

Client/Server Architecture Team Responsibilities

Session 110

May 13, 1996

Michael Schmutz

Deloitte & Touche Consulting
Group

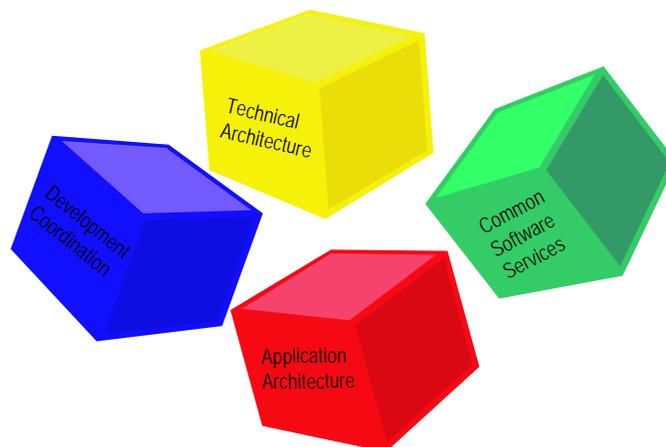
Kate Calarco Kuhn

Kaiser Permanente

Deloitte & Touche Consulting
Group

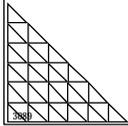


Agenda



Responsibilities of a Client/Server Architecture Team

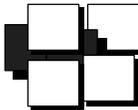
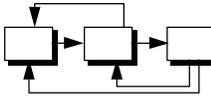
- Provide development coordination and technology integration leadership
- Develop Information, Application, and Technical Architecture
- Define and select the right set of tools
- Ensure infrastructure necessary for all development sites, platforms, and lifecycles
 - ❖ Definition and administration of standards
 - ❖ Creation and support of development environment
- Model management and change control
- Common Software Services
- Maximize reusability of system development components



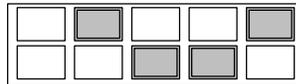
What is ARCHITECTURE?

All Companies have a systems architecture, either by design or by default.

Architecture can evolve on a system by system basis

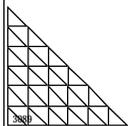
- When components are added onto a core 
- Or, by linking components together after they have been created individually 

OR,
Architecture can be designed at a master level and individual components created as they are needed or funded



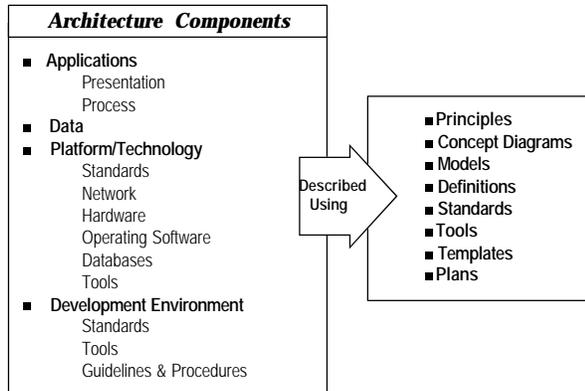
“Developing systems without an overall architecture is like doing a jigsaw puzzle in your lap.”

—Ian Miller, Director of Information Architecture, Merck Pharmaceuticals



What is ARCHITECTURE?

Client/Server Architecture is a framework within which each of the major aspects of systems capability is described separately and evolved somewhat independently. Yet together, they create an integrated whole.



5

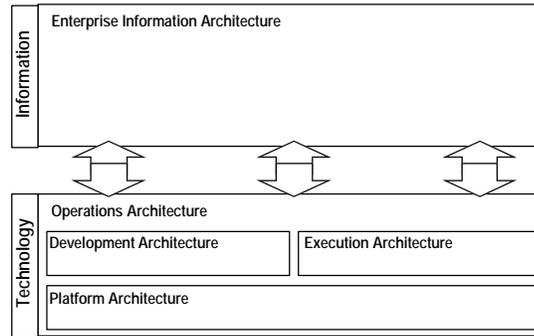
Why Do We Need an Architecture?

- An Application and Technology Architecture is the foundation upon which systems are developed
- A well designed Application & Technology Architecture ensures:
 - ❖ Flexibility
 - ❖ Scalability
 - ❖ Interoperability
 - ❖ Reusability
 - ❖ Reliability and stability
- Without a well defined and cohesive architecture business systems will grow ad hoc to the point at which a crisis will occur requiring a complete reevaluation of all systems.

6

A Framework for Application & Technology Architecture

- **An Enterprise Information Architecture** describes the overall structure of an organizations' information and business processes, and how they are supported by the computing environment.
- A **Technology Architecture** describes the computing environment, its component functionality and component interrelationships.

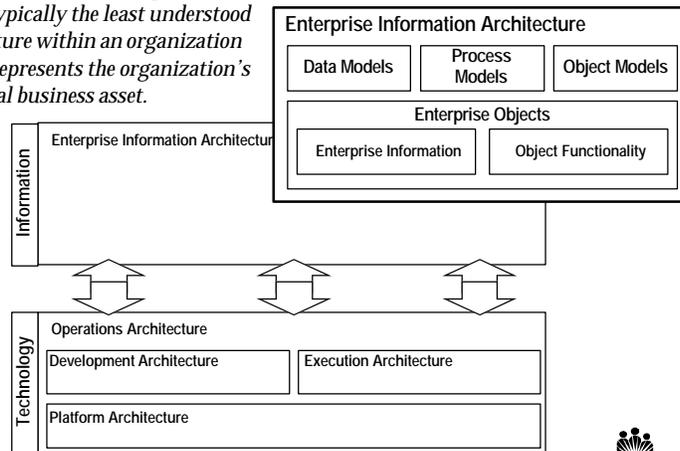


7



Enterprise Information Architecture

*The **Enterprise Information Architecture** describes the overall structure of an organization's information and business processes. This is typically the least understood architecture within an organization but yet represents the organization's most vital business asset.*

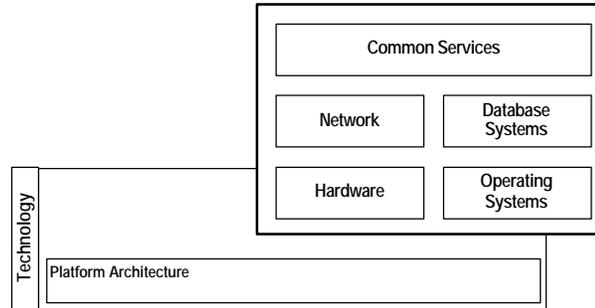


8

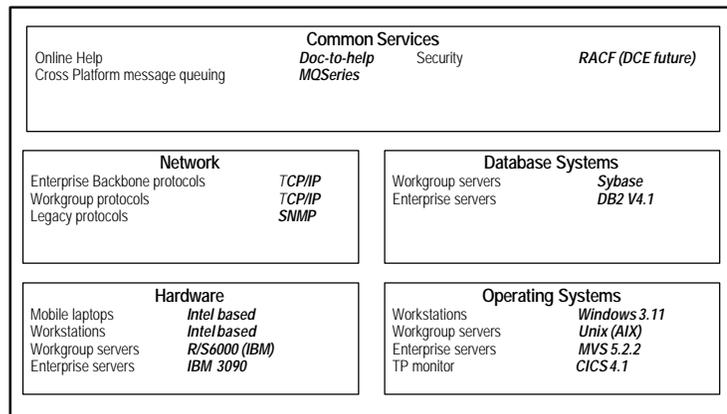


Platform Architecture

The **Platform Architecture** describes the basic hardware, operating systems, networks, database systems and common services upon which an enterprise's information systems are delivered and executed.

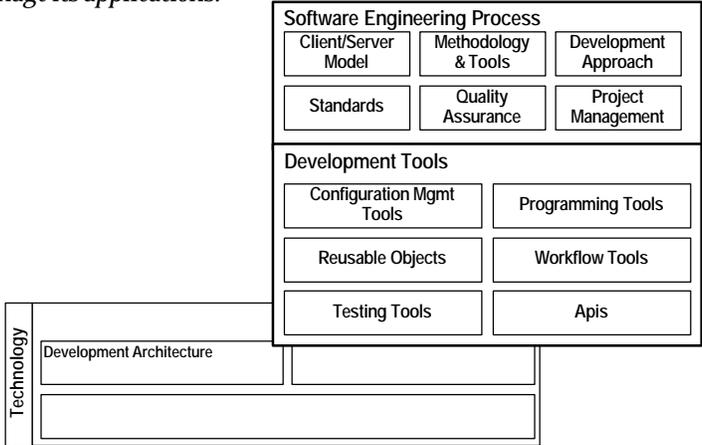


Platform Architecture

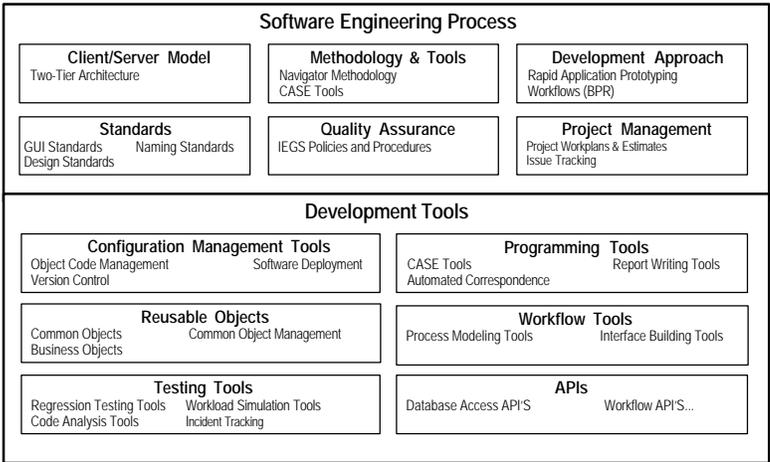


Development Architecture

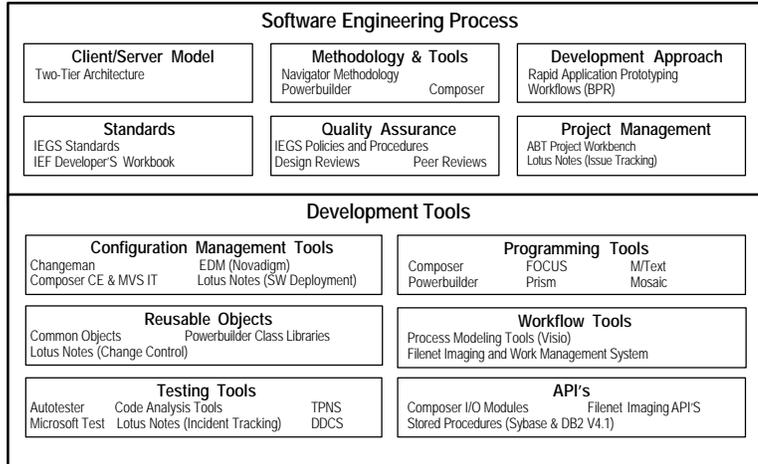
The **Development Architecture** describes the methods, process, techniques and tools that an enterprise utilizes to develop and manage its applications.



Development Architecture

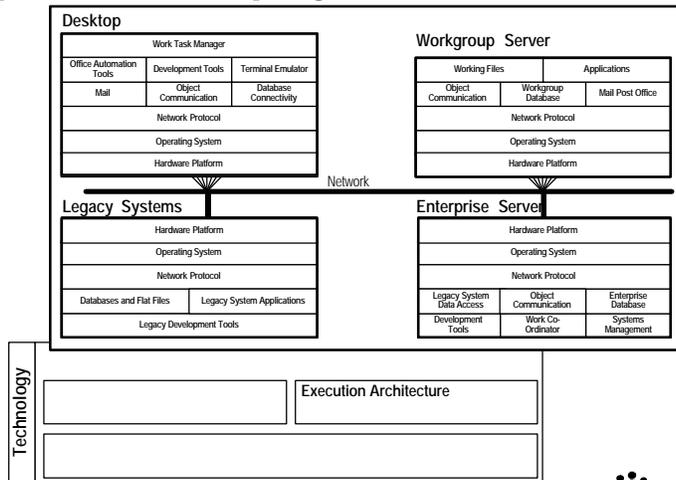


Development Architecture

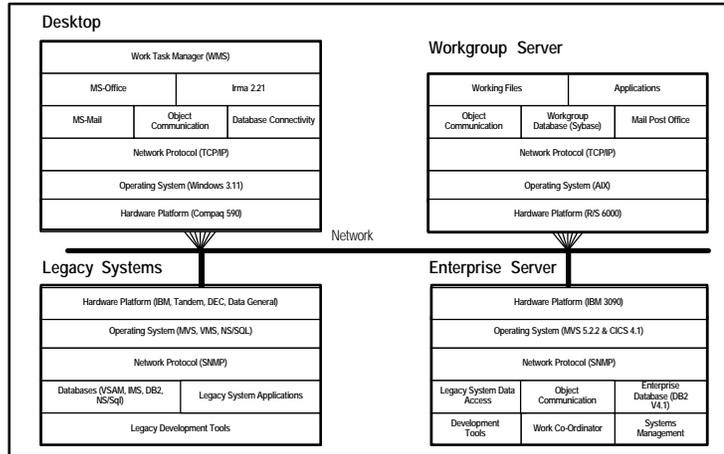


Execution Architecture

The Execution Architecture describes the physical characteristics of the implemented end user computing environment.

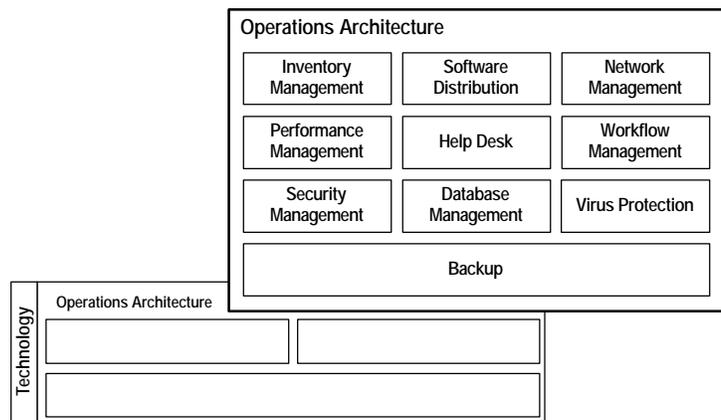


Execution Architecture



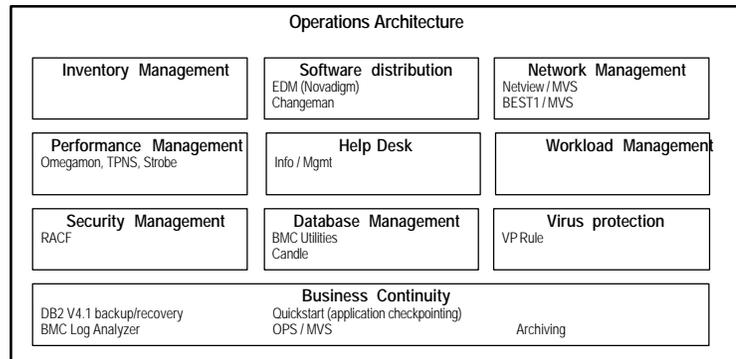
Operations Architecture

The Operations Architecture describes the methods, process, tools and techniques used to support, protect and manage the technology environment.



Operations Architecture

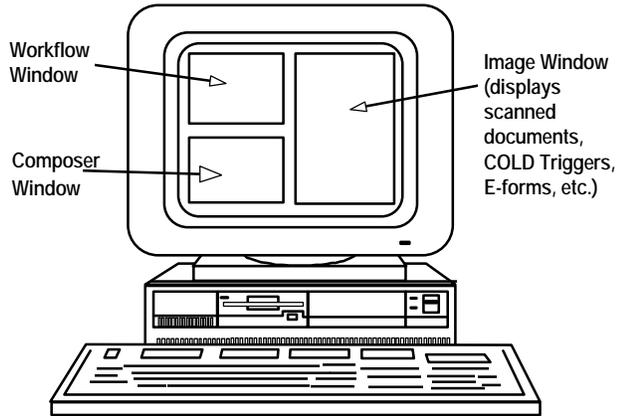
A suite of products that perform systems management in an IBM mainframe environment.



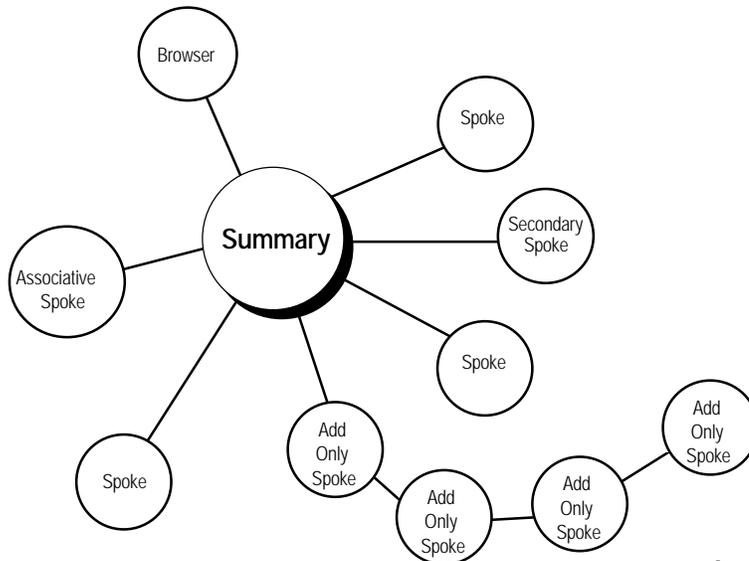
Implementing an Application Architecture

- Development Environment
- Workstation Configuration
- Selecting a User Metaphor
- System Navigation
- GUI Standards and Behavior (IEF Developer's Workbook)
- Application Templates (Client/Server and batch)
- Common Objects
- Common Software Services (Model Management and Change Control)

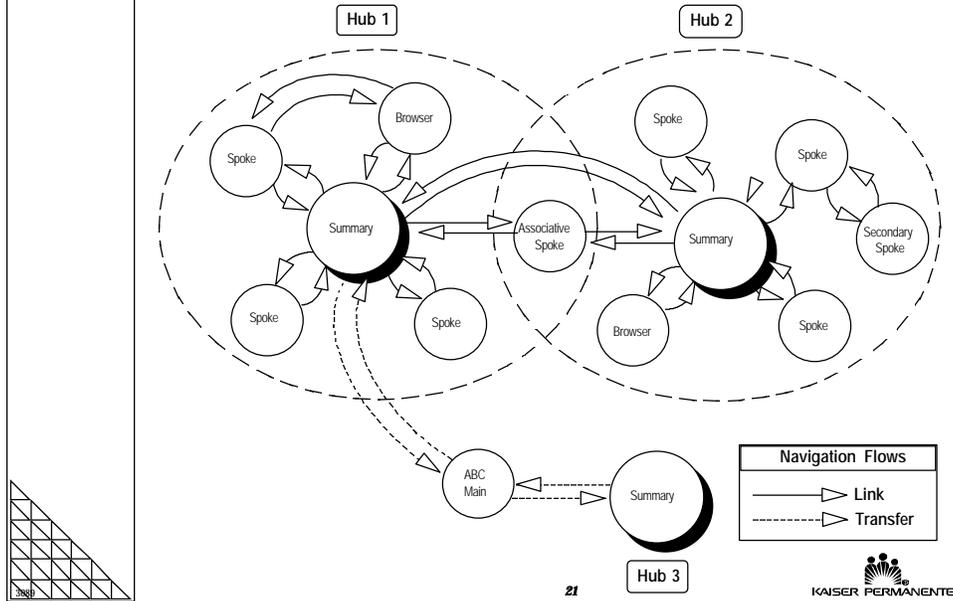
Workstation Configuration



Hub and Spoke User Metaphor



System Navigation

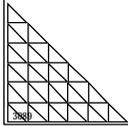


Common Objects

- Application Templates (Client/Server and Batch)
 - ❖ Template for each GUI window type
- Clonable Routines
 - ❖ Code table, address/phone maintenance, text management
- Common Routines (Date Conversions, ID generation, SSN formatting)
- Security Authentication/Authorization (RACF EAB)
- System interface CABs (MQSeries, WMS Triggers)

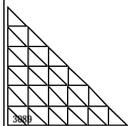
Model Management & Change Control

- Development and Test Models
- Change Control Process
- Software Deployment



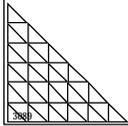
Development Models

- Architecture
- Corporate Data/Technical Design
- Project Business System Implementation
- Common Objects
- Common Business Action Block
- Object Management



Architecture Model

- Logical view of all data and activities
- Minimal migrations allowed into or out of
 - ❖ Migrations out would cause logical objects to overlay physical objects (lose denormalization or other performance tuning changes)

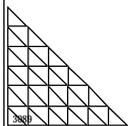


25



Corporate Data/TD Model

- Central point for making and distributing data/TD changes
- DEV: model where data model changes are made
- APRV: model containing approved set of changes ready for distribution

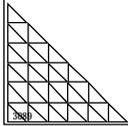


26



Project BSI Model

- Used for developing most procedures, windows, and action blocks
- Phased development of system business processes
- Dedicated development/rework team



27



Common Objects Model

Common Objects

- Part of the production application
- Examples: work sets, exit states, design action blocks

Templates

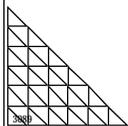
- NOT part of the production application
- Examples: procedures, exit states, cloneable objects (address, phone, comments)

Support Applications

- Generic code table maintenance windows
- Work management and M/Text letter printing batch procedures

DEV: model where common objects are developed

APRV: model containing approved common objects ready for distribution

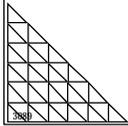


28



Common Business Action Blocks Model

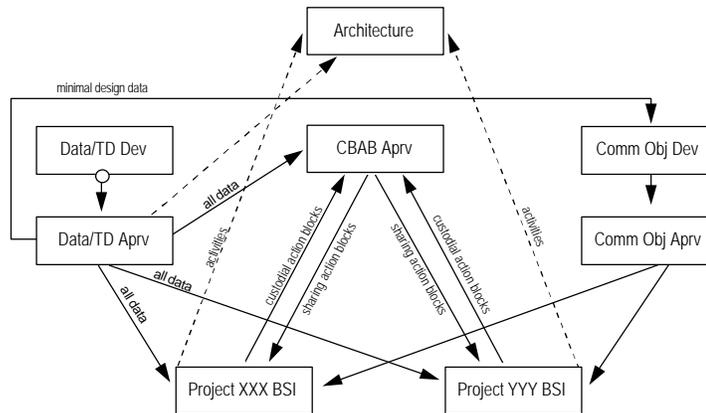
- CBAB is an elementary process action diagram or action block that references business entity types
- Central point for distribution of shared action blocks that reference business data
- CBABs are built and maintained in Project BSI models
- APRV: model containing approved set of CBABs ready for distribution



29



Object Management



Key:
 migrate →
 copy ⇨
 rename ○→
 re-click - ->

30



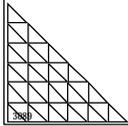
Test Model

Will contain the sum of all the Project BSI Models

- Objects will be migrated in only after they have been tested in Development Release Models

No direct changes allowed

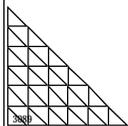
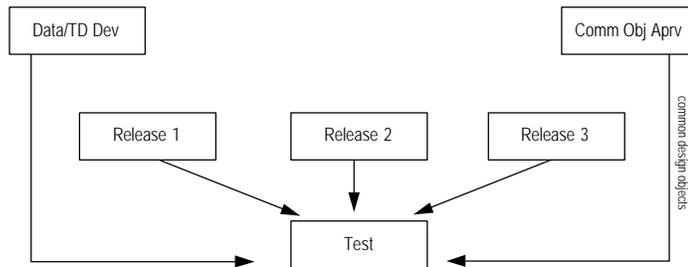
- Developers cannot subset & change
- Model Management can migrate objects in
- Model Administrator: SW Deployment



31



Object Management



Key:
migrate →
copy →
rename ○→
re-click - ->

32

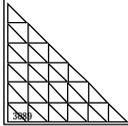


Change Control Process

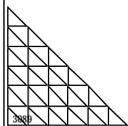
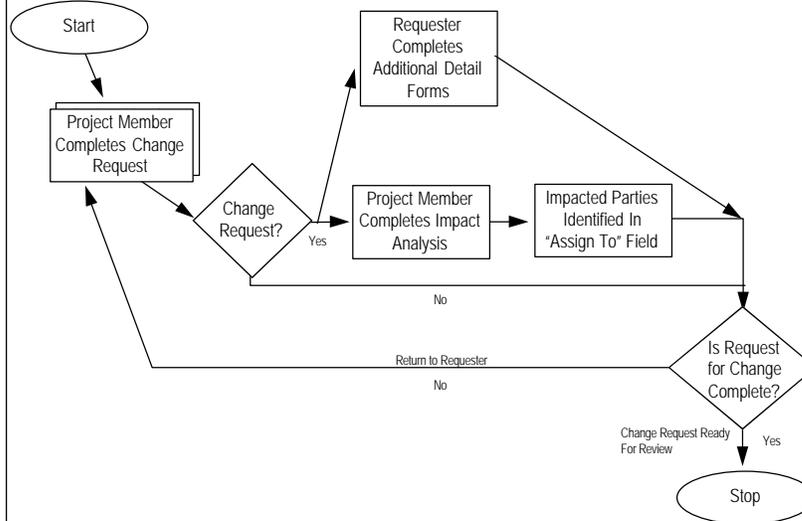
Change control allows orderly modification to shared objects. The general steps followed include:

- Request a change to be made
- Perform impact analysis to identify affected parties
- Have the affected parties review and approve the change request
- Raise the request to project management for a ruling, if consensus cannot be achieved
- Make the approved change and complete the change request

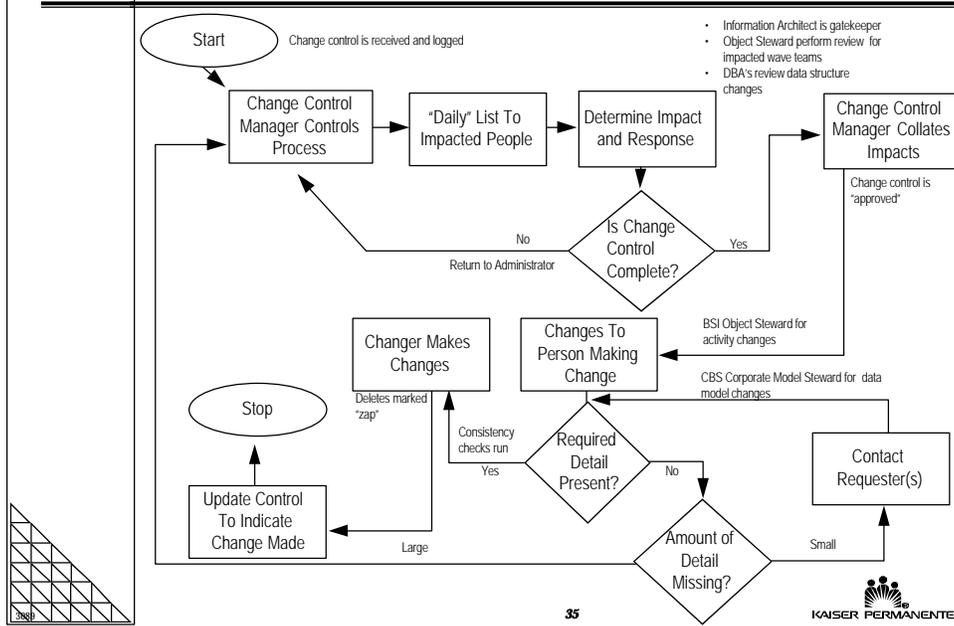
Apply the approved changes to a version of project models



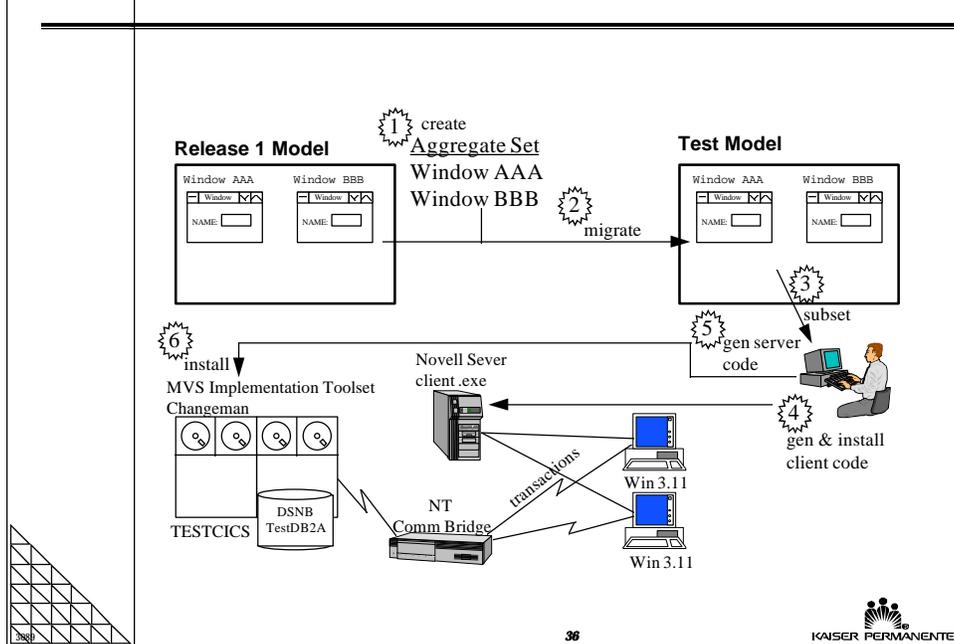
Request for Change Creation Cycle Request for Change Creation Cycle



Process for Change Creation Cycle

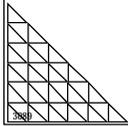


Software Deployment Overview



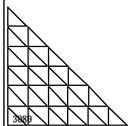
Role of a Pathfinder Project

- Confirmation of Technical Architecture
- Validation and refinement of Application Architecture
- Production of deliverable templates
- Workplan actuals vs. estimates
- Vehicle for client/server benchmarking
 - ❖ Performance testing
 - ❖ Software deployment
 - ❖ Configuration management
 - ❖ Testing methodology and tools



Development Coordination Using Lotus Notes

- Knowledge coordination using workgroup knowledge base
- Discussion databases
- CBS Library
- Deliverables database
- Issue tracking
- Status reporting
- E-mail



Questions & Answers

