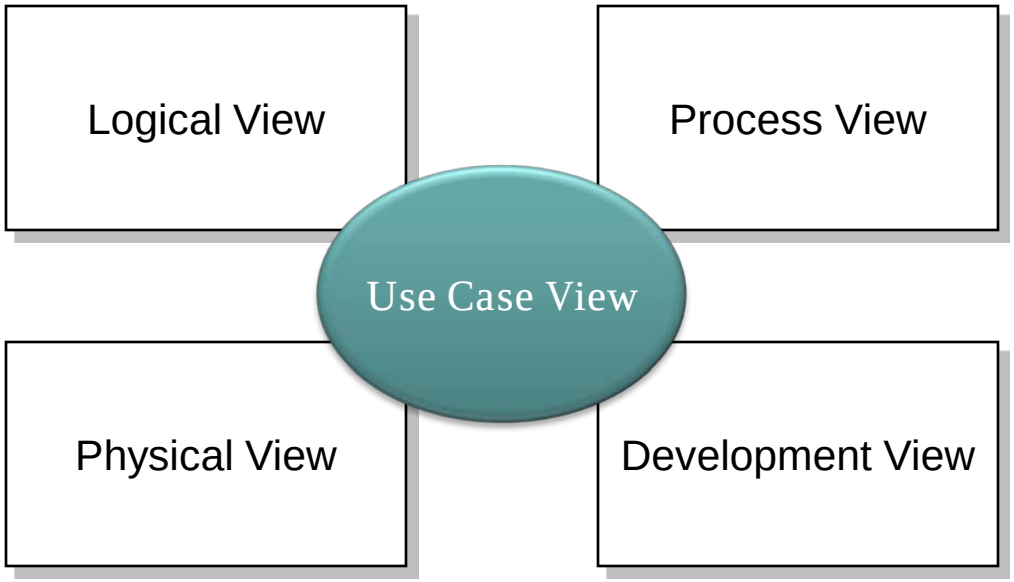
- Models can be transient
  - They exist only as long as it is needed to aid in the communication of ideas or concepts
- Models may be permanent
  - They exist for as long as the concept that they illustrate
  - They can become part of the system documentation
  - They form a historical record

The Unified Modeling Language



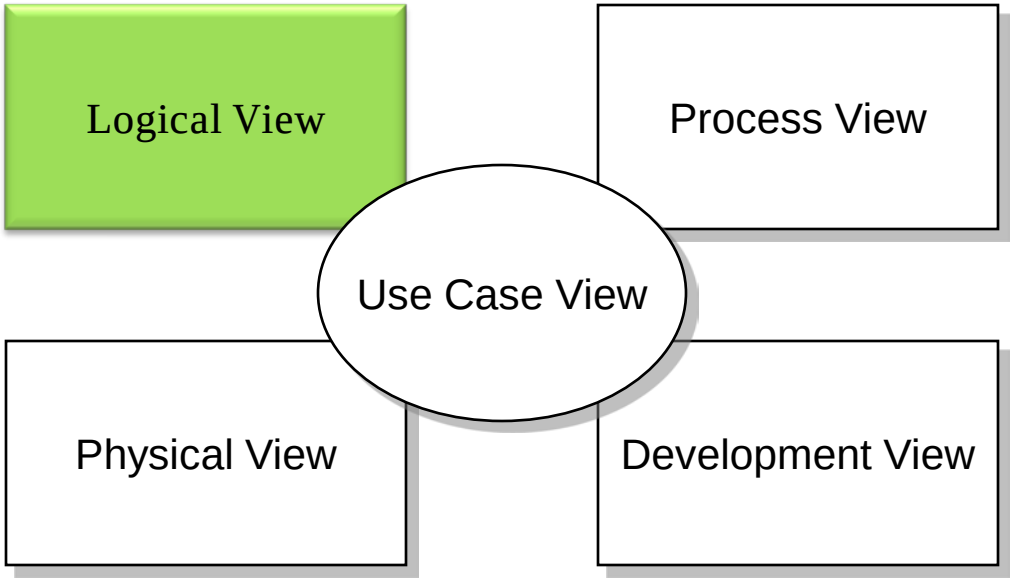
- Evolved from the work of Booch, Rumbaugh, Mellor, Yourdon, Harel, Odell et al.
- Each contributed ideas, concepts, modeling techniques, notations
- Consortium of partners including HP, IBM, TI, DEC, Microsoft and Rational formalized a standard
- Came under the governance of the Object Management Group in 1997 as UML 1.0
- Current version UML 2.2 published February, 2009

Model Views



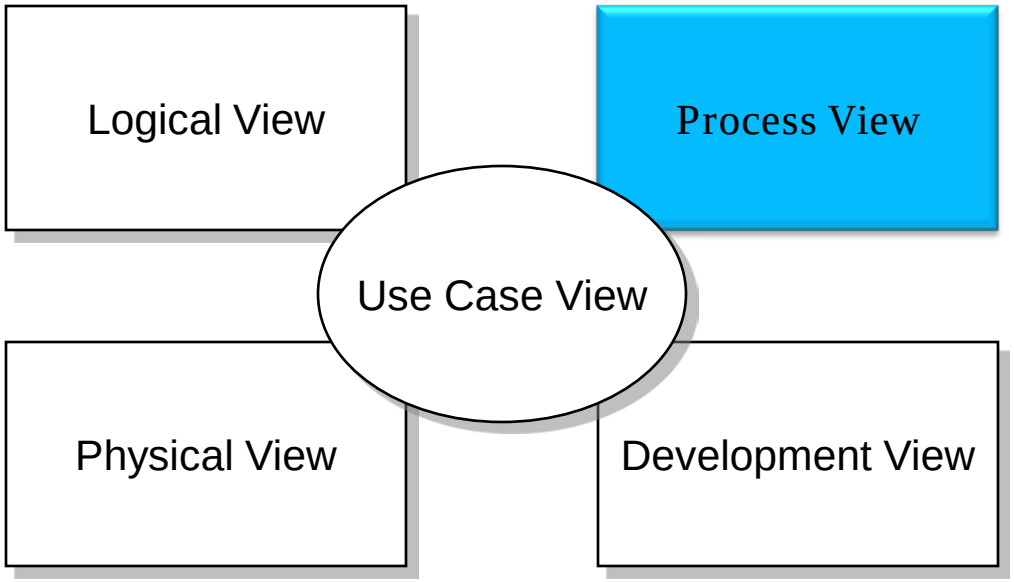
The Use Case view describes the functionality of the system being modeled from the perspective of the outside world - typically from the perspective of the users.

Model Views



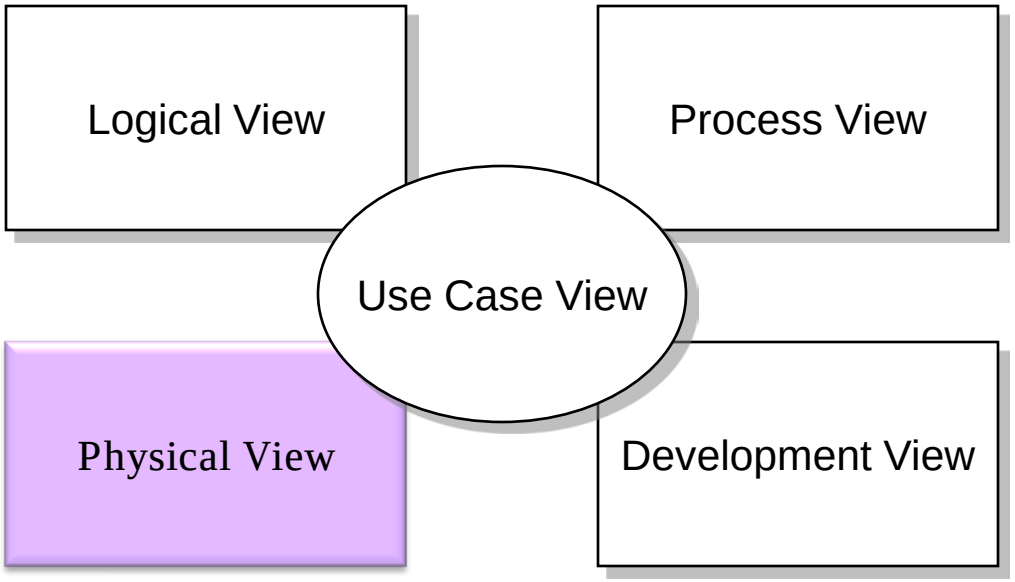
The Logical view describes the abstract concepts that comprise the parts of the system.

Model Views



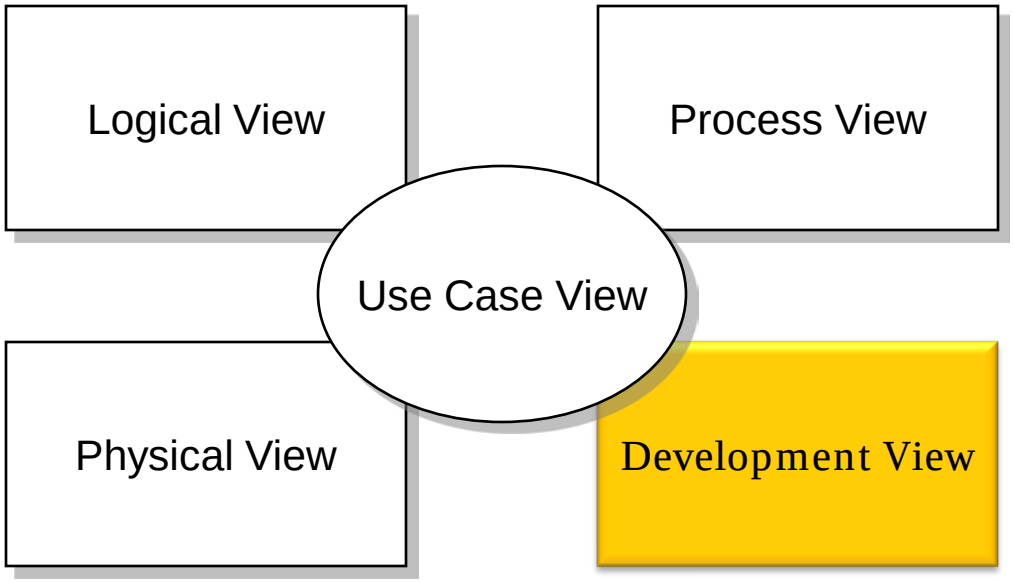
The Process view describes the process within the system – what the system must do to satisfy the use cases.

Model Views



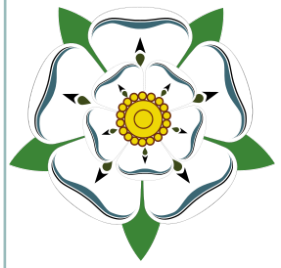
The Physical view describes how the system's design will be implemented as a set of deployment entities.

Model Views



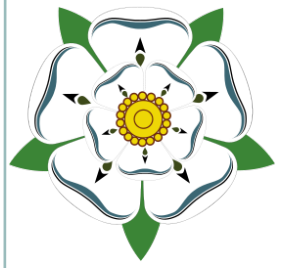
The Development view describes how the various parts of the system are organized into modules and components.

UML 2.0 Diagrams



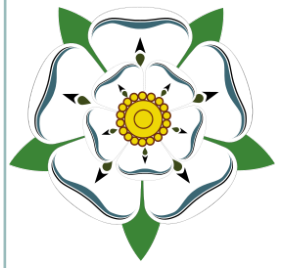
Activity Diagram	Business process, system logical flow
Class Diagram	Static elements of system classes and relationships
Communication Diagram	Dynamic communication between system elements
Component Diagram	High level decomposition of system elements
Composite Structure Diagram	Internal structure of classifiers
Deployment Diagram	Execution architecture of the system or subsystem
Interaction Overview Diagram	Variant of an Activity diagram showing control flow
Object Diagram	Depicts elements and their relationships at a specific instance in time
Package Diagram	Groups elements into organizational units
Sequence Diagram	Shows the time ordering of element interactions
State Machine Diagram	Depicts the states of an element and the transition between states
Timing Diagram	Depicts the change in state or condition of a classifier instance or role over time.
Use Case Diagram	Shows Actors and Use Cases and their relationships

Other Modeling Techniques



- Data Flow Diagram
  - Illustrates how information flows through a system
- Flow Chart
  - Form of activity diagram used in procedural code
- Entity Relationship Diagram
  - Used in database design
- Robustness Model
  - Validates that use case behavior is reasonable
- Free-form Diagrams
  - Ad hoc description of almost anything

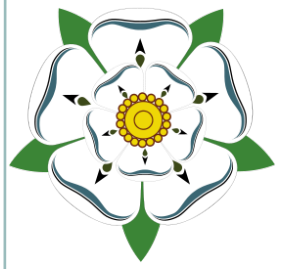
Model Formality



- Models may be drawn formally and precisely using tools such as Rose, Visio or myEclipse
- Models may also be drawn informally using whiteboards
- Level of formality will depend on the intent of the model
  - To document for posterity
  - To express transient ideas

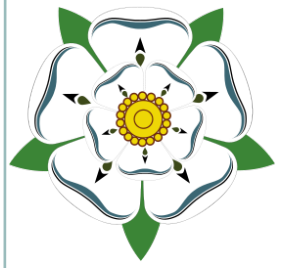
Wealth of Information

Travel Light



- You need far less documentation than you think
- Good documentation should:
  - maximize stakeholder investment
  - fulfill a purpose
  - describe information that is less likely to change
  - describe “good things to know”
  - be sufficiently accurate, consistent, and detailed
  - be sufficiently indexed

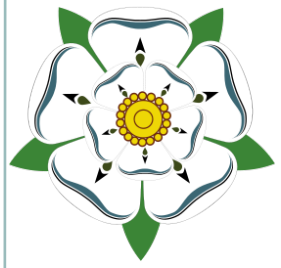
Don’t Rush to Expensive Tools



A "plain old whiteboard (POW)" is my favorite modeling tool, and I stand by my claim that whiteboards are the modeling tool with the greatest install base worldwide.

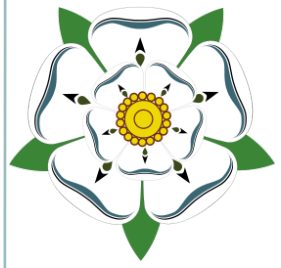
Scott Ambler

Why Document



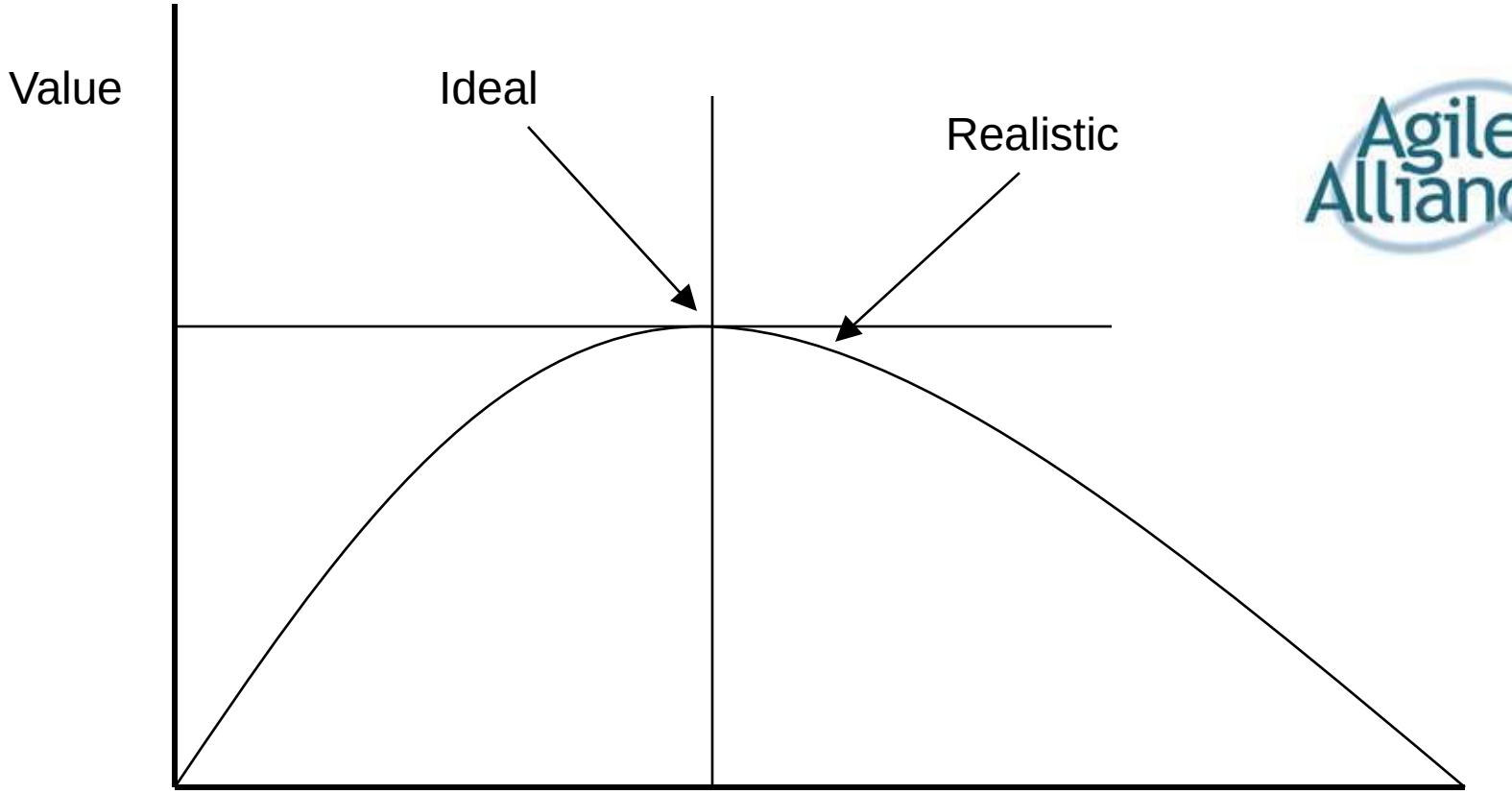
- Valid reasons to document include:
  - Your project stakeholders require (and pay for) it
  - The contract requires (and pays for) it
  - To support communication with an external group
  - To think something through
- Invalid reasons include:
  - Your manager said so!
  - You have some neat development tools
  - There is a section in a PMO template

No BMUF!



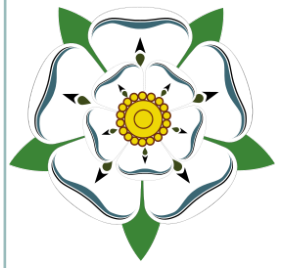
- Big Modeling Up Front is to be avoided
  - Mistakenly compare software development to civil engineering
  - Software cannot be fully specified before construction
  - Not all models have value
  - Typically only the well understood components can be modeled

Just Good Enough!



From Scott Ambler

Summary



- Models aid in understanding a concept
- Models aid in communicating a need
- Artifacts may be models as well as formal documents
- Models may be permanent or temporary
- Don’t waste time on models that have little value