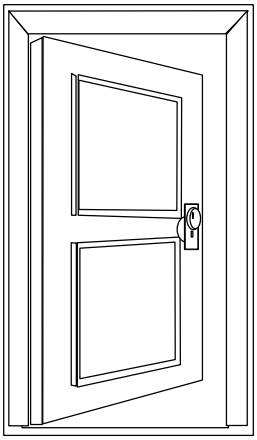
The Role of the Composer DBA

Session 720

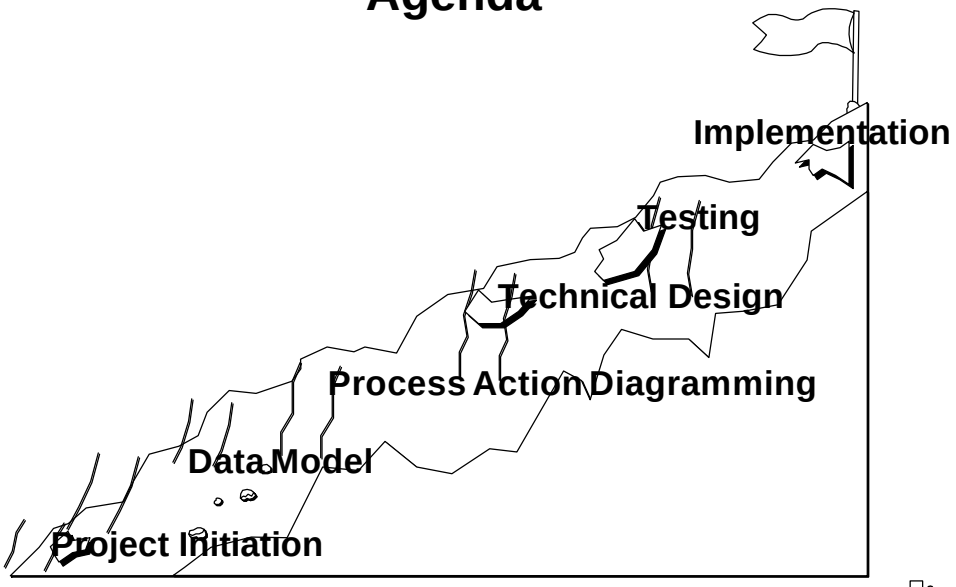
Mark D. Holmes
Texas Instruments

Introduction

- Provide a full life cycle perspective on how a Composer DBA can positively influence the success of a project
- Assumed knowledge base is an understanding of Composer analysis and technical design concepts

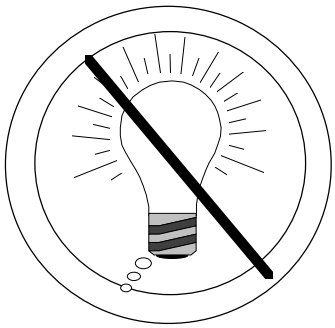


Agenda



The Big Mistake

- DBAs not included Early
- Reactive DBA
- DBAs not trained on the tool

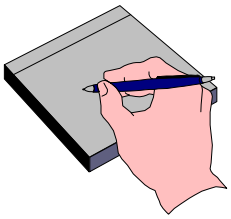


Composer DBA Roles

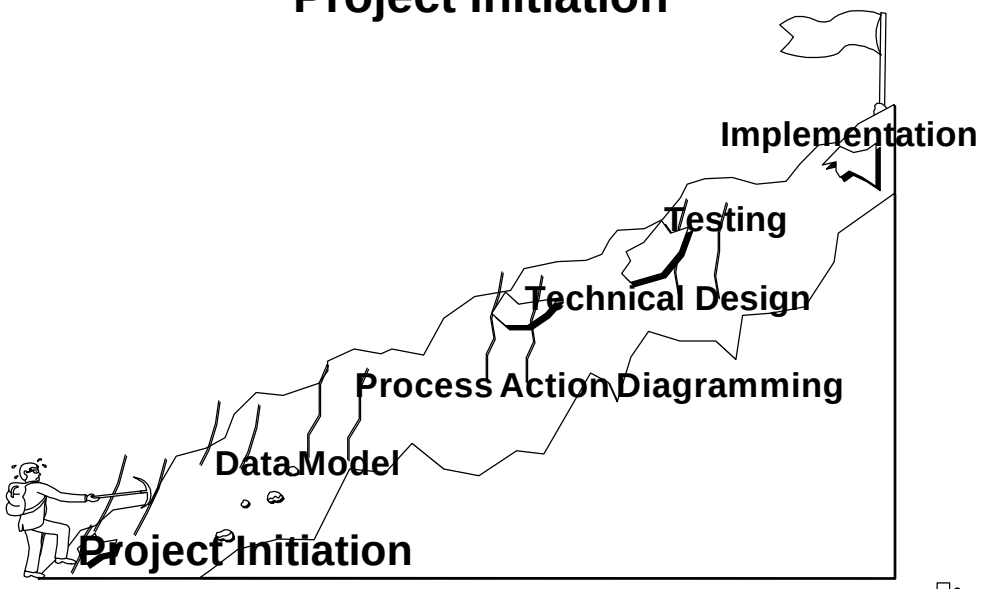
- Project team member
  - May support multiple projects simultaneously
- Full life cycle participant
- Enforce good modeling practices
- Maintain the physical data model and data structure
- Be responsible for the application implementation

Composer DBA Skills

- Be trained in advanced modeling techniques
- Have a working knowledge of how to create Action Diagrams
- Have expertise in the technical architecture



Project Initiation

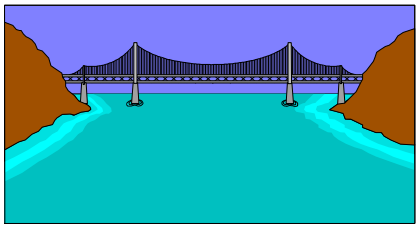


Project Initiation Activities

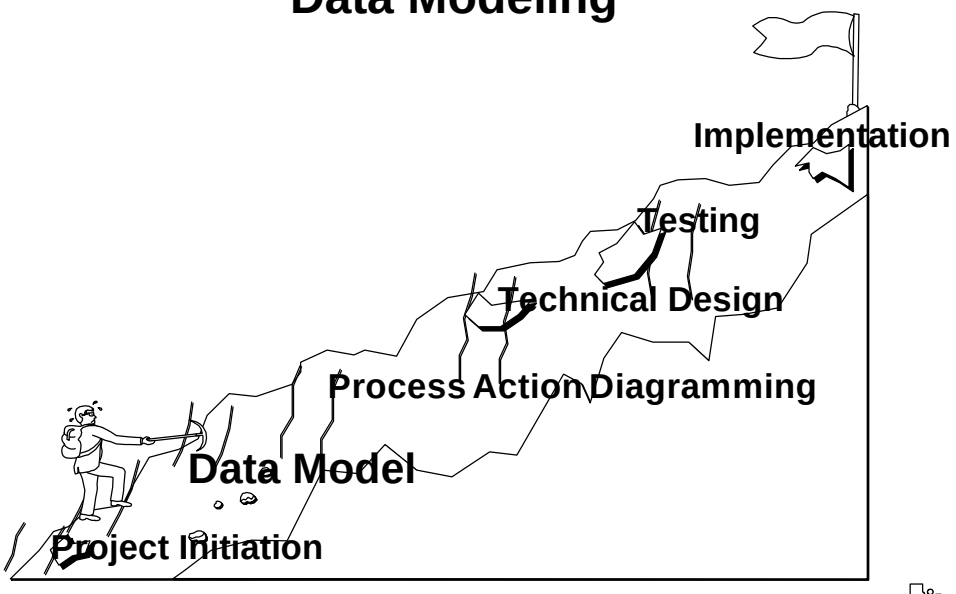
- Define standards
- Set up a shared public interface access
- Identify performance requirements
- Identify database requirements for backup, recovery, maintenance, and utility activities

Bridging and Conversion

- Model interface
- Consider Composer load applications
  - Avoid data in consistencies (permitted values)
  - Default values
  - Document conversion rules
  - Adhere to model business rules



Data Modeling



Review Point

- At least one review should be held at the conclusion of data modeling
- Performance, implementation and stability issues should be considered

Things that Don't Transfer Across Environments

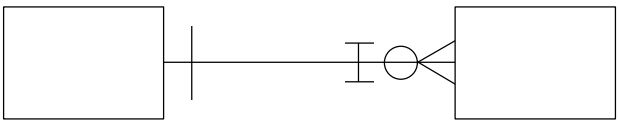
- DBMS differences in column length, number of columns, number of tables, number of tables in a join, reserve words
- Differing implementations of VarChar
- Timestamp (Oracle)
- Precision available in COBOL is not available in C

Entity Type Definition

- Code/Lookup tables identified
- Data Structure Diagram names for attributes and entity types become field and record names
- Review implications of external loads
  - Default values
  - Permitted values
- Volumetric data should be included during analysis
  - Large tables identified
  - Total disk space needed identified
  - Should be as accurate as possible

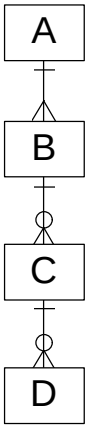
Relationship Definition

- All relationships should be one-to-many, optional and identifying, or the many side
- Foreign key chaining identified
  - Surrogate identifiers selected



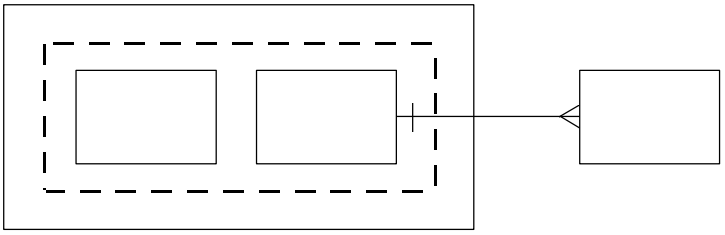
Relationship Definition

- Select the delete option on each relationship membership
  - Cascade Delete
  - Cyclical Cascade Delete
  - Set Nulls
  - Pendant Delete
  - Restricted Delete
- Specify referencing on the many side of relationship membership to improve concurrence



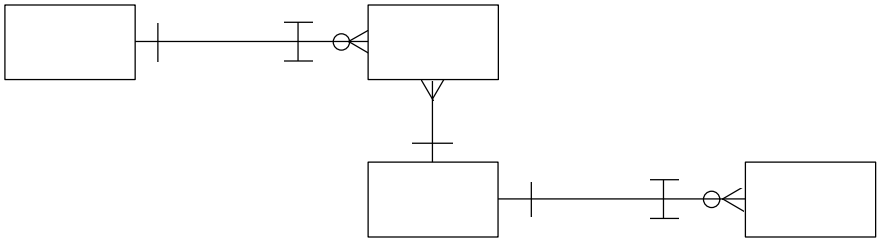
Other Considerations

- Add redundant relationships for performance
- Consider derived attributes in place of redundant relationships
- Consider promoting subtypes

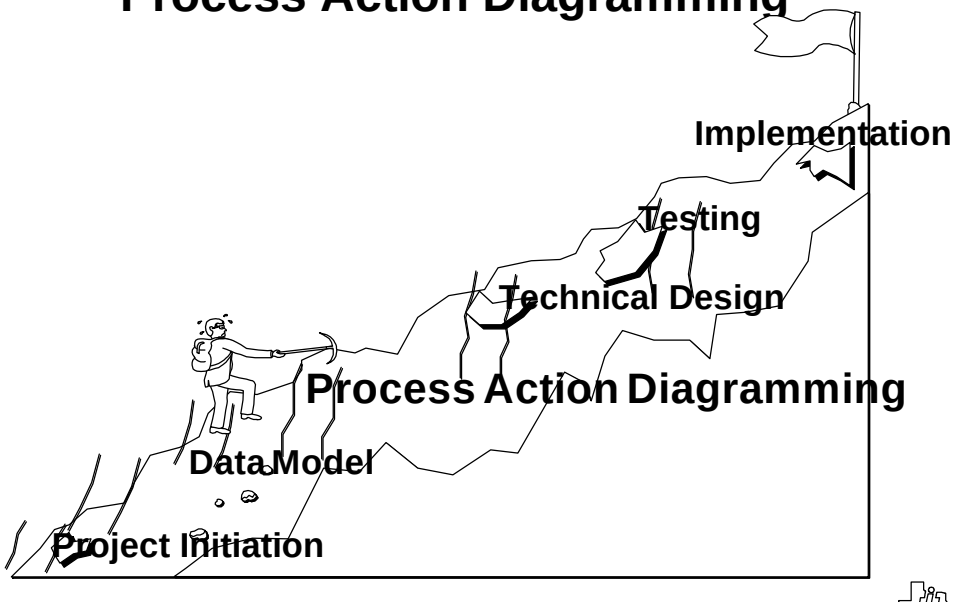


Stabilize...Stabilize...Stabilize

- Stabilize your data model before performing action diagramming



Process Action Diagramming



Review Point

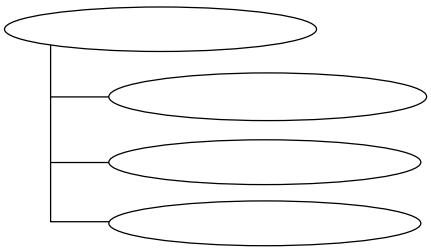
- At least one review should be held at the conclusion of process action diagramming
- Performance, implementation, and stability issues should be addressed

Examine READ Statements in Process Action Diagrams

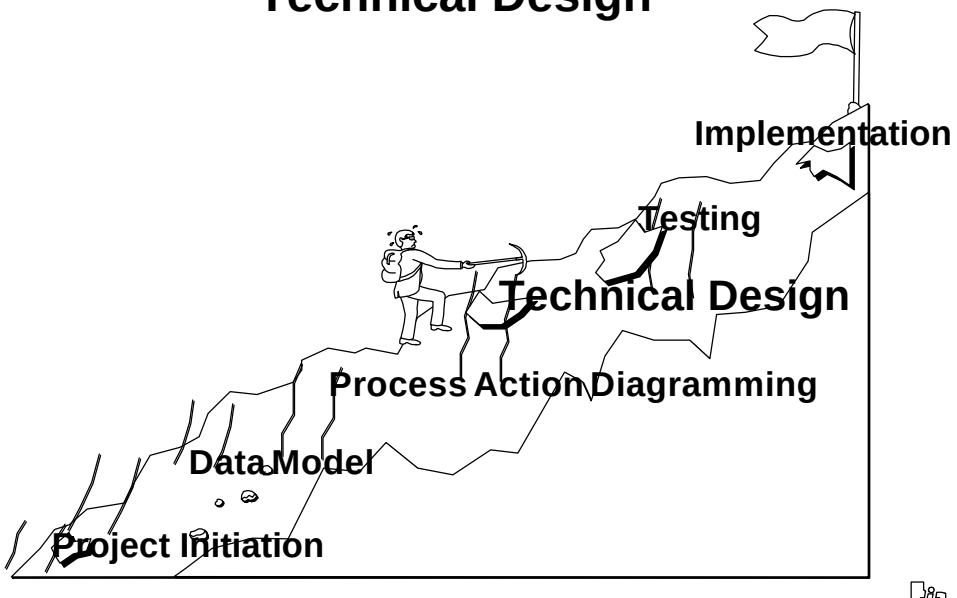
- Use extended READs in action diagrams to allow optimization in generated code to access denormalized fields rather than performing tables joins
- Number of joins identified
- Note READs not qualified by identifiers

Process Review

- Starve entity action views
- High-volume elementary processes identified
  - Concentrate on critical path processes



Technical Design



Review Points

- After completing Procedure Action Diagramming
  - Examine READ statements
- Prior to performing Transformation
- After performing Transformation
- Performance, implementation, and stability issues should be addressed

Pre-Transformation Review

- Identifiers
  - Names become index names
  - Avoid multiple identifiers
- Select *"all"* DBMS reserve word defaults

Pre-Transformation Review

- Composer vs. DBMS-generated Referential Integrity
  - Select DBMS-enforced Referential Integrity whenever possible to decrease trigger module size and increase performance
- Review items whose implementations are different in the unit test environment vs. the target environment

Post-Transformation Review

- Change nulls to not nulls for foreign keys of mandatory relationships
- Delete extra entry points on foreign keys
- Review foreign keys names

Post-Transformation Review

- Add entry points on non-identifier fields that are used to qualify READs in high volume processes or procedures
- Normalize those fields that are referenced frequently in high-volume processes or procedures
- Enforce DBMS naming standards within the tool

Transformation Options

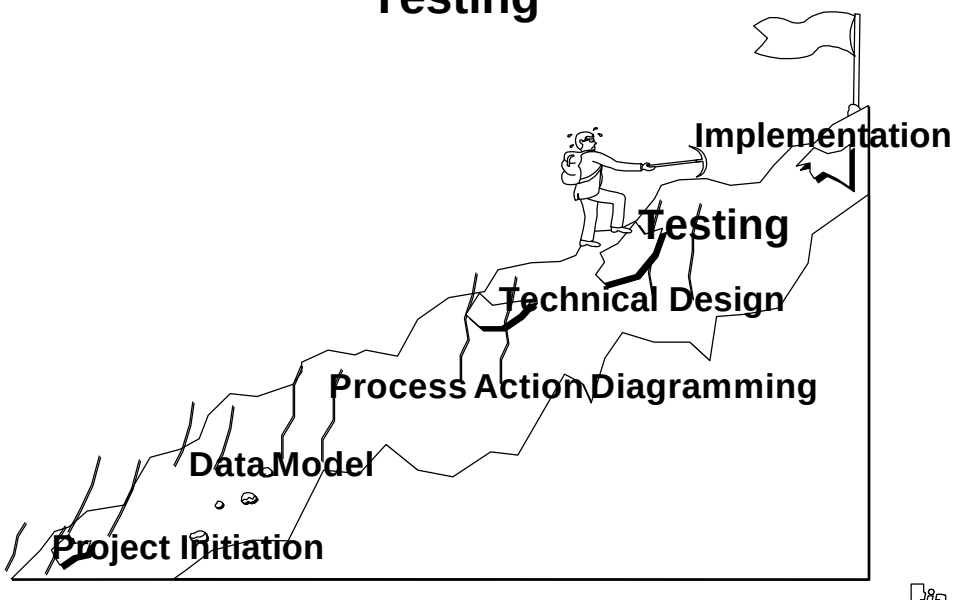
- Initial Transformation creates Data Structure Diagram records
- Intelligent Retransformation used when changes are made to the data model
- “Implement Entity Type” option can be used to implement changes to specific entity types

Coordination Days

- Transformation/ Retransformation performed
- Objects tagged for deletion removed
- Batch database changed and performed on a routine schedule

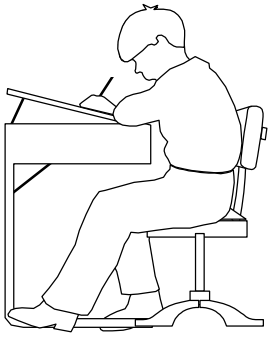
1999						
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Testing

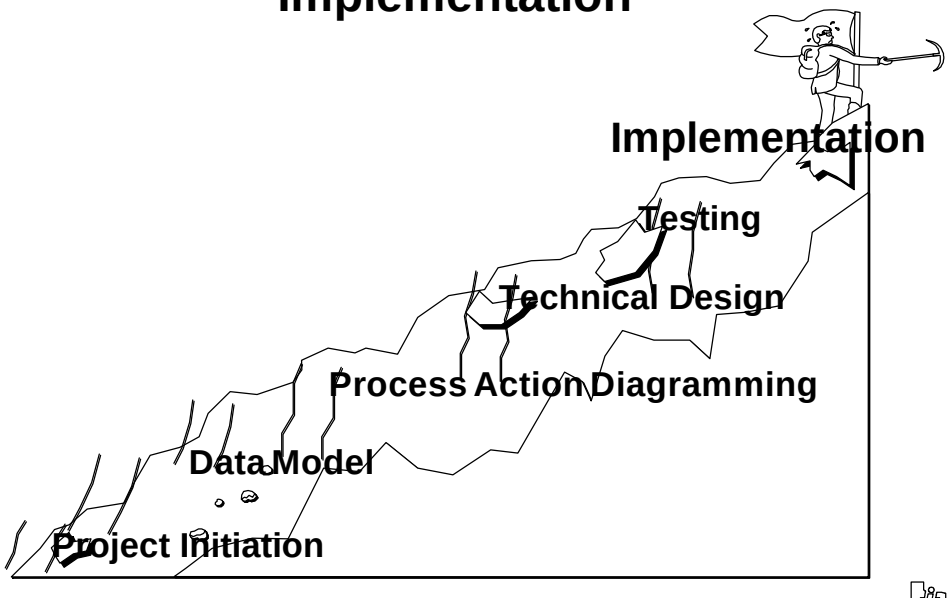


System Test Should Include

- Production volume of data
- Performance testing
- Backup and recovery
- Utilities and maintenance routines

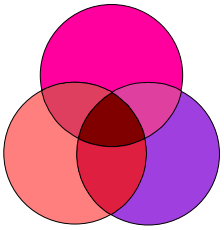


Implementation



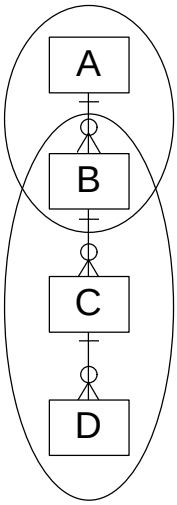
Shared Objects

- Objects that exist in more than one place should be the same
- Migrate objects that are the same versus creating them in the new source
- Must be tested in an integrated test environment
- Manage local databases and test data as shared



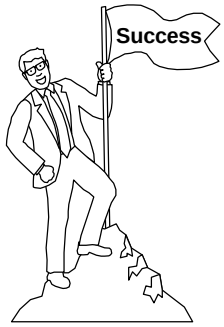
Shared Objects in Production

- Projects released as multiple models
  - All entity types in both places
  - Entity types included that cascade
  - Don’t implement new relationships
- Project combined into one model
- Shared object model



Summary

- DBAs should be well-trained on the tool
- Maintain the database within Composer
- Composer DBA should be full life cycle participant



The Role of the Composer DBA

Session 720

Mark D. Holmes
Texas Instruments