ACE4 Working Group Session, March 29, 2006

Moving from 4+1 to the 5+2 View Modeling of Architecture: ULCM^{sm} Views as Extensions of Architectural Views

Dr. Peter Hantos

The Aerospace Corporation

THE AEROSPACE CORPORATION

Acknowledgements

- **This work would not have been possible without the following:**
  - ❖ **Feedback**
    - **Dr. Sergio Alvarado**, Software Architecture & Engineering Department
    - **Suellen Eslinger**, Software Engineering & Acquisition Subdivision
    - **Dr. Leslie J. Holloway**, Software Acquisition and Process Department
  - ❖ **Funding source**
    - The Aerospace Corporation Independent Research & Development (**IR&D**) Program

Agenda

- **Problem/Goal/Objectives**
- **The 4+1 View Model of Software Architecture**
- **Evolution of Architecture Artifacts in Iterative Development**
  - ❖ Snapshots of views associated with architecture evolution
  - ❖ Concurrent development of artifacts in Iterative Development
- **Unified Life Cycle Modeling (ULCM^{sm})**
  - ❖ ULCM – The 10,000-foot view
  - ❖ The Spiral Model's ULCM generator view using UML activity diagrams
  - ❖ ULCM enactment view of pre-planned increments
- **Integrating the Views - A Little "View Algebra"**
- **Conclusions**
- **Acronyms**
- **References**
- **Backup Slides**

® UML is registered in the U.S. Patent and Trademark Office by the OMG
^{sm} ULCM is Service Mark of The Aerospace Corporation

Problem Statement

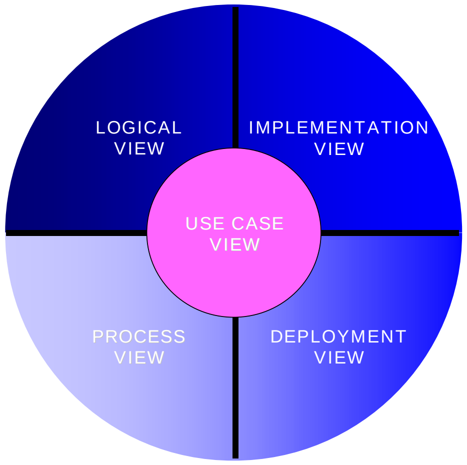
- **Process Architecting is not well aligned with System and Software Architecting**
  - ❖ Separately and together they are high-leverage activities of acquisition and development
  - ❖ Alignment would result in
    - – Earlier identification and better management of problems
    - – Ultimately, reduced life cycle cost

Presentation Goal and Objectives

- **Goal**
  - ❖ Show a key aspect of synergy between Architecture Views and Unified Life Cycle Modeling (ULCM) Views
- **Objectives**
  - ❖ Review how Architecture Views evolve during acquisition and development
  - ❖ Present ULCM Views*
  - ❖ Integrate ULCM Views with P. Kruchten's 4+1 View Model of Architecture**

Sources:
****   (Hantos 2005), (Hantos 2006)***
***** (Kruchten 1995)***

The 4+1 View Model of Software Architecture



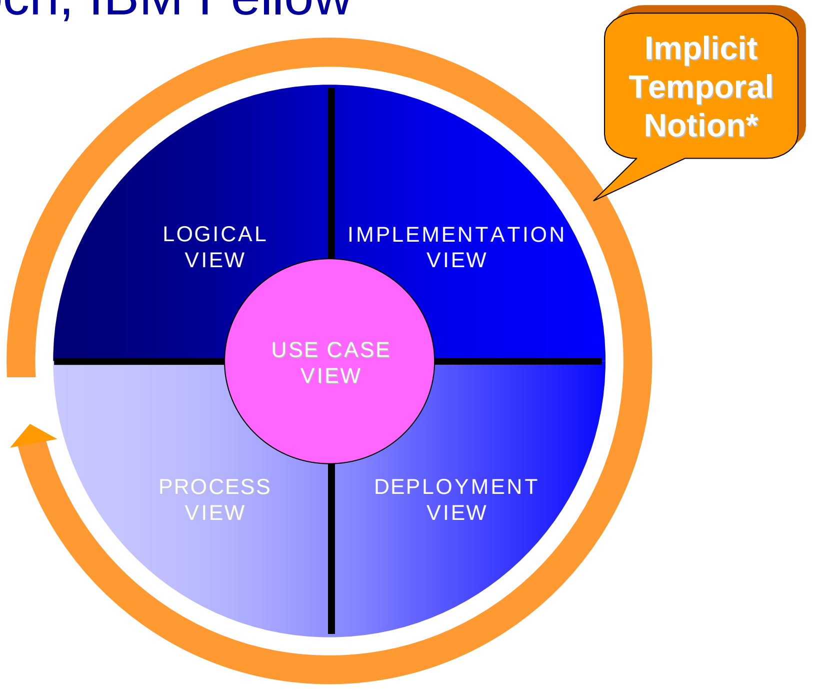
VIEW*	DETAIL	STAKEHOLDERS	COMMENTS
LOGICAL	Subsystems, Classes	End User	Functionality
IMPLEMENTATION	Components, Packaging, Layering	Developer, Project Manager	Used to be called Development View
DEPLOYMENT	Topology, Mapping to Platforms	System Engineer	Used to be called Physical View
PROCESS	Performance, Throughput, Concurrency	System Integrator	It is a Computer Engineering term
USE CASE	Architecture Discovery, View Validation	Analyst, Tester	Sometimes called Scenarios

** Diagram and view names based on (Kruchten 1998). The author apparently instituted some name changes for selected views since the first publication (See Kruchten 1995).*

Challenging Booch …

- **"The 4+1 view model has proven to be both necessary and sufficient for most interesting systems"**

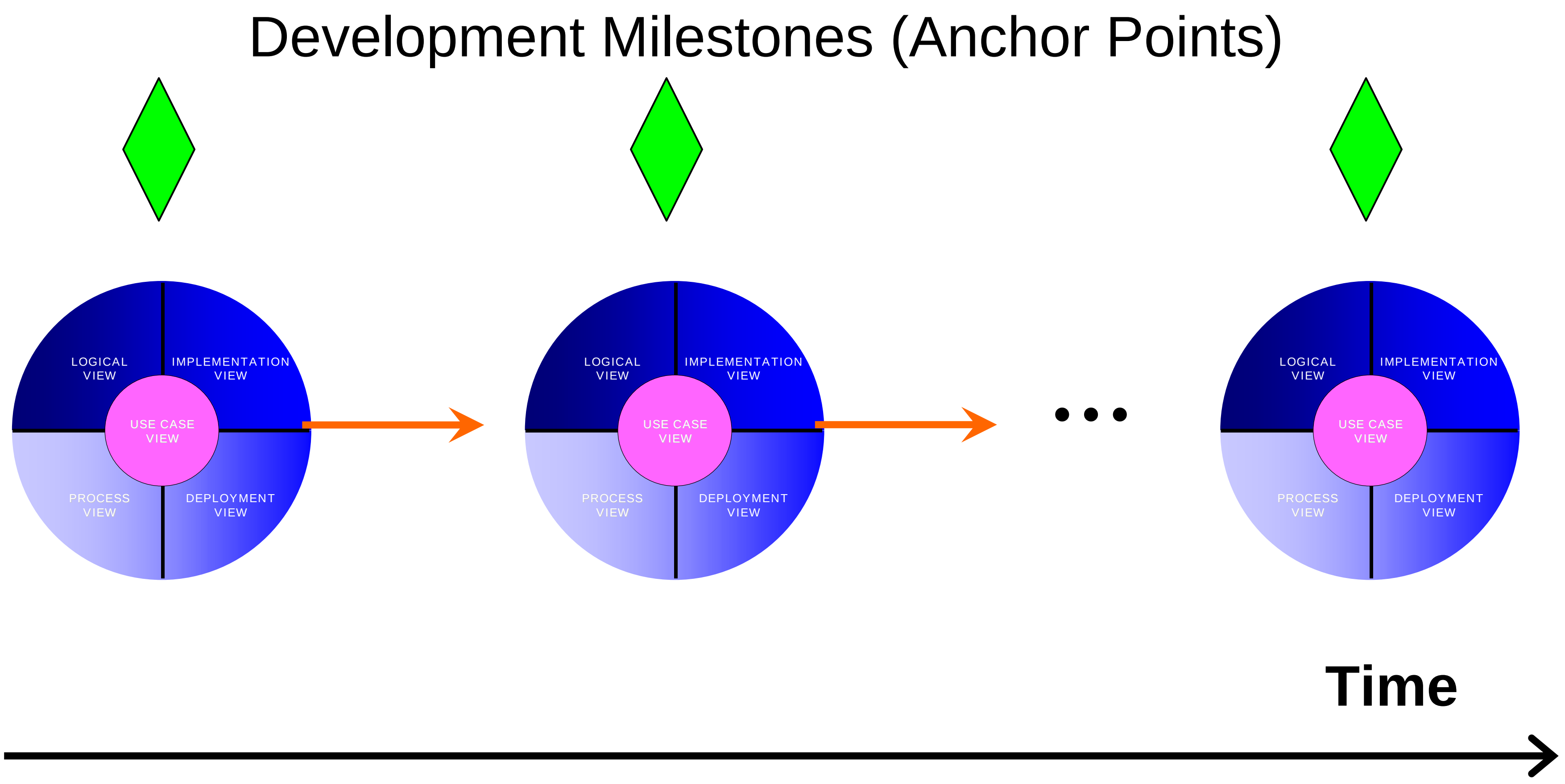
  --- Grady Booch, IBM Fellow



This implicit temporal notion does not reflect the complete dynamics

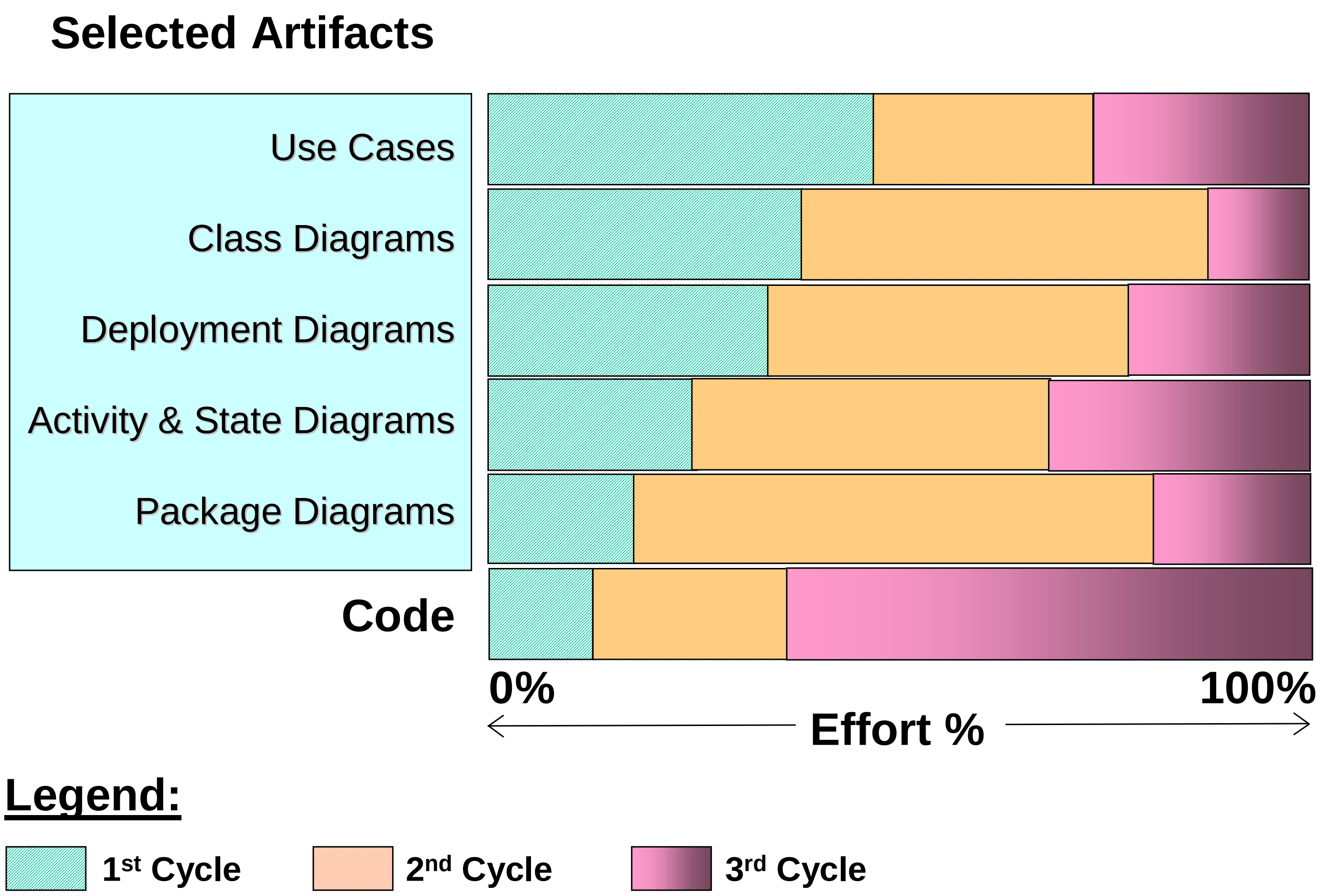
**** Emphasis by Hantos***

Snapshots of Views Associated With Architecture Evolution



Architecture evolution is reflected in view evolution

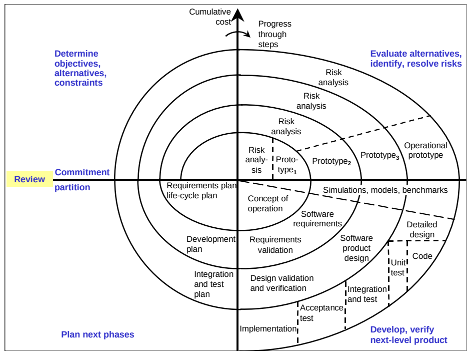
Concurrent Development of Artifacts in Iterative Development



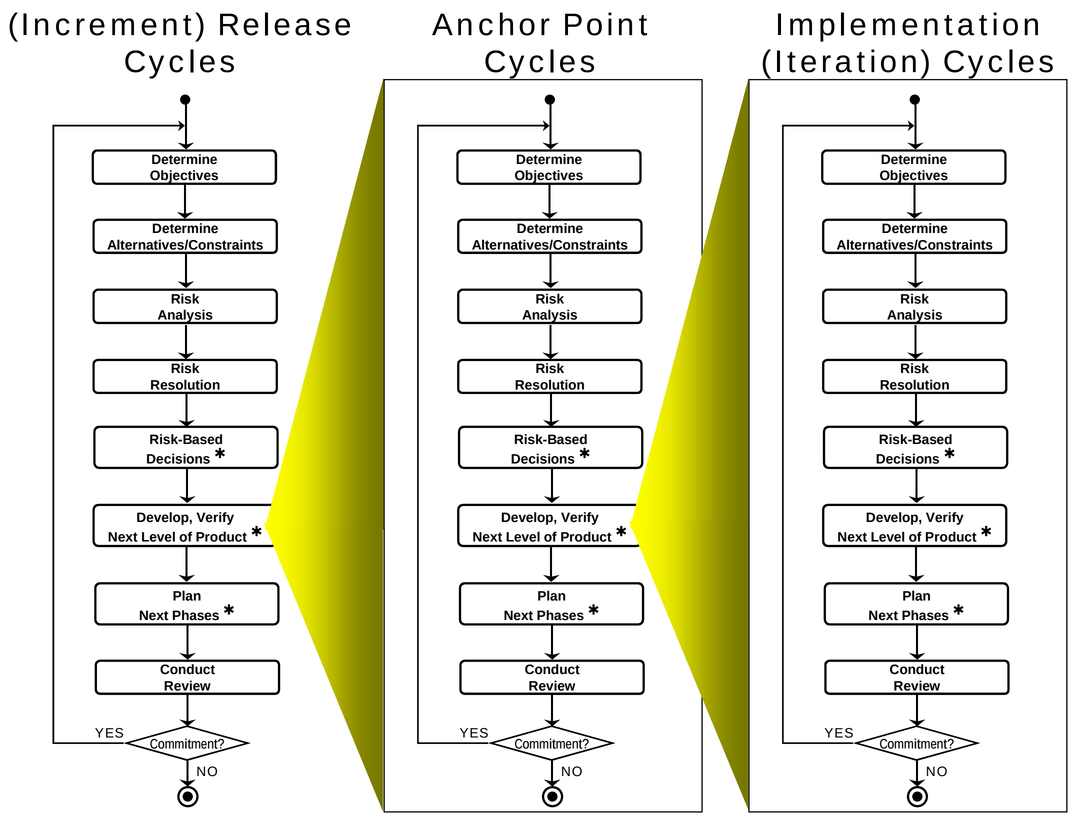
ULCM – The 10,000-Foot View

- **ULCM is a highly intuitive, pattern-based approach for specifying, constructing, visualizing and documenting the life cycle processes of software-intensive system development**
- **ULCM aspires to be the “Occam’s Razor” of Life Cycle Modeling**
  - ❖ The medieval rule of parsimony: “Plurality shouldn’t be assumed without necessity”
    - – William of Ockham, 14th century philosopher
  - ❖ The LCM rule of parsimony: All Life Cycle Models are constructs or derivatives of 4 basic LCM **patterns**
- **ULCM defines two views of life cycle models**
  - ❖ Generator View
    - – Algorithmic and sequencing aspects
  - ❖ Enactment View
    - – Temporal or trace dimension

The Spiral Model's ULCM Generator View Using UML Activity Diagrams



Boehm's Spiral Diagram*



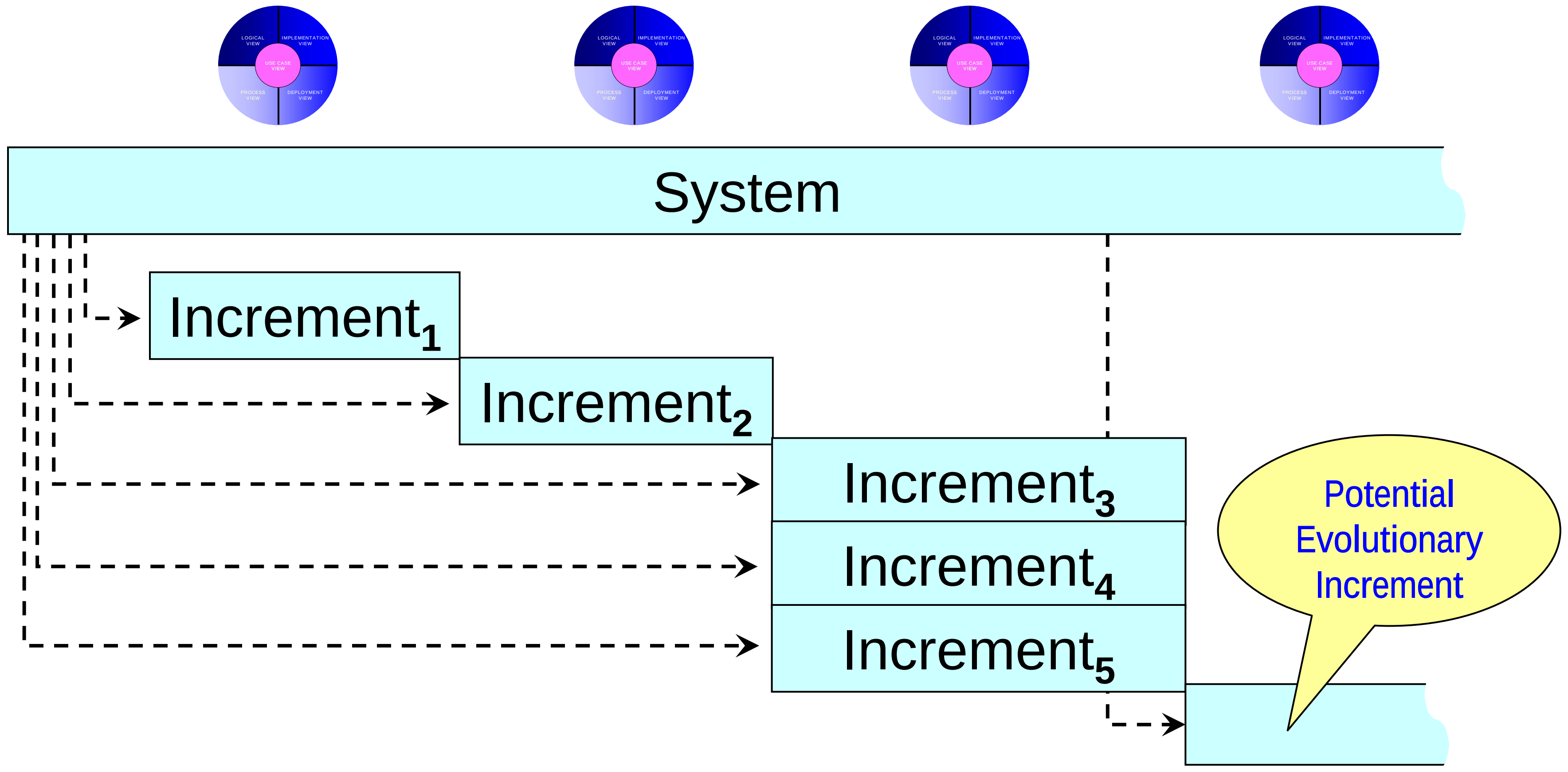
View shows activity sequencing and concurrency details of increments

Sources:

****   Spiral Diagram: (Boehm 1988)***

***** Nested Spirals: (Hantos 2006)***

ULCM Enactment View of Pre-planned Increments



View shows temporal dimension:
timing, duration, and synchronization of increments

Integrating the Views - A Little "View Algebra"

- **Why is Kruchten's Model called the 4+1 View Model?**
  - ❖ The first 4 views are independent
  - ❖ Use Cases of the 5th view cross-cut across the other views
    - – Initially used for discovery and design the architecture
    - – Later they can be used to validate the integrity of the views
- **How would ULCM Views fit into this structure?**
  - ❖ The Generator View would be a 5th, independent view
  - ❖ The Enactment View has an overarching function
    - – As such, it should belong to the **"+"** category, similarly to the Use Case View
- **And the result is:**

5+2 View Model

Conclusions

- **Architectural View Models are not static during the acquisition and development life cycle**
- **Life Cycle Models are key in ensuring the synergy across Architecture Evolution, Elaboration, and Evaluation**
- **In view modeling ensuring integrity across views is critical**
  - ❖ The solution is to use overarching, cross-cutting views
- **Using ULCM ensures the consistent level of modeling formality of architecture and life cycle models**
  - ❖ The use of these views could provide direct help to our SMC/NRO customers in implementing the guidelines of the current Software Development Standard for Space Systems (Adams 2005)

Acronyms

DBMS	Data Base Management System
IEEE	Institute of Electrical and Electronics Engineers
IR&D	Independent Research & Development
OMG	Object Management Group
TCP/IP	Transmission Control Protocol/Internet Protocol
ULCM	Unified Life Cycle Modeling
UML	Unified Modeling Language

References

Adams, R.J., et al, Software Development Standard for Space Systems, TOR-2004(3909)-3537, Revision B

Boehm, B. W., *A Spiral Model of Software Development and Enhancement*, IEEE Computer, May 1998

Hantos, P., Unified Life Cycle Modeling Tutorial, INCOSE 2005, Rochester, New York, July 2005

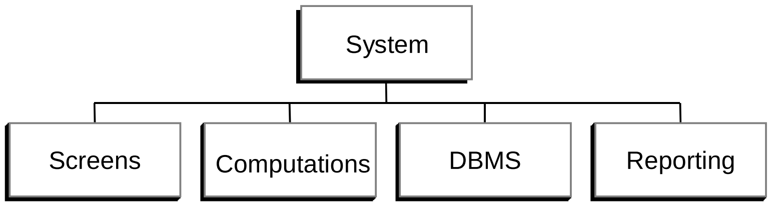
Hantos, P., *Interpreting the Spiral Model of Software-Intensive System Development – A ULCM^{SM} Approach*, CSER 2006, Los Angeles, California, April 2006 (To be published)

Kruchten, P. B., *The 4+1 View Model of Architecture*, IEEE Software, November 1995

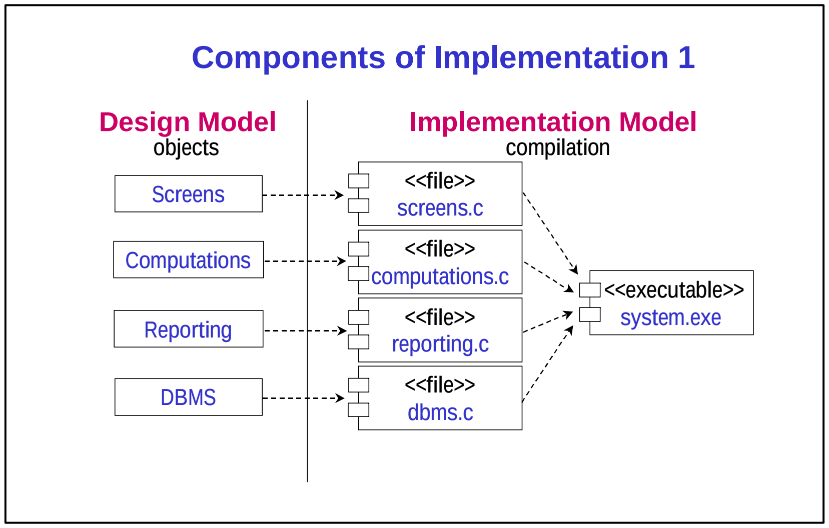
Kruchten, P.B., The Rational Unified Process An Introduction, Addison-Wesley, 1998

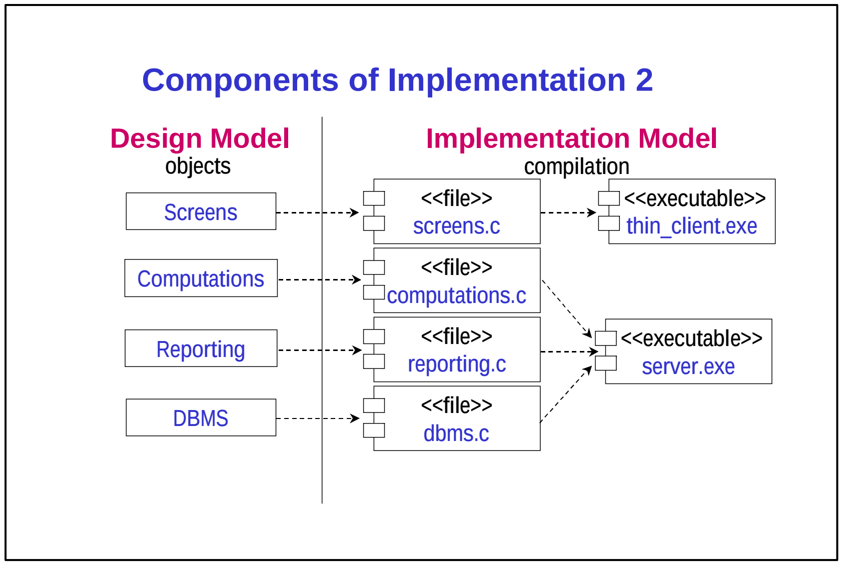
Backup Slides

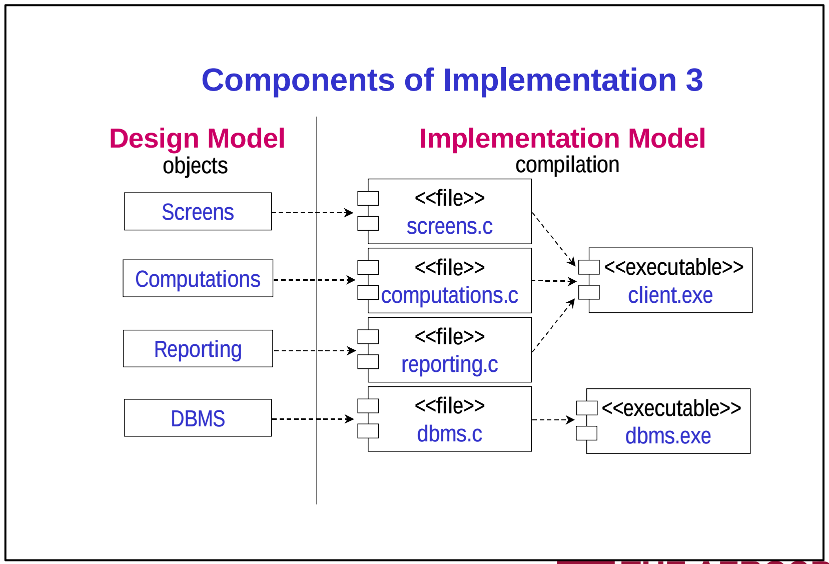
Logical and Implementation View Examples



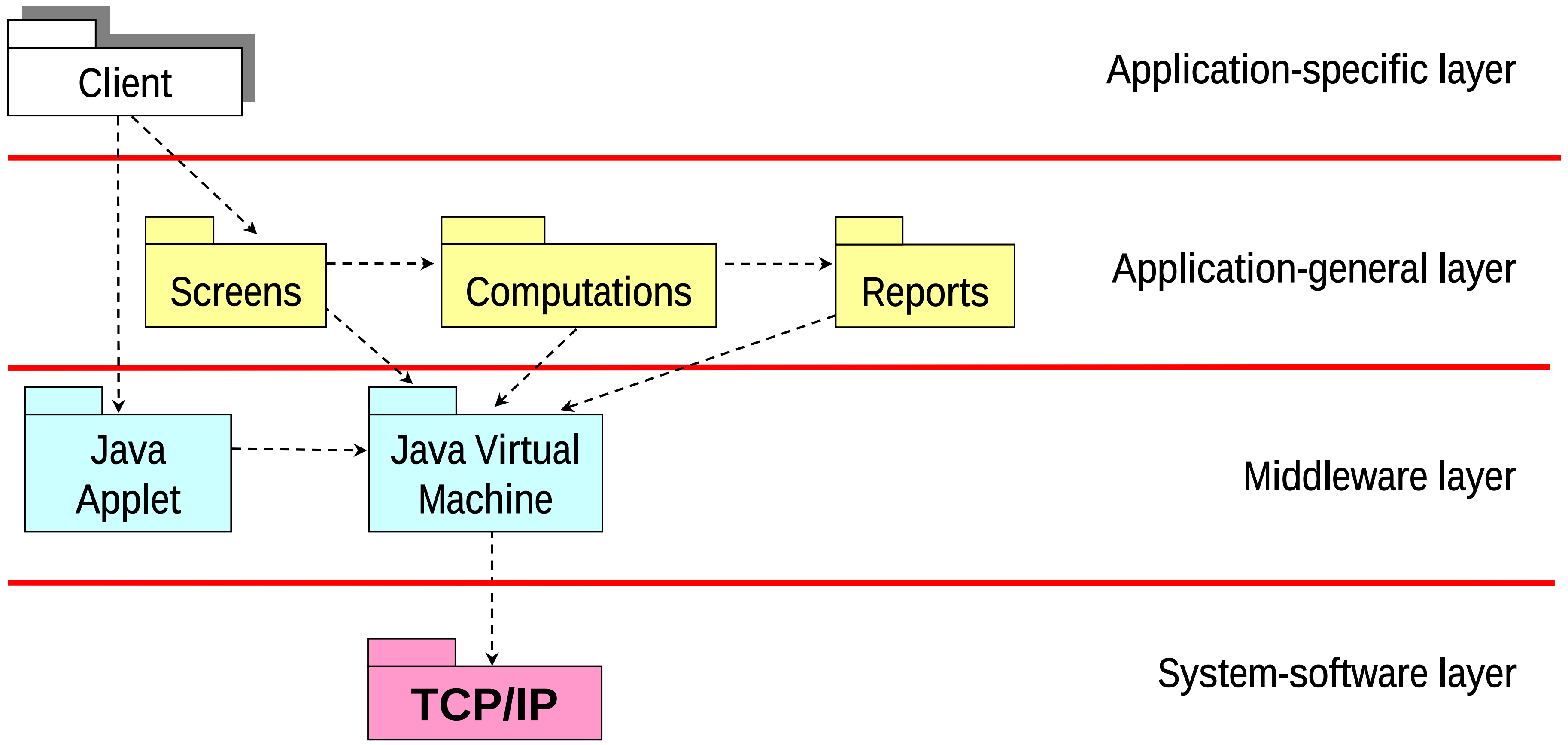
- **Implementation 1:**
  - **Mainframe with display terminal**
- **Implementation 2:**
  - **Client/Server – using "thin client"**
- **Implementation 3:**
  - **Client/Server – using dedicated database server**





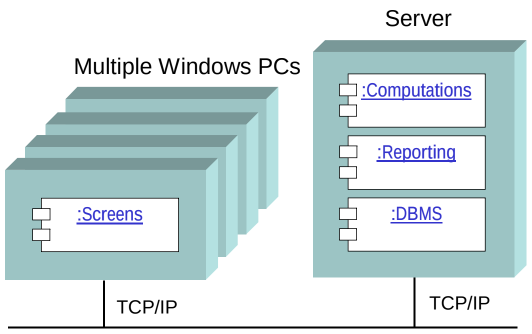


Packages and Layers Example for Implementation 3

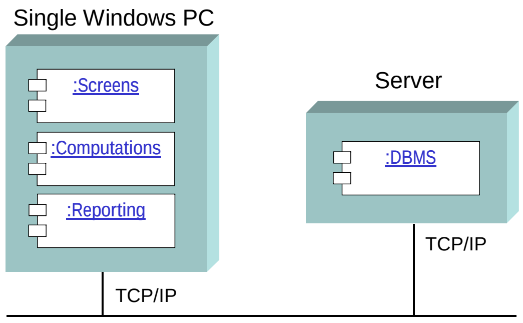


Deployment Views: Evaluating Deployment Options

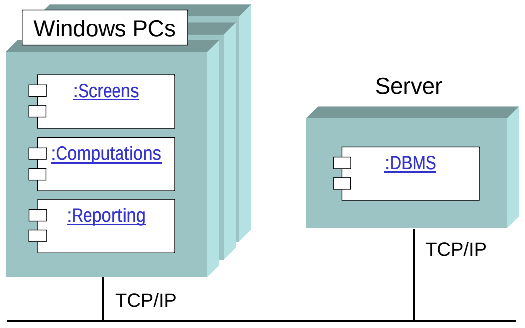
Deployment of Implementation 2 Using Multiple PCs



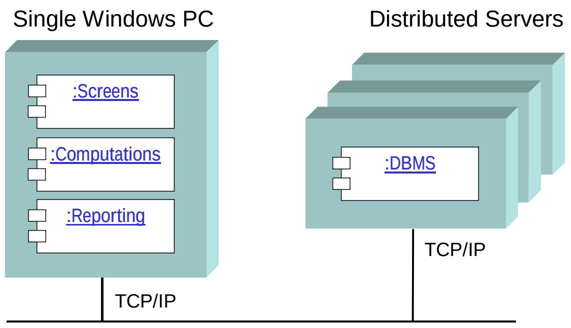
Deployment of Implementation 3 Using One PC



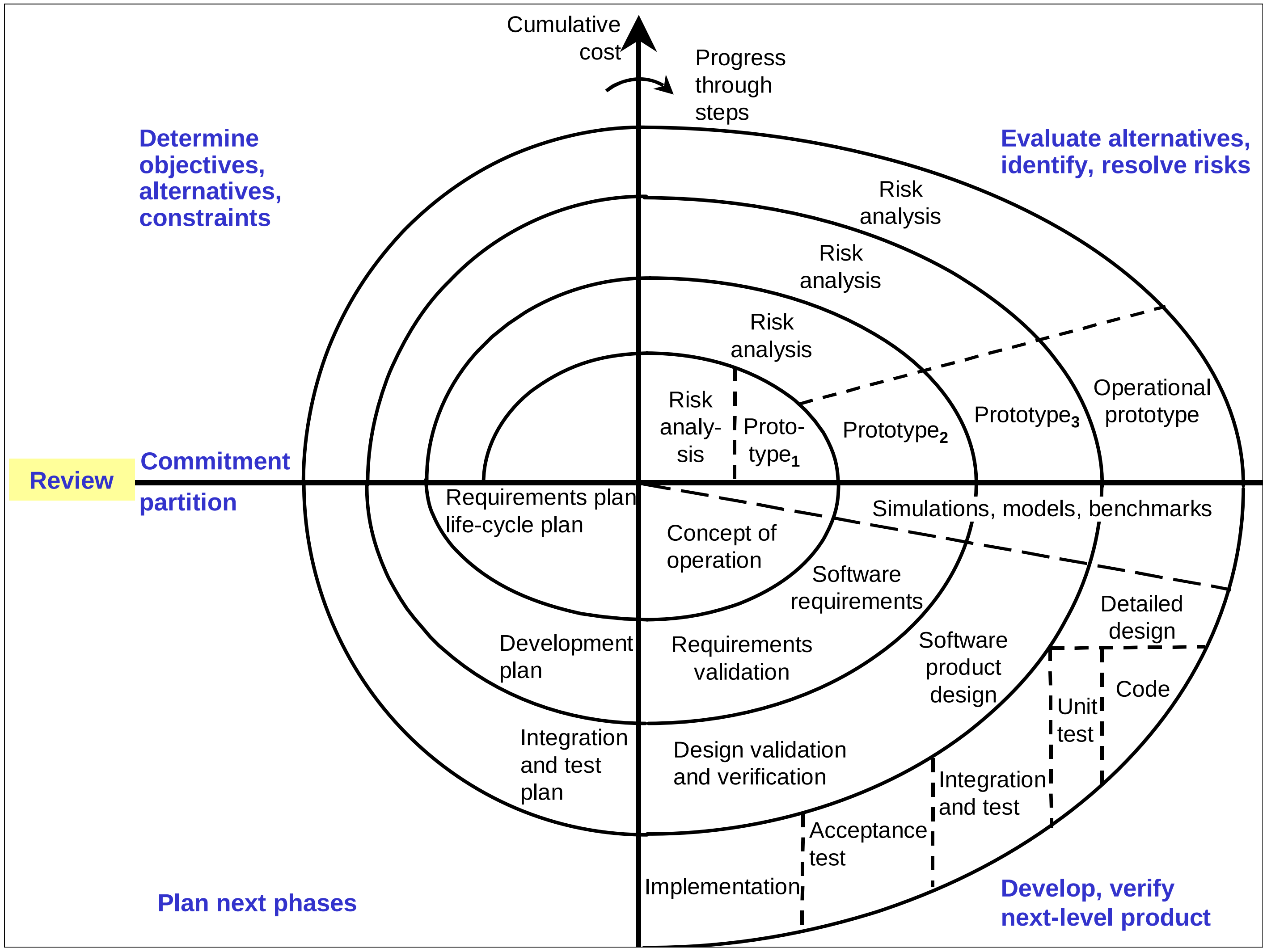
Deployment of Implementation 3 on Multiple PCs



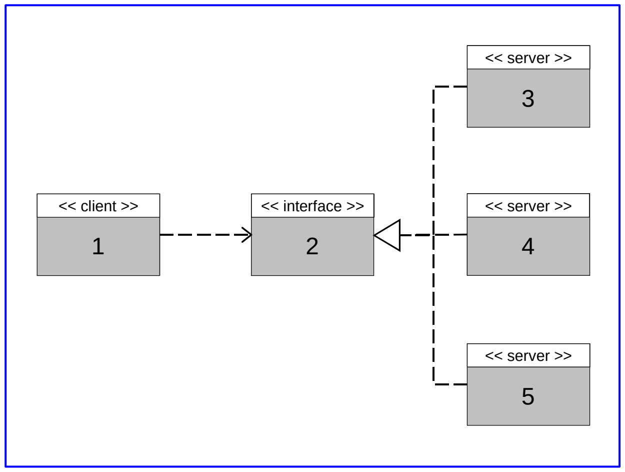
Deployment of Implementation 3 Using Distributed Servers

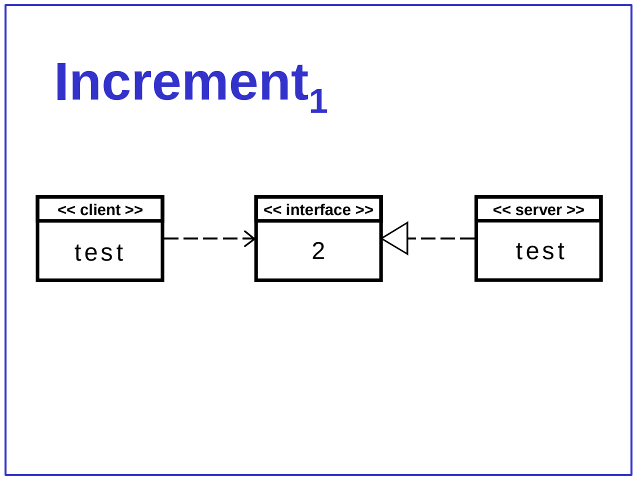


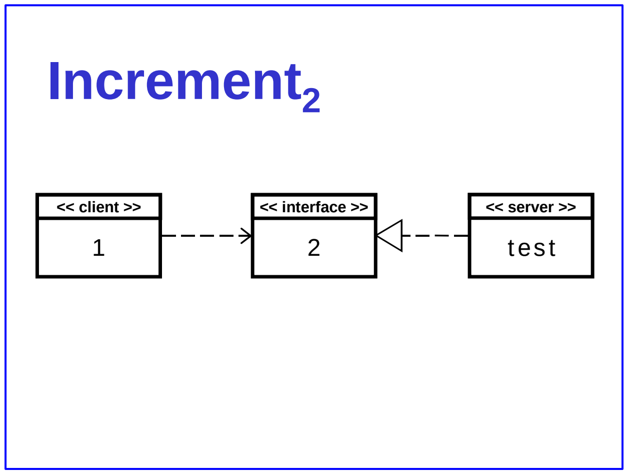
Boehm’s Spiral Model

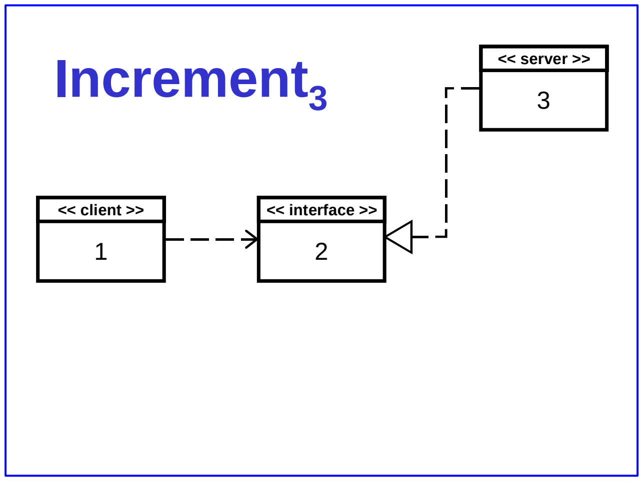


Increment Planning Example with Risk-based Considerations









Software Development Standard for Space Systems

Relevant references from the Standard:

5.6 Software design

"… if the system is developed in multiple builds, its design may not be fully defined until the final build. Software design in each build is interpreted to mean the design necessary to meet the software item requirements to be implemented in that build"

G.3 Scheduling deliverables

"To the maximum extent possible … leaving the door open for incremental delivery of software products, staggered development of software items…"

5.18.2 Joint management reviews

"The developer shall plan and take part in joint management reviews…"

Appendix E: Candidate joint management reviews

- E.3.4b "The architectural design of the system/segment/…"
- E.3.6b "The architectural design of a software item"