

*Exceptional service in the national interest*



# US NDC System Modernization

## Rational Unified Process Overview

Ryan Prescott

9 September 2013



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

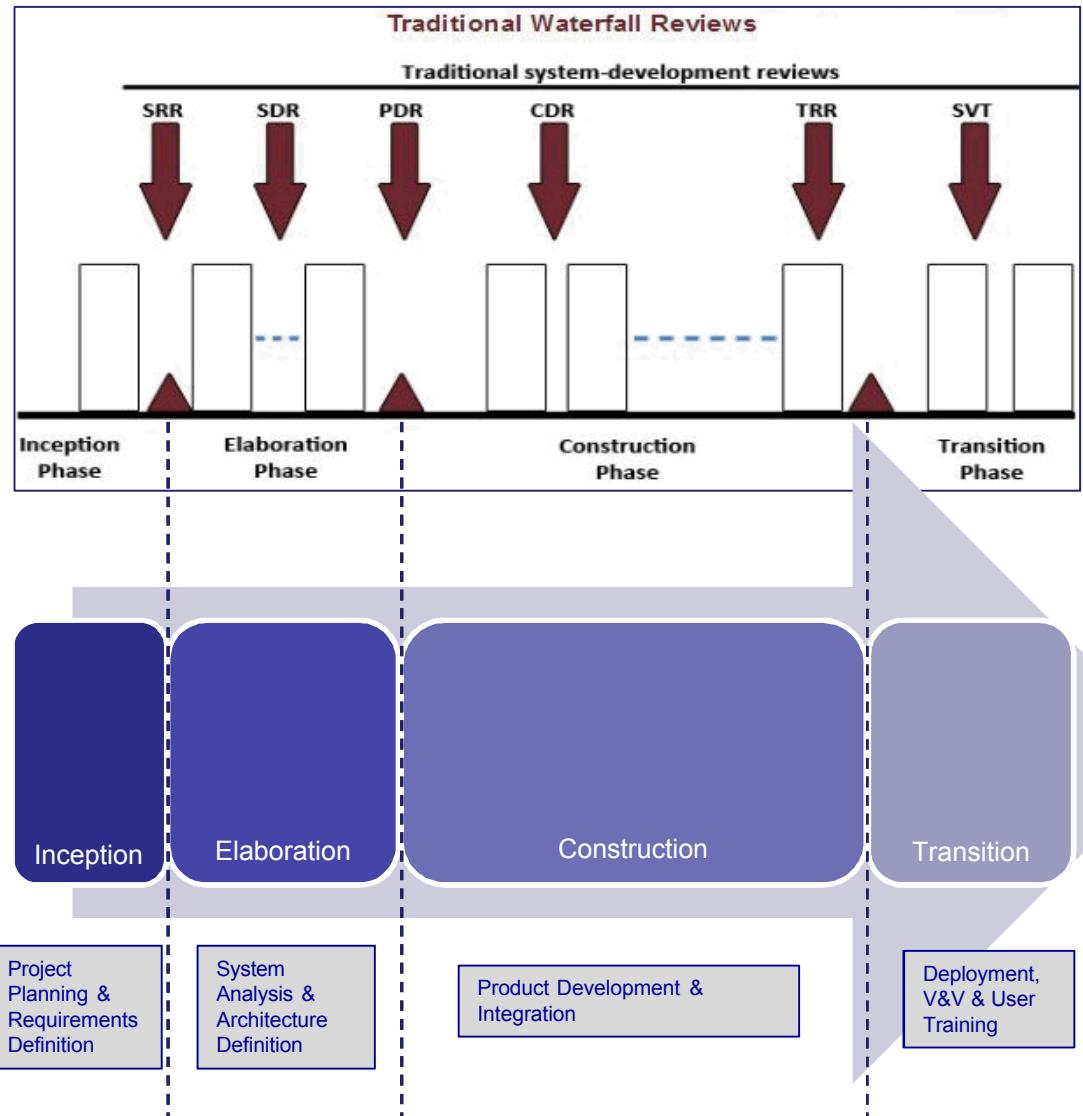
# Background

- The US NDC modernization project is following a software engineering process based on the Rational Unified Process (RUP)
  - Demonstrated success on large (1+ Million LOC), complex, mission-critical remote sensing projects
- RUP is an adaptable software engineering process framework
  - Incremental & Iterative
  - Architecture-centric
  - Use-case driven
  - Customizable based level of project formality

# Core Elements of RUP

- Iterative development with parallel analysis, design, implementation, and testing activities
- Product-focused, incremental development with releases of a testable product at the end of each development iteration
- Risk-driven development addressing high-risk items first
- Early development of executable prototypes validating the system architecture prior to full-scale development
- Use case analysis to support requirements definition and traceability through system design, implementation & test
- Architecture modeling promotes consistency between design and implementation

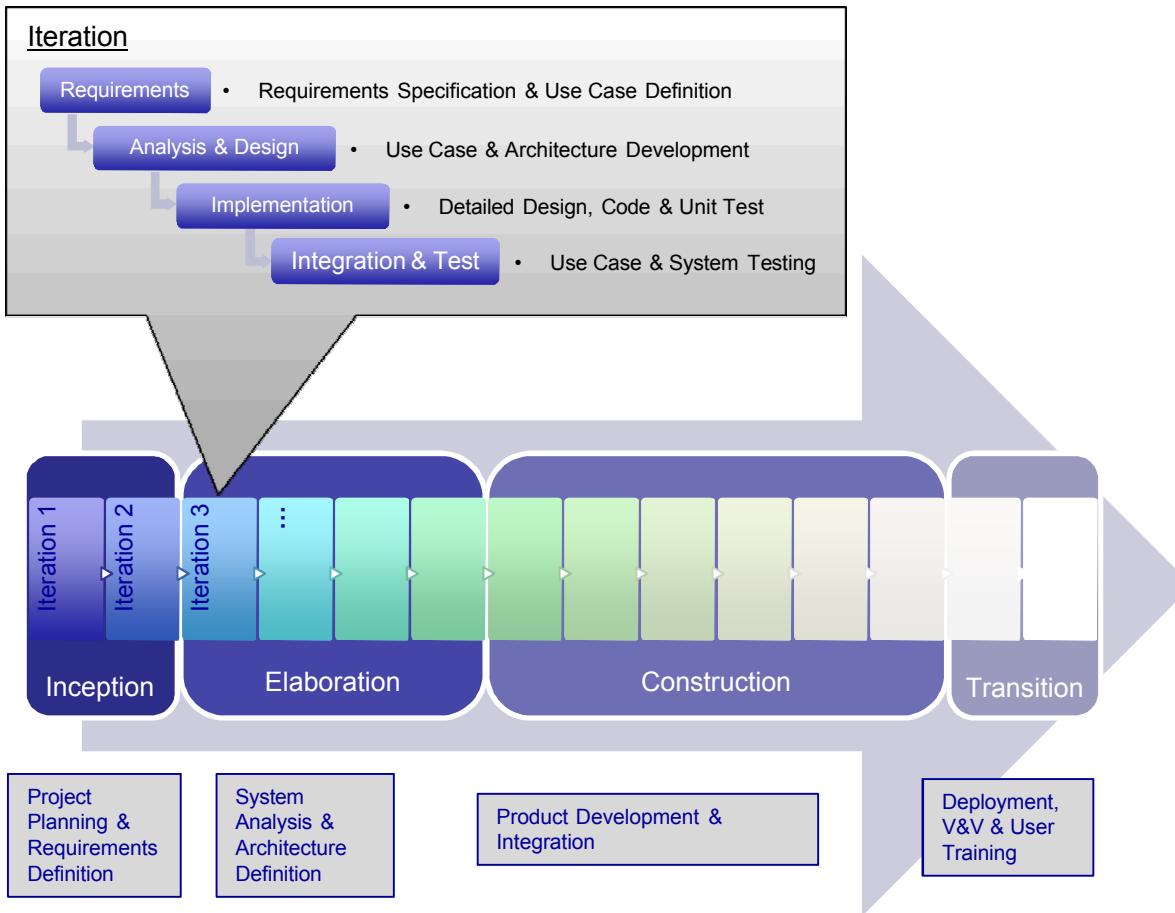
# RUP Project Lifecycle - Phases



The project is divided into four development phases:

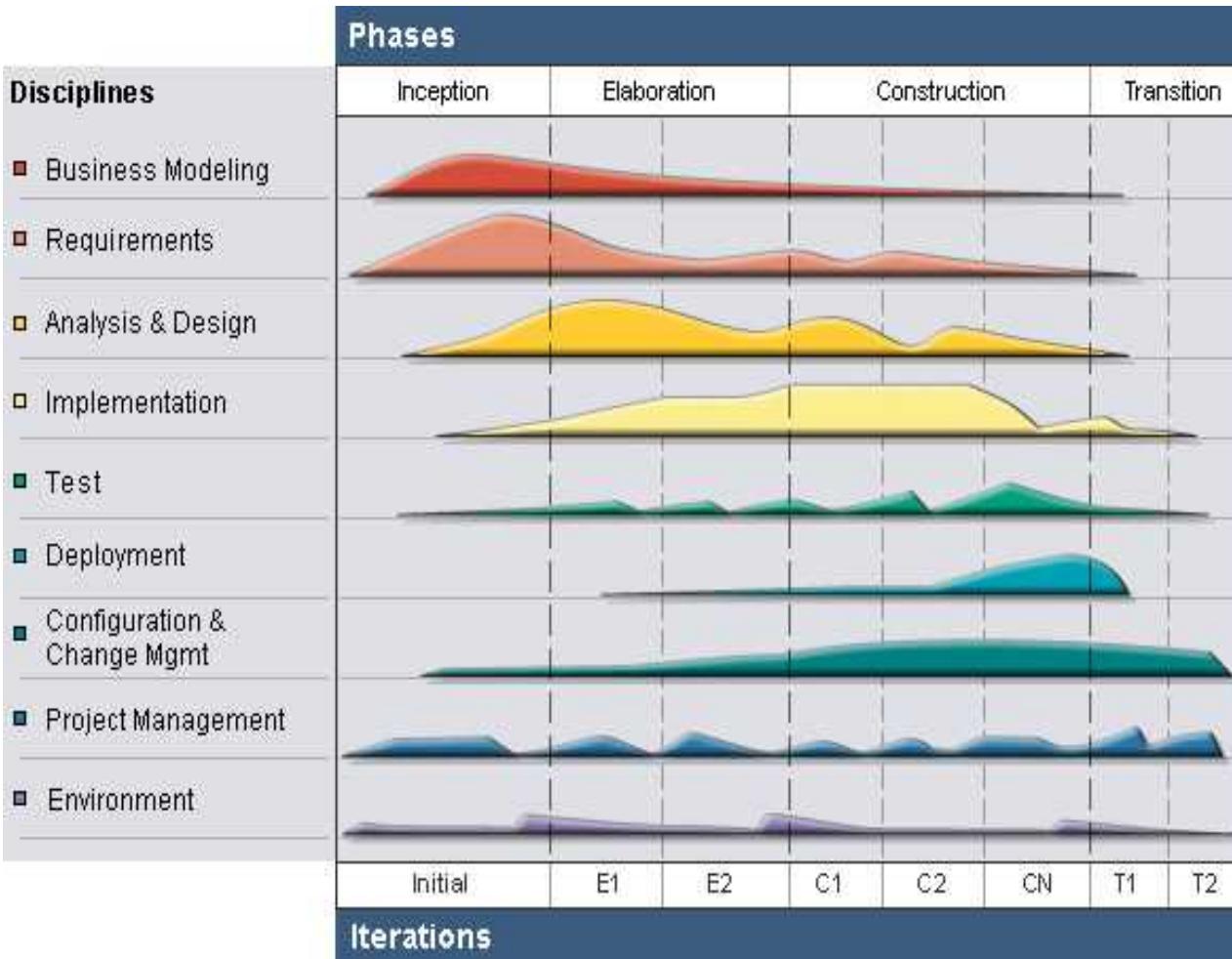
- Inception**
  - Define project scope
  - Estimate cost and schedule
  - Identify risks
  - Define system requirements
    - Specifications and use cases
  - Prepare the supporting environment
- Elaboration**
  - Elaborate system requirements
    - Specifications and use cases
  - Define the architecture
    - Use case realizations
    - Demonstrate an executable prototype
  - Develop a detailed plan for the Construction phase
- Construction**
  - Develop, integrate and test the product
  - Document the product
- Transition**
  - Complete user acceptance testing
  - Deploy the product to the user community
  - Train user community

# RUP Project Lifecycle - Iterations



- **Iteration:** “A development loop ending in a release of a subset of the final product”
  - Fixed duration – length determined at project inception (e.g. 6 months)
  - Schedule driven
- Activity within an iteration follows a traditional model
  - Requirements, Design, Implementation, Test, Documentation
- The focus within an iteration changes over the project’s life
  - Early iterations are focused on planning and high risk work
  - Later iterations are focused on development and delivery
- **Iteration Reviews**
  - Current system capability is demonstrated to the customer

# RUP Disciplines Across the Lifecycle



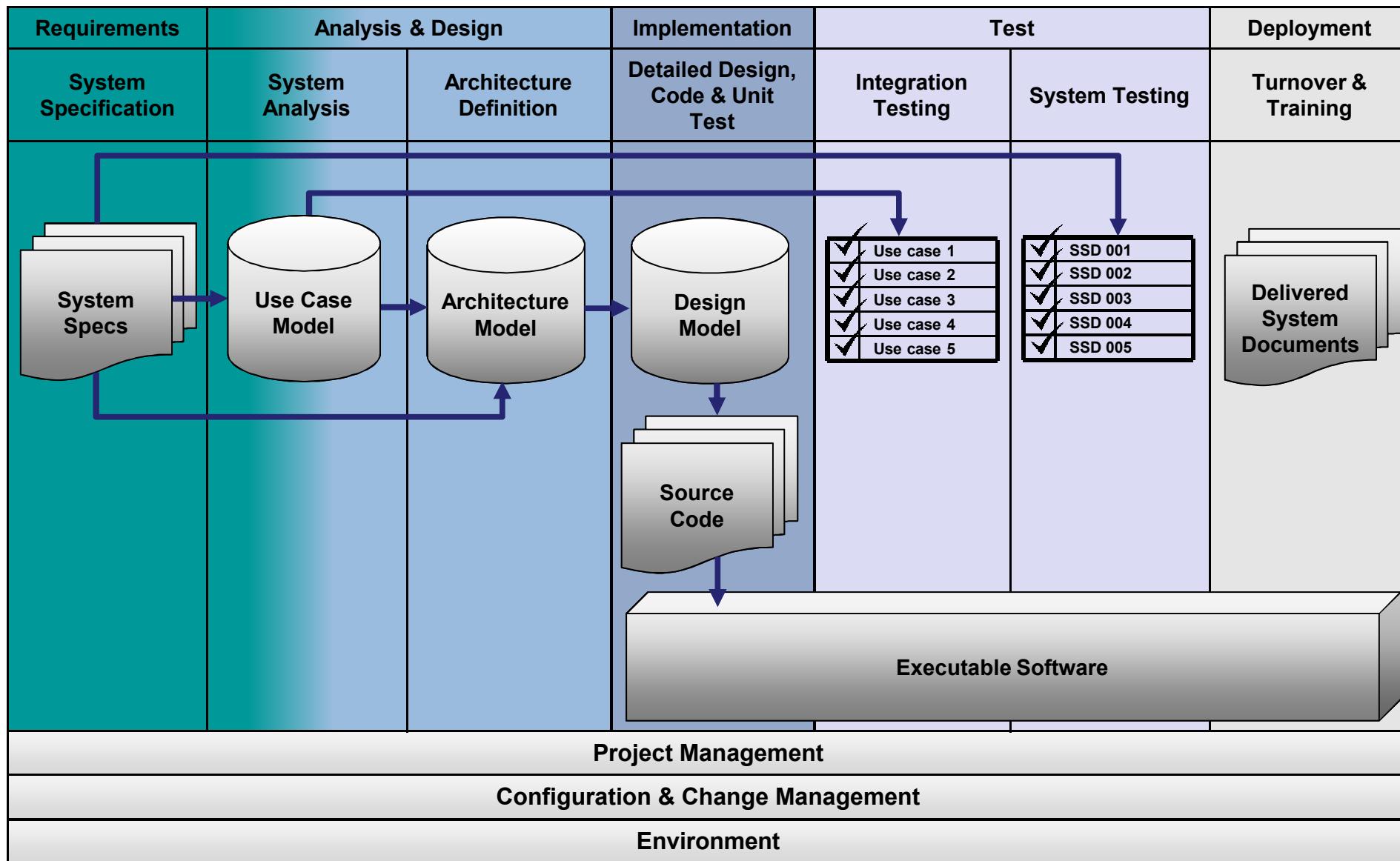
## Best Practices

- Develop iteratively to reduce risk
- Manage requirements
- Employ a component-based architecture
- Model software visually
- Continuously verify quality
- Control changes

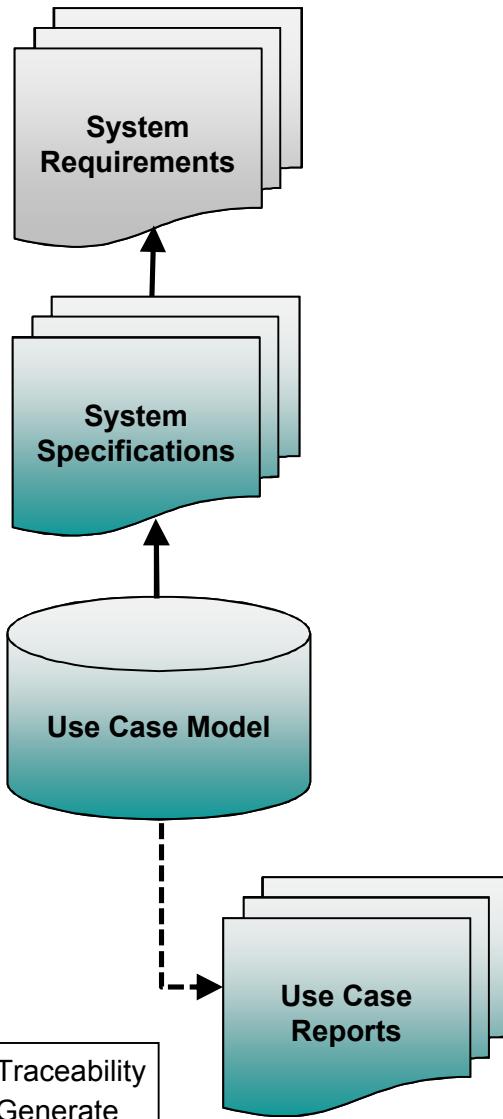
## Key Principles

- Adapt the Process
- Balance Competing Stakeholder Priorities
- Collaborate Across Teams
- Demonstrate Value Iteratively
- Elevate Level of Abstraction
- Focus Continuously On Quality

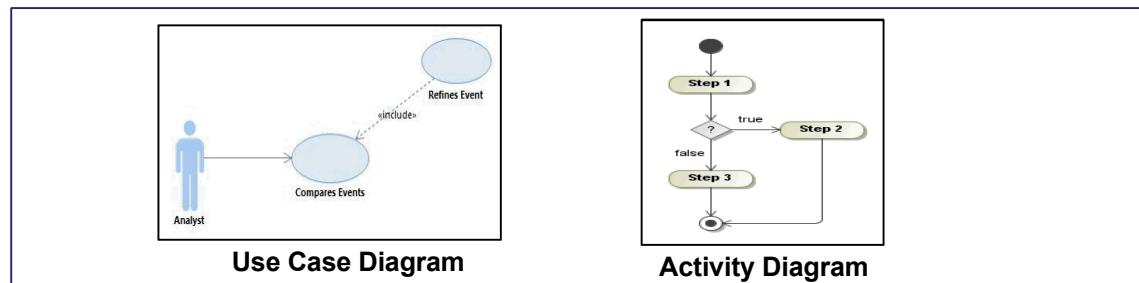
# Workflow/Artifacts Each Iteration



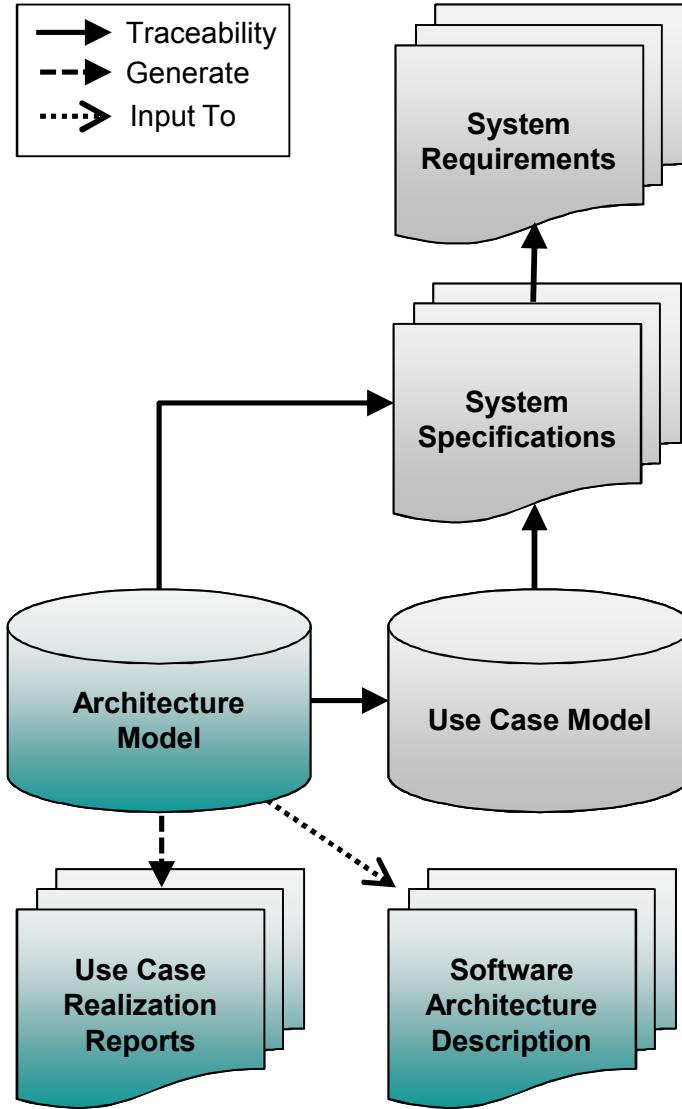
# System Specification & Use Cases Analysis



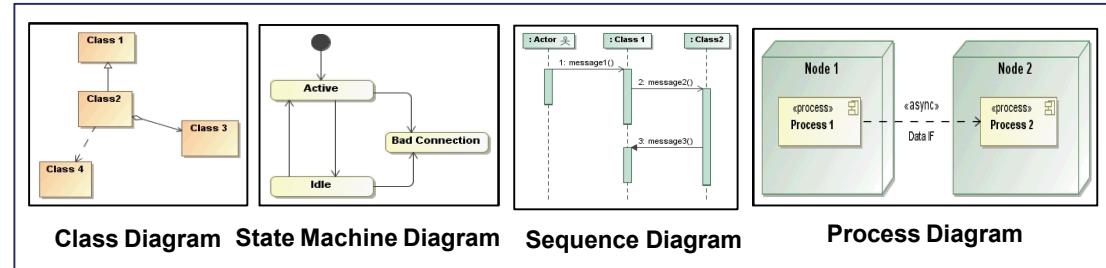
- System specifications are defined based on the customer's requirements
  - Requirements reflect the customer's operational capability needs
  - Specifications are developed by the project team as an elaboration of the requirements
- Use cases are developed to further define the required behavior of the system
  - Each use case describes the interactions of an actor (user or external interface) with the system to perform a single function
  - No specific architecture or implementation is expressed
- System specifications are mapped to the use cases, establishing traceability
- Use cases form the basis of the system architecture definition
- Use cases are modeled using UML
  - Use Case documents are generated from the model



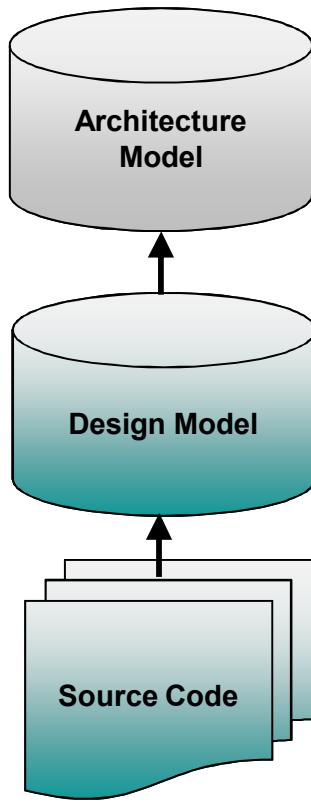
# System Architecture Definition



- System architecture is defined based on the use case model
  - The architecture model includes a Use Case *Realization* (UCR) for each use case
  - UCRs describe the system architecture supporting the associated use case (transition from “what” to “how”)
- UCRs are modeled in UML
  - UCR documents are generated from the model
  - Design artifacts such as the System Architecture Document incorporate content from the architecture model
- System specifications are mapped to the UCRs, establishing traceability

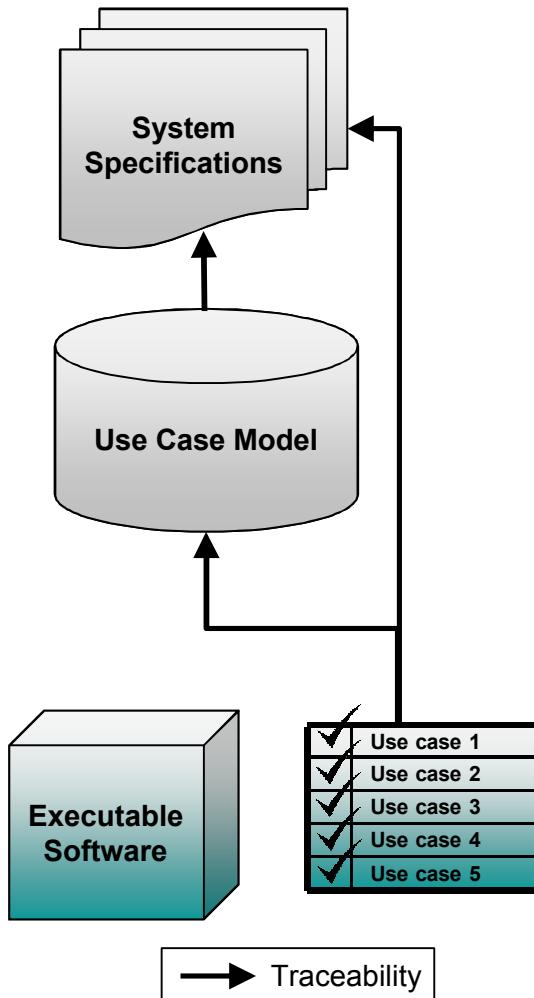


# Implementation



- Implementation includes:
  - Detailed design modeled in UML
    - Pre-Implementation Design
    - Optional Code generation from the design model
  - Implementation of logic in software
  - Unit testing
  - Code Reviews
  - Testing developed code on test beds

# Integration Testing



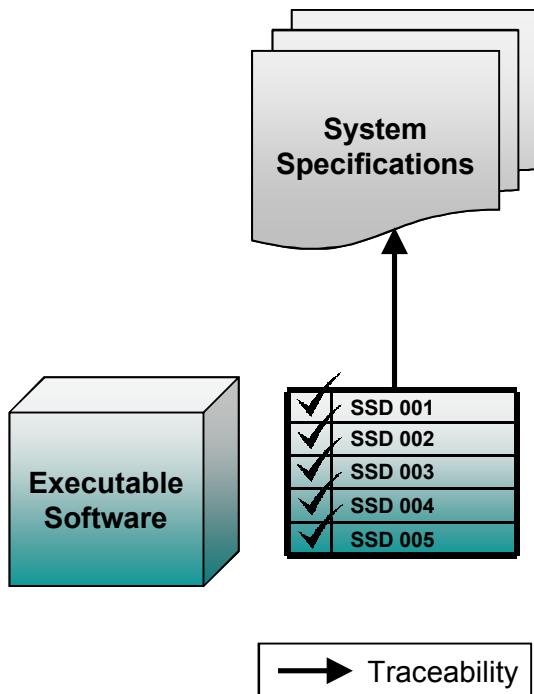
## 1. Use Case Testing

- Test against Use Case Descriptions and UI Prototypes
- Regression testing

## 2. Soak testing

- Execute the system for minimum required duration to ensure correct system performance
- Test system features and constraints that can only be exercised with continuous operation
- Monitor CPU, Memory, IO, etc.
- Supported by developers

# System Testing



- System is built and delivered to system testbed every iteration
- System Testing
  - Verifies functionality against System specifications that are ready to be verified
  - Regression tests previously verified system specifications
- Conducts acceptance testing witnessed by the customer

# Deployment

- User Manual and other documentation
- User Training
- Delivery of hardware and software
- Customer acceptance testing
- O&M post-deployment
  - Operational site data available on test network
  - Reach-back capability to bring logs from sites, replay as if live