



Maxim Systems' Use of IBM's Rational Unified Processing (RUP) and Unified Modeling Language (UML) to Develop a Satellite Communications Systems Model

Jack Turner
June 8, 2004

- Introduction
 - What a Modeling Development Process Should do
 - Define RUB: Rational Unified Process
 - Discuss RUP's Six Best Practices
 - Review RUP's Structure
 - A bit on Phases and Iterations
 - Types of Project Strategies
 - The RUP Strategy we choose

Next

- A Modeling Development Process Should:
 - Define the steps that lead to deliverables and who is responsible for them
 - Help control the project and reduce confusion
 - Help project management to resource, plan, and measure progress
 - Reduce risk
 - Make the model's development predictable, repeatable, and measurable

Next

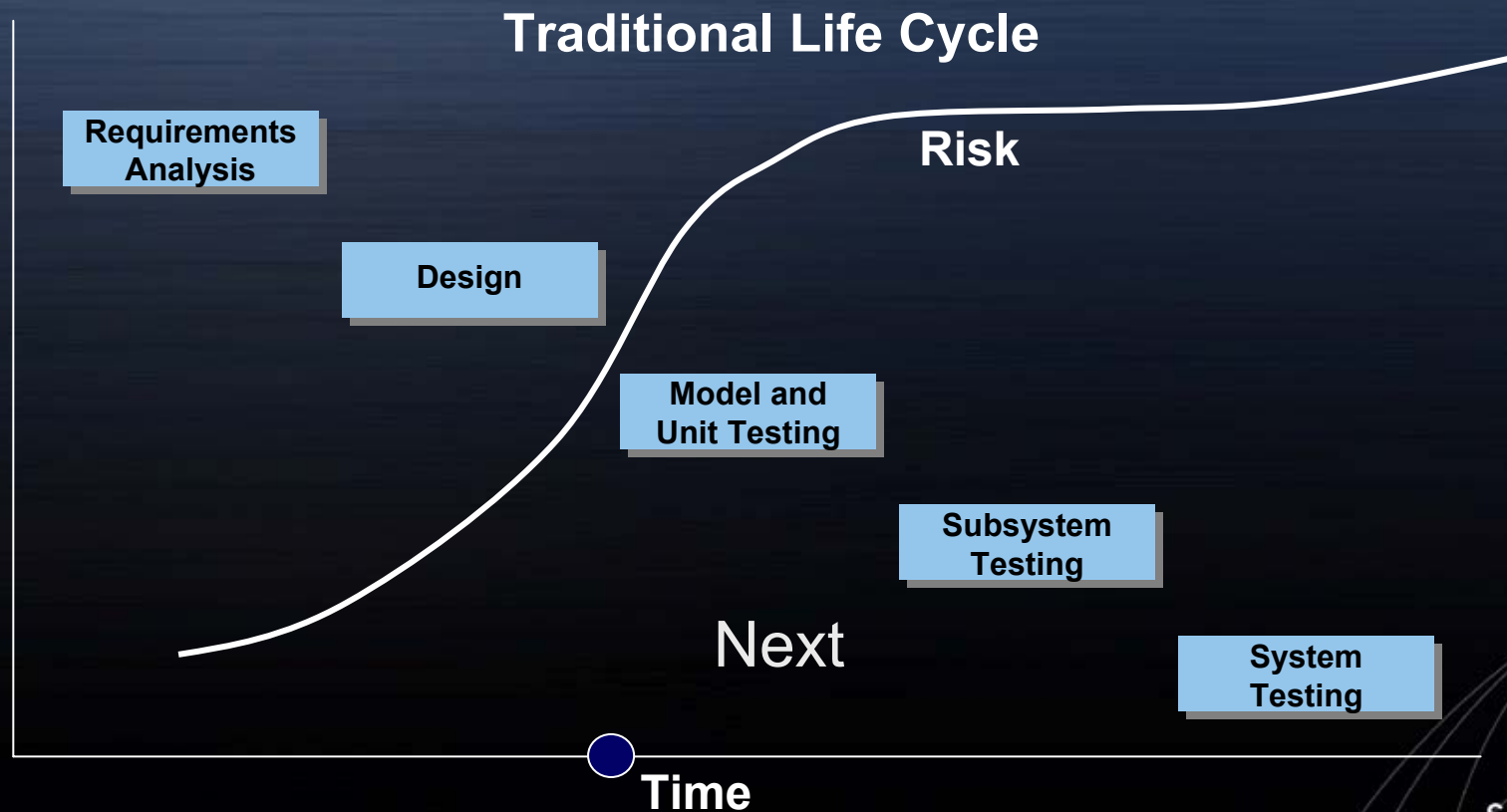
- What is the Rational Unified Process?
 - The Rational Unified Process (RUP):
 - Is a software and systems engineering process as well as a software and systems engineering process product
 - Searchable, knowledge-base delivered in a web site format
 - Provides project members with an online mentor covering the full lifecycle
 - Incorporates and presents best practices
 - Defines a project in terms of phases and iterations
 - Organizes a project into workers, artifacts, activities, and workflows

Next

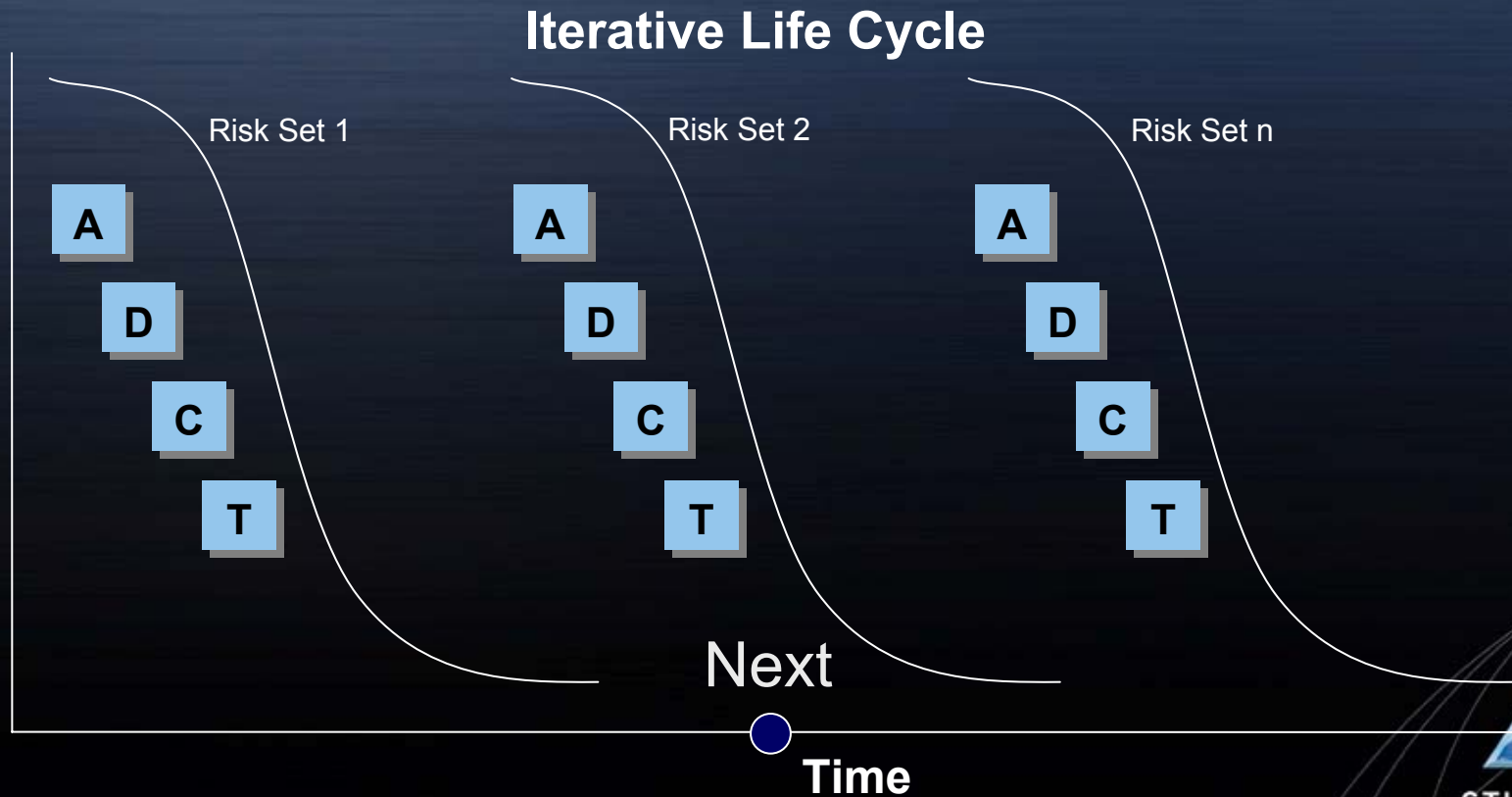
- RUP is based on Six Best Practices:
 - Develop iteratively
 - Manage requirements
 - Use component-based architectures
 - Visually model software and systems (in our case Model elements in STK, OpNet, or MATLAB elements)
 - Verify quality
 - Control changes

Next

- Practice 1: Develop Iteratively



- Practice 1: Develop Iteratively



- Practice 2: Manage Requirements
 - A requirement is a condition or capability a system must meet
 - The management of requirements encompasses:
 - Eliciting
 - Organizing
 - Documenting
 - the system's required functionality and constraints.
 - Requirements change! Requirements management requires evaluating these changes and assessing their impact throughout the Modeling Development lifecycle
 - Manage Requirements is where most projects fail

Next

- Practice 2: Manage Requirements
 - Managing Requirements Entails:
 - Analyzing the problem
 - Understanding the user needs
 - Defining the system
 - Managing scope
 - Refining the system definition
 - Building the right system
 - Verifying all requirements
 - Maintaining traceability

Next

- Practice 3: Use Component Architecture
 - A Model component is a nontrivial module, package, or subsystem
 - that fulfills a clear function, has a clear boundary, and can be integrated in a well-defined architecture
 - A system's architecture encompasses decisions about:
 - The organization of a Modeling system
 - The selection of the structural components and their interfaces
 - The behavior and collaborations among those components
 - The composition of these components into progressively larger systems

Next

- Practice 3: Use Component Architecture

- Advantages:

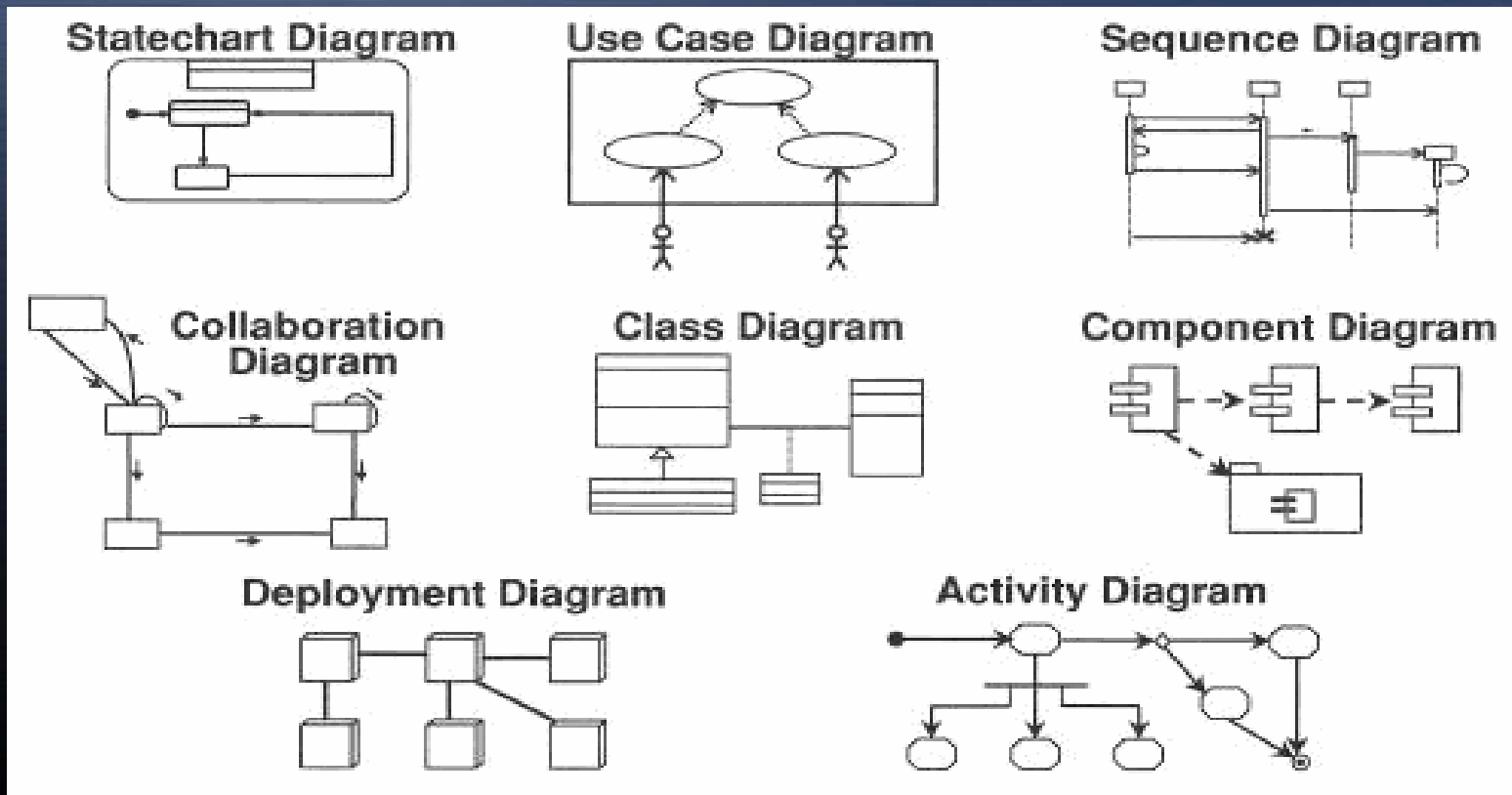
- The iterative approach allows us to progressively identify components – some we develop, some we buy
- In defining a very modular architecture, we identify, isolate, design, develop, and test well-formed modeling components
- By focusing on architecture in the early iterations, it allows us to articulate the structure of our system – the components and the ways in which they integrate
- We address complexity by segmenting the model into components, encapsulating the functionality and making them highly reusable
- The smaller the component, the more reusable, testable, and resilient

Next

- Practice 4: Model the Component Architecture Visually
 - Unified Modeling Language (UML):
 - Helps the development team visualize, specify, construct, and document the structure and behavior of a system's architecture
 - Lets you hide or expose details appropriate for the task
 - We do not want to make assumptions about the requirements in the coding stage – with UML, we can build the model's Component Architecture, assessing and verifying as we go
 - In building a visual representation of a model's Component Architecture, many different diagrams are needed to represent different views of the system
 - Helps to facilitate collaboration across groups – system engineering, software engineering, test engineers, etc. – all can share the same diagrams

Next

- Major UML Diagrams



Next

- Practice 5: Verify Quality
 - Software and Systems problems are 100 to 1000 times more costly to find and repair after delivery to the customer
 - Dimensions of software/systems quality:
 - Functionality: Does my system do what's required?
 - Reliability: Does my system respond consistently?
 - Performance: Does the system perform under production load?
 - Iterative development benefits testing
 - Test each iteration: functionality, reliability, and performance

Next

- Practice 6: Control Changes
 - An iterative development process must control, track, and monitor changes
 - Challenges:
 - Multiple developers, teams, sites, iterations, releases, projects, platforms
 - Recommended:
 - Audited baselines that include a bill-of-materials needed to determine the exact content of a release
 - Secure workspaces to assure isolation from changes in other workspaces
 - Control of all artifacts – models, code, documents, etc.

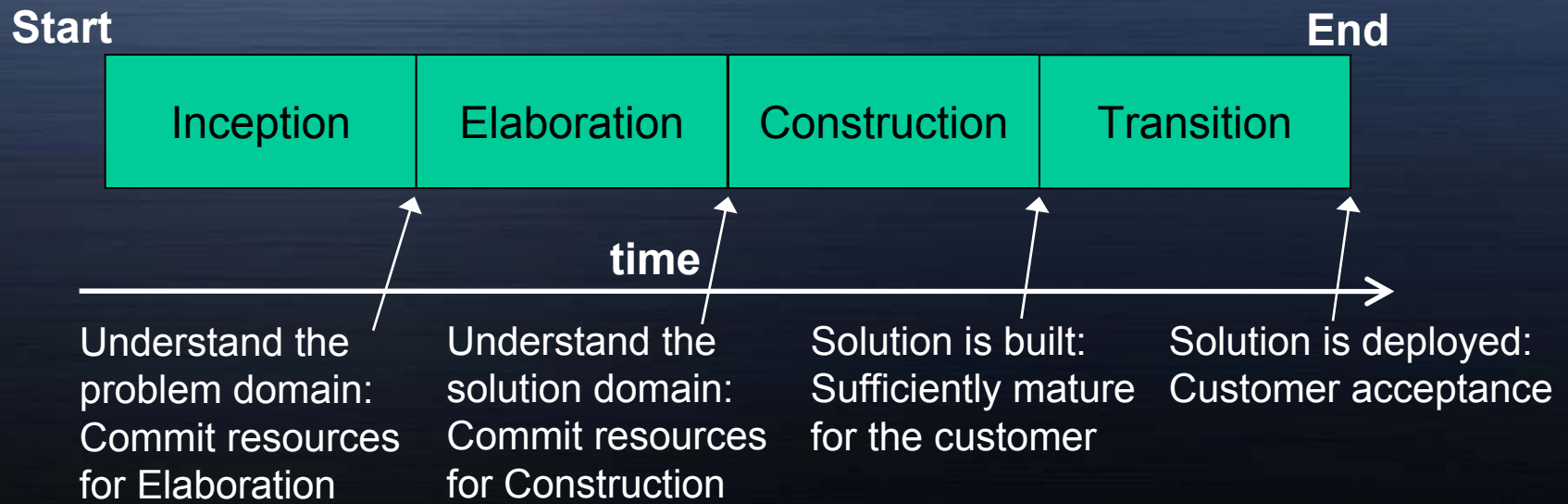
Next

- RUP Structure

- The RUP has a dynamic structure as well as a static structure
- The dynamic structure refers to the lifecycle aspect of the process – how the process rolls out within the duration of a project
 - Phases and iterations
- The static structure refers to the core process workflows, which group activities by nature
 - Workers, artifacts and activities

Next

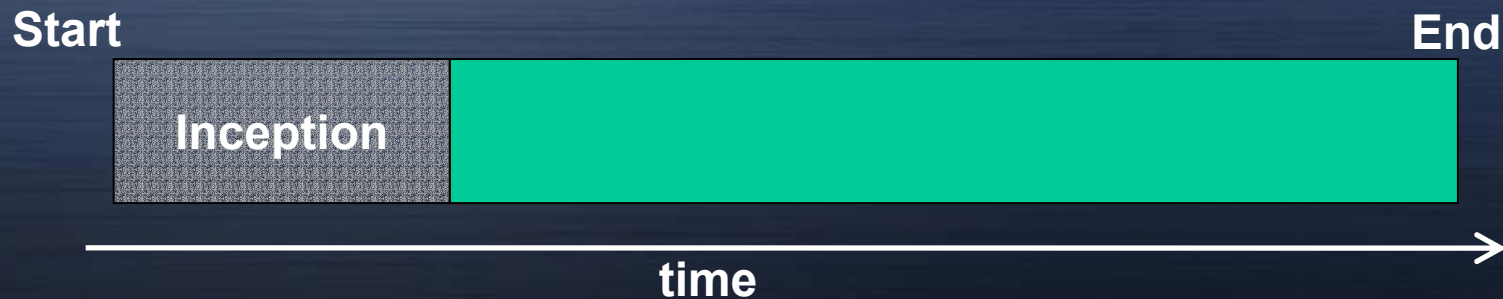
- Development Phases



- The phases of the Rational Unified Process were chosen such that the phase boundaries correspond to significant decision points in the life of a project

Next

- Inception Phase



The Inception Phase: Get the “Go / No Go” decision

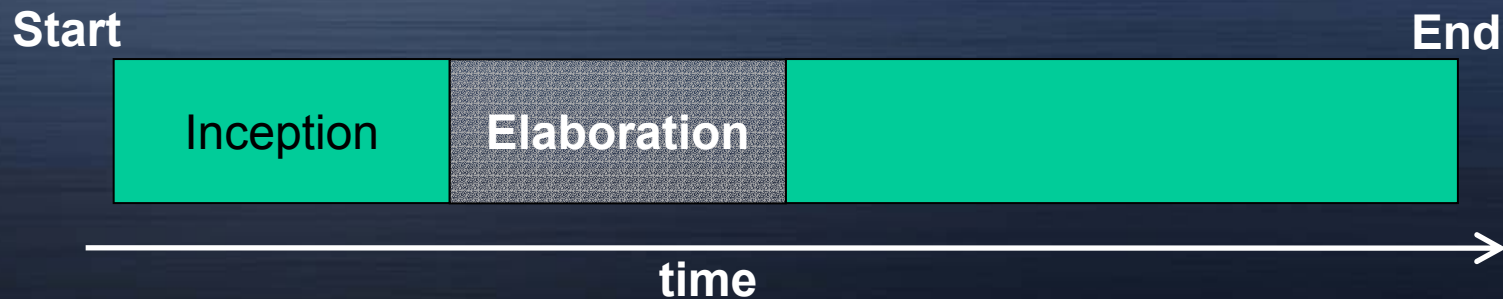
- Objectives:
 - Establish project scope and boundary conditions
 - Determine use cases and primary scenarios
 - Estimate overall cost and schedule
 - Exhibit a candidate architecture
 - Identify potential risks

Next

- Inception Phase Outcomes
 - Vision document
 - General vision of the core project's requirements, key features, and main constraints
 - Use case model survey
 - Lists all use cases and actors that can be initially identified
 - Initial business case
 - Business context, success criteria, financial forecast
 - Risk assessment
 - Initial project plan
 - Shows phases and iterations

Next

- Elaboration Phase



The Elaboration Phase: The “*Architectural Milestone*”

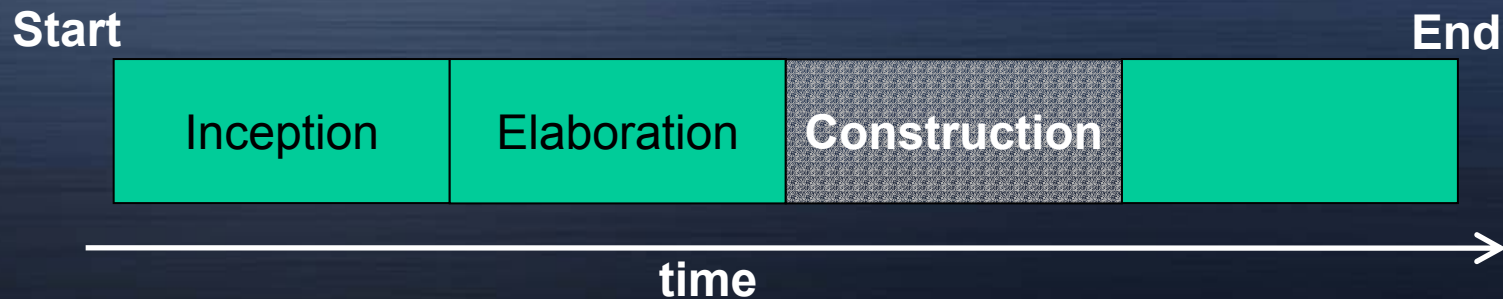
- Objectives:
 - Define, validate, and baseline the architecture as rapidly as practical
 - Evaluate the stability of the product vision
 - Demonstrate that the baseline architecture will support the vision at a reasonable cost in a reasonable period of time

Next

- Elaboration Phase Outcome
 - A use-case model (at least 80% complete)
 - Supplementary requirements
 - A software/systems architectural description
 - A preliminary executable architecture
 - All high risk items have been tackled at this point
 - A revised risk list and a revised business case
 - A development plan

Next

- Construction Phase



The Construction Phase: *“From Design to Implementation”*

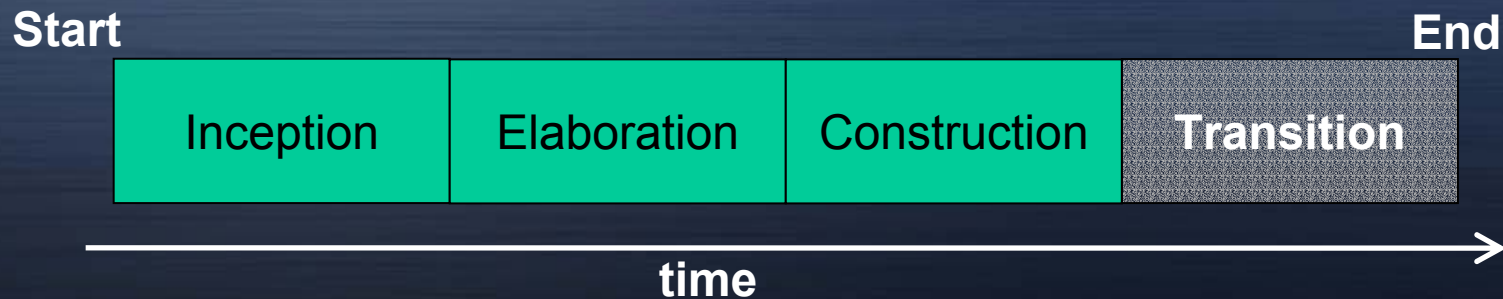
- Objectives:
 - Minimize development costs by optimizing resources
 - Achieve adequate quality
 - Achieve useful versions (alpha, beta, and other test releases)

Next

- Construction Phase Outcome
 - The the modeling product
 - Integrated on the indented platform(s)
 - Revised architectural models
 - User manuals as necessary
 - A description of the current release

Next

- Transition Phase



The Transition Phase: “*Deliver the product and ...*”

- Objectives:
 - Achieve user self-supportability
 - Achieve stakeholder concurrence that delivered baseline is complete
 - Achieve final baseline as rapidly and cost effectively as possible
- Outcome:
 - The completed system

Next

- Summary of Phases

- Inception phase:

- Focus is on understanding requirements and scope of the development effort

- Elaboration phase:

- Focus is on establishing a sound architectural foundation to support the requirements, developing the project plan, and eliminating the project's high risk elements

- Construction phase:

- Focus is on fleshing out the design and on implementation – evolving the initial prototype into the first operational product

- Transition phase:

- Focus is on delivering the product to the user community, fixing bugs, training users, adjusting features, and adding missing elements

Next

- Iterations

Inception		Elaboration		Construction		Transition	
Preliminary Iteration	Architect Iteration	Architect Iteration	Develop Iteration	Develop Iteration	Develop Iteration	Transition Iteration	Transition Iteration

- **Each iteration is a mini-waterfall and involves activities of:**
 - Requirements
 - Analysis and Design
 - Implementation
 - Test
- **From one iteration to the next and from one phase to the next, the focus on the activities will change**

Next

- Types of Project Strategies
 - The length of time, number of iterations, resources, etc. will vary based upon:
 - Type of project
 - Size of project
 - Experience level
 - Customers

Next

- Strategy – Incremental

I	E	Construction								Transition	
#1	#1	#1	#2	#3	#4	#5	#6	#7	#8	#1	#2

- Characteristics:
 - Stable architecture
 - Enhancement is understood
- Process tailoring:
 - Multiple iterations in construction
 - Little time spent in inception, elaboration, or transition

Next

- Strategy – Evolutionary

I	Elaboration						Construction			Transition	
#1	#1	#2	#3	#4	#5	#6	#1	#2	#3	#1	#2

- Characteristics:
 - No existing architecture
 - Customer doesn't want partial deliverables
- Process tailoring:
 - Multiple iterations in elaboration and construction
 - Few iterations in inception and transition

Next

- Strategy – Incremental Delivery

I	E	C	Transition								
#1	#1	#1	#2	#1	#2	#3	#4	#5	#6	#7	#8

- Characteristics:
 - Very stable architecture
- Process tailoring:
 - Early transition and it has the most iterations
 - Early usage of the application

Next

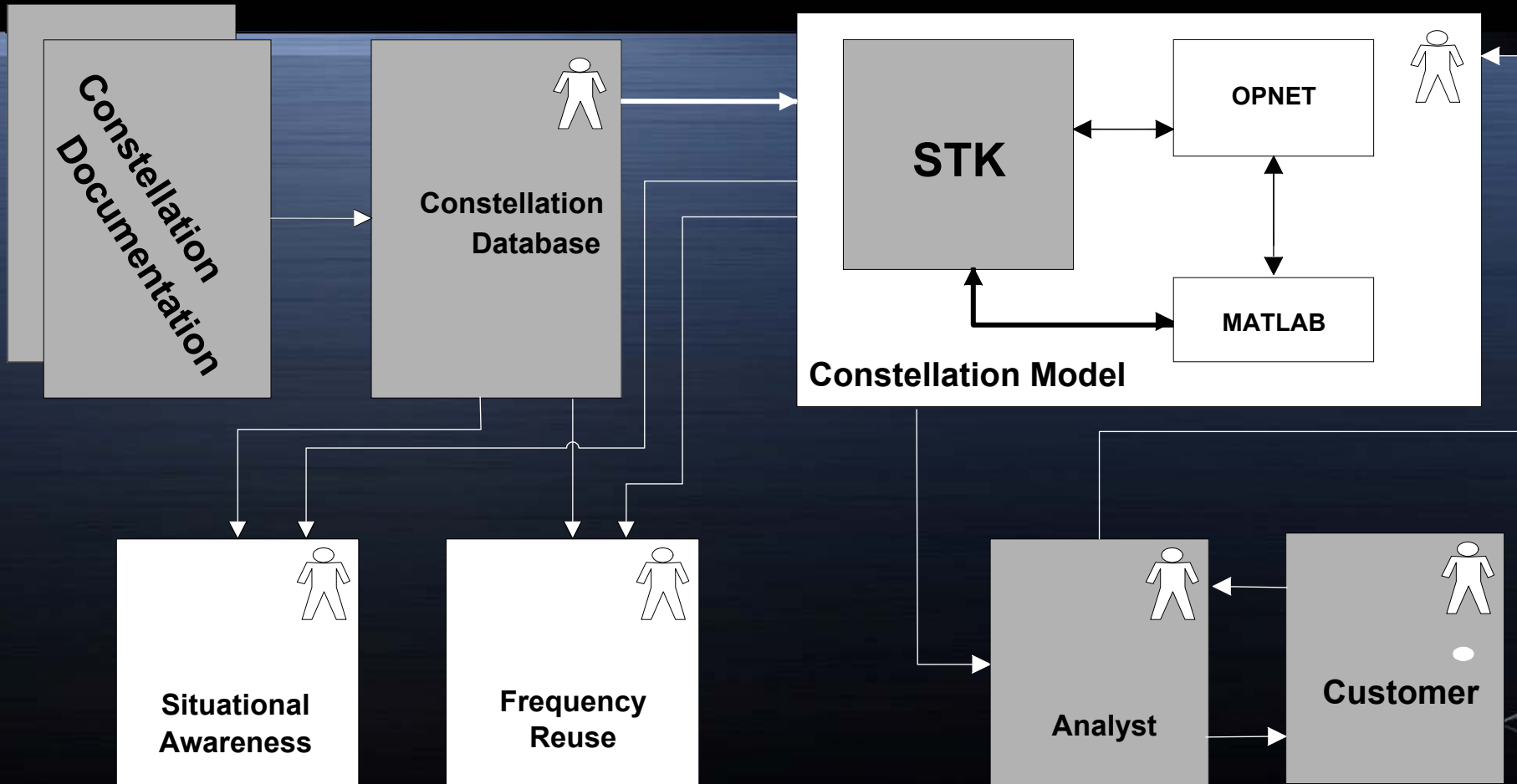
- For the Satellite Communications Systems Model



- We have chose an Evolutionary Approach
- Currently emerging into the second stage of the Elaboration
- Characteristics:
 - There was no existing architecture
 - Customer doesn't want partial deliverables
 - Multiple iterations in elaboration and construction
 - One iteration in inception with Multiple iterations in Elaboration
 - Vision Document and Basic Architecture are understood.

Next

Maxim Systems Satellite Communications Systems Model Development using RUB and UML



- Conclusion
 - What a Modeling Development Process Should do
 - Define RUB: Rational Unified Process
 - Discuss RUP's Six Best Practices
 - Review RUP's Structure
 - A bit on Phases and Iterations
 - Types of Project Strategies
 - The RUP Strategy we choose