Model-Based System Architecting and Software Engineering (MBASE)

CS 577a, 510

Barry Boehm, USC

2000 Version

University of Southern California
Center for Software Engineering

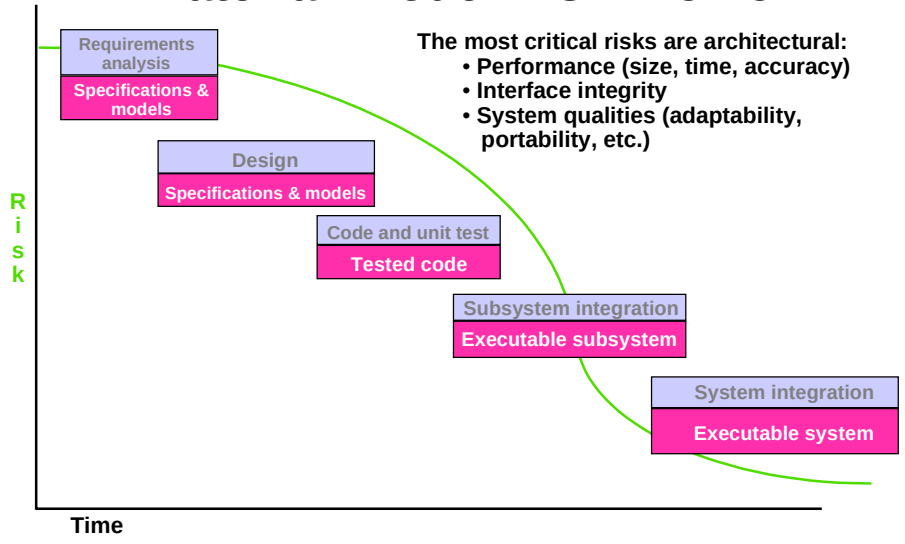
Outline

- A short history of software processes
  - Waterfall, spiral, MBASE/Rational/Benefits Realization
- MBASE Guidelines and Electronic Process Guide
- Application to digital library projects, CCPDS-R

A Short History of Software Processes - necessarily oversimplified

Decade	Orientation	Example(s)	Problems
1950's	Engineering	SAGE	Engineering orientation
1960's	Programming	Code and fix	Rework, scalability
1970's	Requirements	Waterfall	Risk, GUI, COTS/reuse
1980's	Many	Evolutionary, Incremental, Spiral, Helix, JAD, RAD	Overgeneralization/ Overspecialization
1990's	Emergence	Win-Win Spiral, Rational, Adaptive (model generators)	Value and economics
2000's	Value	Benefits realization, MBASE	Integration with emergence

Waterfall Model Risk Profile

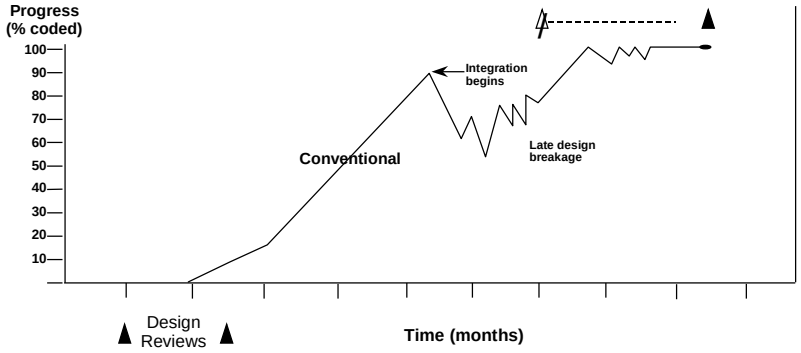


The most critical risks are architectural:
- Performance (size, time, accuracy)
- Interface integrity
- System qualities (adaptability, portability, etc.)

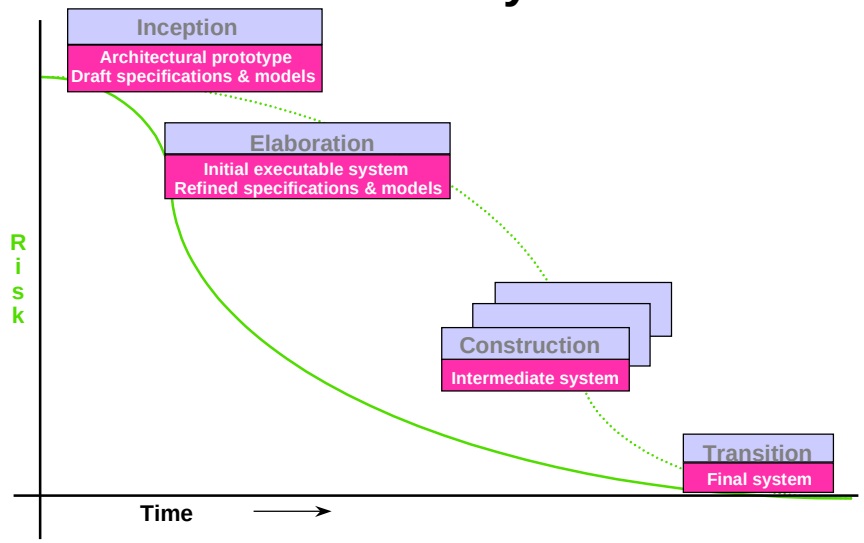
Conventional Software Process

Problem: Late Tangible Design Assessment

Standard sequence of events:
- Early and successful design review milestones
- Late and severe integration trauma
- Schedule slippage

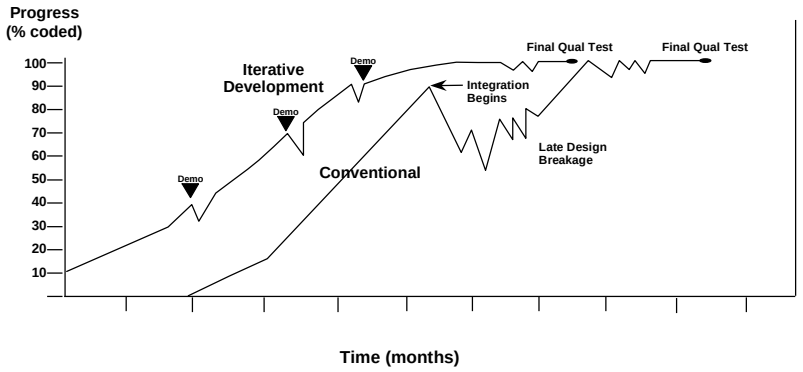


Iterative Life-Cycle Model

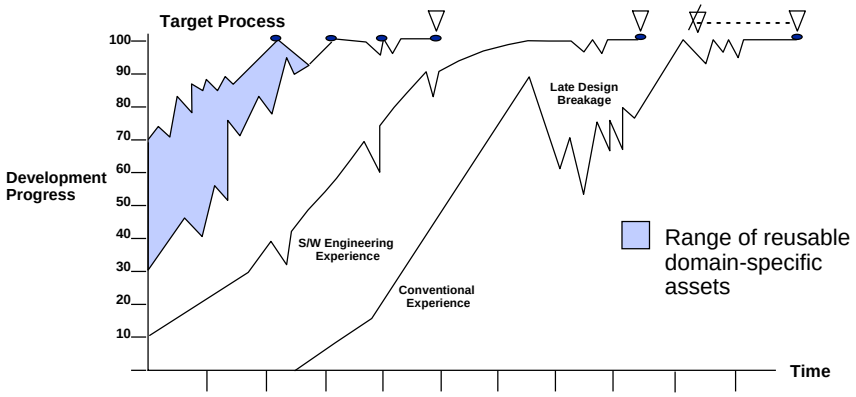


Project Results

Continuous Integration

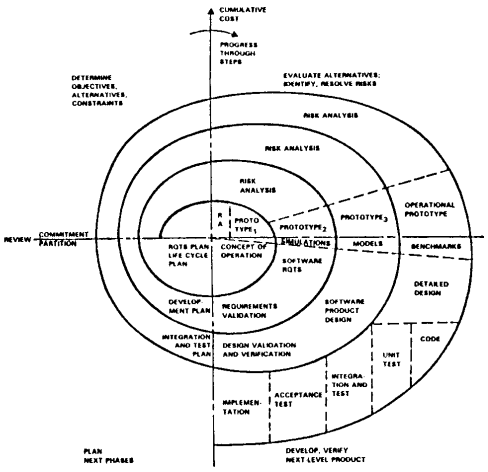


Modern Process/Technology Benefits



- Integrated environments
- Reusable components (architectures, instrumentation, etc.)
- Platform independence
- Integrated components (reuse, custom, COTS, adapted)
- Off-the-shelf solutions to most of the difficult computer science issues

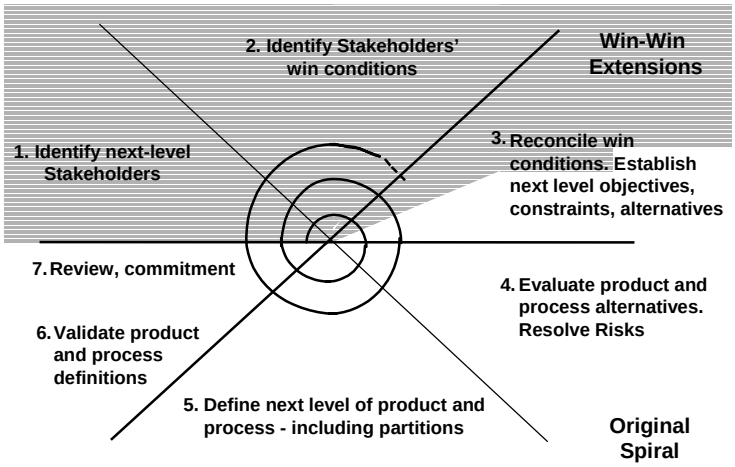
Spiral Model of Software Process



Experience with the Spiral Model

- Good for getting risks resolved early
- Good for tailoring process to situation
- Good for accommodating COTS, reuse, prototyping
- Where do objectives, constraints, and alternatives come from?
  - WinWin Spiral Model
- No common life cycle milestones
  - Anchor Points

The WinWin Spiral Model



Life Cycle Anchor Points

- **Common System/Software stakeholder commitment points**
  - **Defined in concert with Government, industry affiliates**
  - **Coordinated with Rational's Unified Software Development Process**
- **Life Cycle Objectives (LCO)**
  - **Stakeholders' commitment to support system architecting**
  - **Like getting engaged**
- **Life Cycle Architecture (LCA)**
  - **Stakeholders' commitment to support full life cycle**
  - **Like getting married**
- **Initial Operational Capability (IOC)**
  - **Stakeholders' commitment to support operations**
  - **Like having your first child**

Win Win Spiral Anchor Points

(Risk-driven level of detail for each element)

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	‣ Top-level system objectives and scope - System boundary - Environment parameters and assumptions - Evolution parameters ‣ Operational concept - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities (stakeholders)	‣ Elaboration of system objectives and scope of increment ‣ Elaboration of operational concept by increment
System Prototype(s)	‣ Exercise key usage scenarios ‣ Resolve critical risks	‣ Exercise range of usage scenarios ‣ Resolve major outstanding risks
Definition of System Requirements	‣ Top-level functions, interfaces, quality attribute levels, including: - Growth vectors and priorities - Prototypes ‣ Stakeholders' concurrence on essentials	‣ Elaboration of functions, interfaces, quality attributes, and prototypes by increment - Identification of TBD's((to-be-determined items) ‣ Stakeholders' concurrence on their priority concerns
Definition of System and Software Architecture	‣ Top-level definition of at least one feasible architecture - Physical and logical elements and relationships - Choices of COTS and reusable software elements ‣ Identification of infeasible architecture options	‣ Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain-architecture and architectural style choices ‣ Architecture evolution parameters
Definition of Life-Cycle Plan	‣ Identification of life-cycle stakeholders - Users, customers, developers, maintainers, interoperators, general public, others ‣ Identification of life-cycle process model - Top-level stages, increments ‣ Top-level WWWWWHH* by stage	‣ Elaboration of WWWWWHH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	‣ Assurance of consistency among elements above - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures	‣ Assurance of consistency among elements above ‣ All major risks resolved or covered by risk management plan

*WWWWWHH: Why, What, When, Who, Where, How, How Much

Initial Operational Capability (IOC)

- Software preparation
  - Operational and support software
  - Data preparation, COTS licenses
  - Operational readiness testing
- Site preparation
  - Facilities, equipment, supplies, vendor support
- User, operator, and maintainer preparation
  - Selection, teambuilding, training

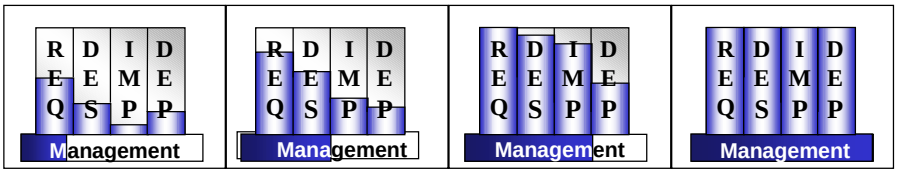
Anchor Points and Rational USDP Phases

Engineering Stage	Manufacturing Stage

Inception	Elaboration	Construction	Transition

LCO LCA IOC

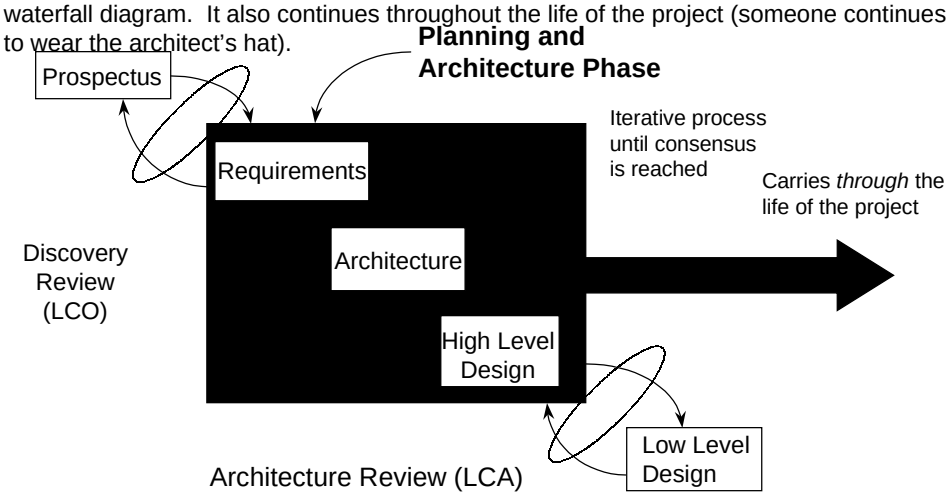
Feasibility Iterations	Architecture Iterations	Usable Iterations	Product Releases



RATIONAL Software Corporation

Architecture in a Project's Life Cycle

It encompasses the requirements, architecture and high level design phases of the typical waterfall diagram. It also continues throughout the life of the project (someone continues to wear the architect's hat).



Lucent/AT&T Architectural Review Boards:

How a Review Is Conducted

- Chairperson meets with the project to determine technical focus and required expertise for review
- Chairperson assembles review team of stakeholders and subject matter architecture experts; project sends out review material
- A 2 or 3 day review is conducted. Detailed talks are presented on key technical areas. Issues raised during discussions are recorded on cards
- Immediate readout is given to the team at the end of the review. Cards are grouped by- Things Done Right, Issues, and Recommendations
- Chairperson follow up with a written report and presentation to the project's management if requested
- Used regularly since 1988, with over 10% project savings

MBASE Invariants and Variants

Invariants	Variants
1. Defining and sustaining a stakeholder win-win relationship through the system's life-cycle.	1. Use of particular success, process, product, or property models.
2. Using the MBASE Model Integration Framework.	2. Choice of process or product representation.
3. Using the MBASE Process Integration Framework.	3. Degree of detail of process, product, property, or success modeling.
4. Using the LCO, LCA, and IOC Anchor Point milestones.	4. Number of spiral cycles or builds between anchor points.
5. Ensuring that the content of MBASE artifacts and activities is risk-driven.	5. Mapping of activities onto Inception-Elaboration-Construction-Transition phases.
	6. Mapping of staff levels onto activities.

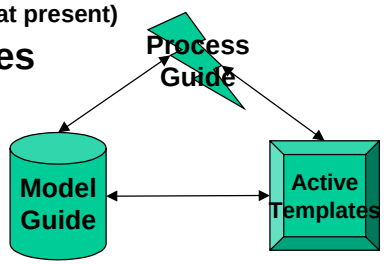
Outline

- A short history of software processes
  - Waterfall, spiral, MBASE/Rational/Benefits Realization
- MBASE Guidelines and Electronic Process Guide
- Application to digital library projects, CCPDS-R

MBASE Tools

- MBASE Model Guide
  - describes models
  - describes suggested model creation approach (variant)
- MBASE Process Guide
  - Describes activities and dynamic model integration
    - Large and non-linear so provided in hyperlinked format
    - redundant with model guide (at present)
- MBASE Active Templates

"The Trinity"



MBASE Guidelines

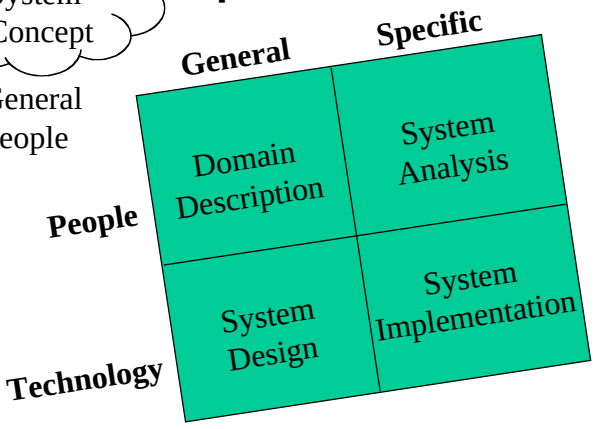
- **LCO/LCA Deliverables**
  - **Operational Concept Description**
  - **System and Software Requirements Description**
  - **System and Software Architecture Description**
  - **Life Cycle Plan**
  - **Feasibility Rationale Description**
- **IOC Deliverables**
  - **Detailed Construction Plan**
  - **Iteration Plans, Assessments**
  - **Test Plans, Procedures, Results**
  - **Inspection Plans, Reports**
  - **Detailed Design, Code, Release Notes**
  - **Transition Plan and Results**
  - **User's Manual, Training**
- **Appendices: WinWin Taxonomy, Levels of Service, Terminology**

MBASE Electronic Process Guide (1)

MBASE Electronic Process Guide (2)

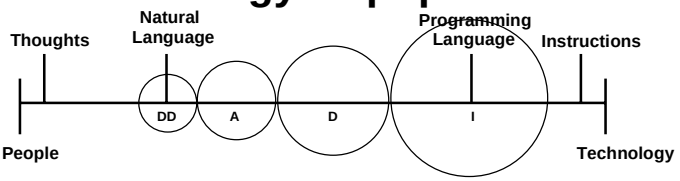
People-Technology Gap: product models

System Concept
- General
- People



System Representation
- Technology
- Specific

People-Technology Gap: product models



Characteristics of People

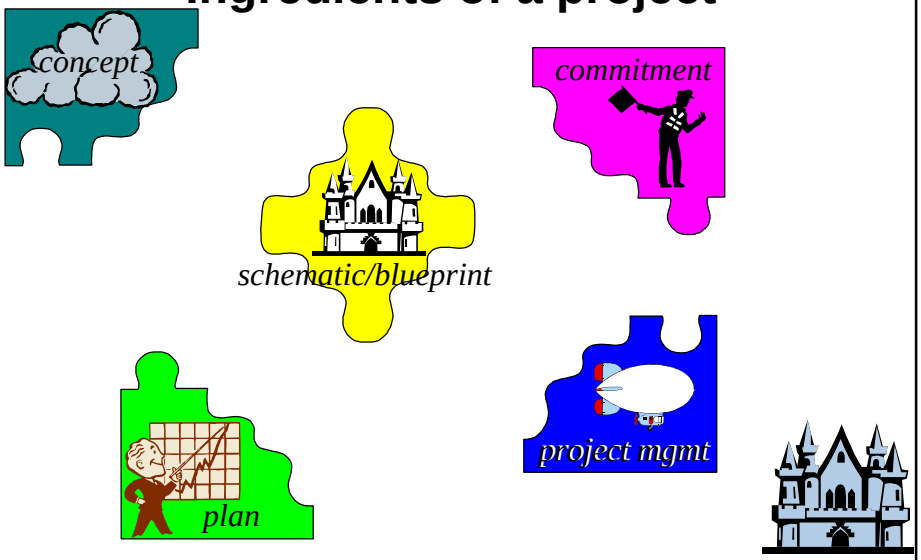
1. Non-linear
2. Abstract
3. Continuous
4. Context sensitive
5. Active
6. Creative
7. Inconsistent
8. Need doughnuts
9. Semantic
10. Highly parallel (small task)
11. Approximate (PAC, PACE)
12. Learn
13. Do as they want
14. Make choices
15. Flexible (usually)
16. Desires power
17. Informal

Characteristics of Computers

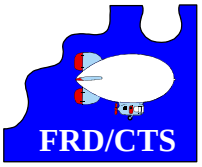
1. Linear
2. Concrete
3. Discrete
4. Context free
5. Passive
6. Logical
7. Consistent
8. Need batteries
9. Syntactic
10. Sequential
11. Exact
12. Represent
13. Do as they're told
14. Compute
15. Rigid
16. Needs power
17. Formal

"[Software] people tend to focus more on the technologies context rather than the customer." - E.Port

Ingredients of a project



MBASE Ingredients



MBASE Models*

OCD
- Org. Background
- Org. Goals
- Entity Model
- Org. Activities
- System Shortfalls
- System Capabilities
- Levels of Service
- Op. Stakeholders
- Op. Impacts

SSRD
- ~~Proj. Reqs.~~
- Capability Reqs.
- Level of Service Reqs.
- Evolution Reqs.

SSAD
- Component Model
- Behavior Model
- Enterprise Model
- Architectural Views
- Object Model
- Class Model
- Operations Model

FRD/CTS
- Business Case
- Reqs. Satisfaction
- Process Rationale
- Risk Assessment
- Iteration Plan
- Quality Plan

LCP
- Milestones and Products
- Responsibilities
- Approach, Resources

CTS
- Iterations
- Release Description
- Test Plan
- Test Results
- Inspection Report
- Users Manual

* Not exhaustive

One View: Models Needing Integration

Success Models

- OCD 2.2 Organization Goals
- OCD 2.7 Current System Shortfalls
- OCD 3.3 Levels of Service
- FRD 2.2 Requirements Satisfaction

Product Models

- OCD 2.4 Entity Model
- OCD 2.6 Org. Activity Model
- OCD 3.2 Capabilities
- OCD 2.3 Description of Current Sys
- OCD 3.4 Proposed System Descriptio
- SSRD 3.2 System Requirements
- SSRD 6. Evolution Requirements
- SSAD 2. Architectural Analysis
- SSAD 3. System Design
- CTS Release Description

MBASE

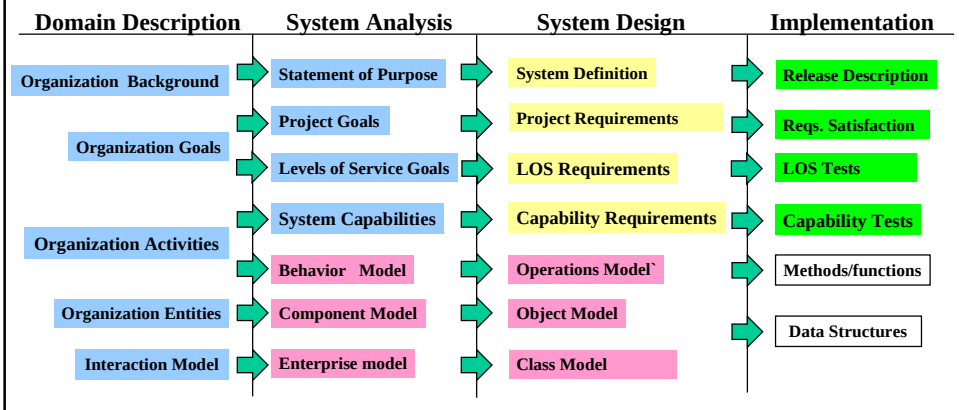
- LCP 2. Milestones and Products
- LCP 3. Responsibilities
- LCP 4. Approach
- LCP 5.1 Work Breakdown Structure
- CTS Iteration Plans
- CTS Test Plans
- CTS Transition Plans
- CTS Support Plans

Process Models

- SSRD 2. Project Requirements
- SSRD 5. Level of Service Requirements
- LCP 5.2 Budgets
- FRD 2.1 Business Case Analysis
- FRD 4. Project Risk Assessment

Property Models

Coverage/Traceability of MBASE Product Models*



Outline

- A short history of software processes
  - Waterfall, spiral, MBASE/Rational/Benefits Realization
- MBASE Guidelines and Electronic Process Guide
- Application to digital library projects, CCPDS-R

MBASE Example I - Digital Library Applications

- The Challenge
- MBASE Approach
- 1996-97 Results
- 1997-98 Results

The Challenge

- 20 Digital Library Applications
  - 2 sentence problem statements
  - Librarian clients
- 100 Graduate Students
  - 30% with industry experience
  - Largely unfamiliar with each other, Library ops.
* Develop LCA packages in 12 weeks
- Re-form teams from 30-40 continuing students
* Develop IOC packages in 12 more weeks
  - Including 2-week beta test and transition

Problem Statement #4:

Medieval Manuscripts

Ruth Wallach, Reference Center, Doheny Memorial Library

I am interested in the problem of scanning medieval manuscripts in such a way that a researcher would be able to both read the content, but also study the scribe's hand, special markings, etc. A related issue is that of transmitting such images over the network.

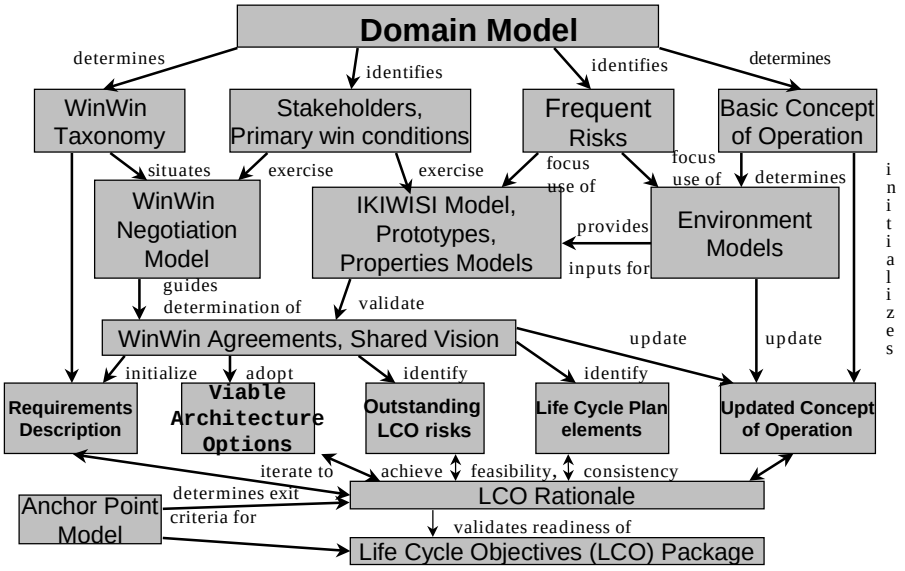
Elements of Critical Front End Milestones

(Risk-driven level of detail for each element)

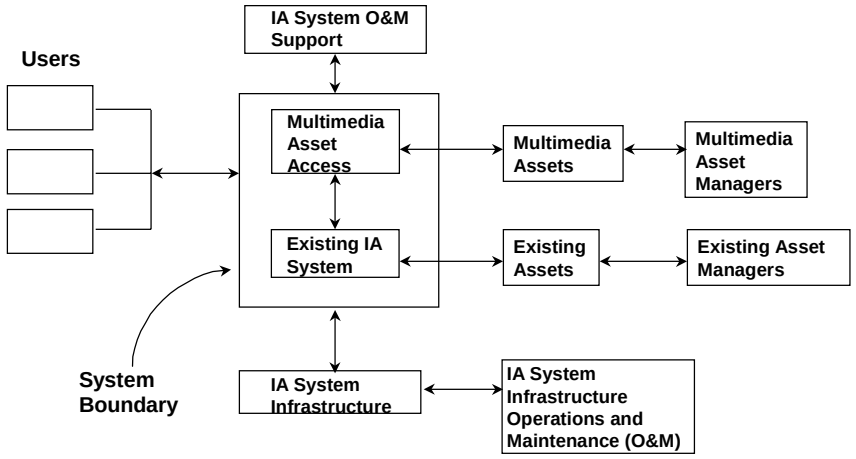
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	· Top-level system objectives and scope - System boundary - Environment parameters and assumptions - Evolution parameters · Operational concept - Operations and maintenance scenarios and parameters - Organizational life-cycle responsibilities (stakeholders)	· Elaboration of system objectives and scope of increment · Elaboration of operational concept by increment
System Prototype(s)	· Exercise key usage scenarios · Resolve critical risks	· Exercise range of usage scenarios · Resolve major outstanding risks
Definition of System Requirements	· Top-level functions, interfaces, quality attribute levels, including: - Growth vectors and priorities - Prototypes · Stakeholders' concurrence on essentials	· Elaboration of functions, interfaces, quality attributes, and prototypes by increment - Identification of TBD's((to-be-determined items) · Stakeholders' concurrence on their priority concerns
Definition of System and Software Architecture	· Top-level definition of at least one feasible architecture - Physical and logical elements and relationships - Choices of COTS and reusable software elements · Identification of infeasible architecture options	· Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain-architecture and architectural style choices · Architecture evolution parameters
Definition of Life-Cycle Plan	· Identification of life-cycle stakeholders - Users, customers, developers, maintainers, interoperators, general public, others · Identification of life-cycle process model - Top-level stages, increments · Top-level WWWWWHH* by stage	· Elaboration of WWWWWHH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	· Assurance of consistency among elements above - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures	· Assurance of consistency among elements above · All major risks resolved or covered by risk management plan

*WWWWWHH: Why, What, When, Who, Where, How, How Much

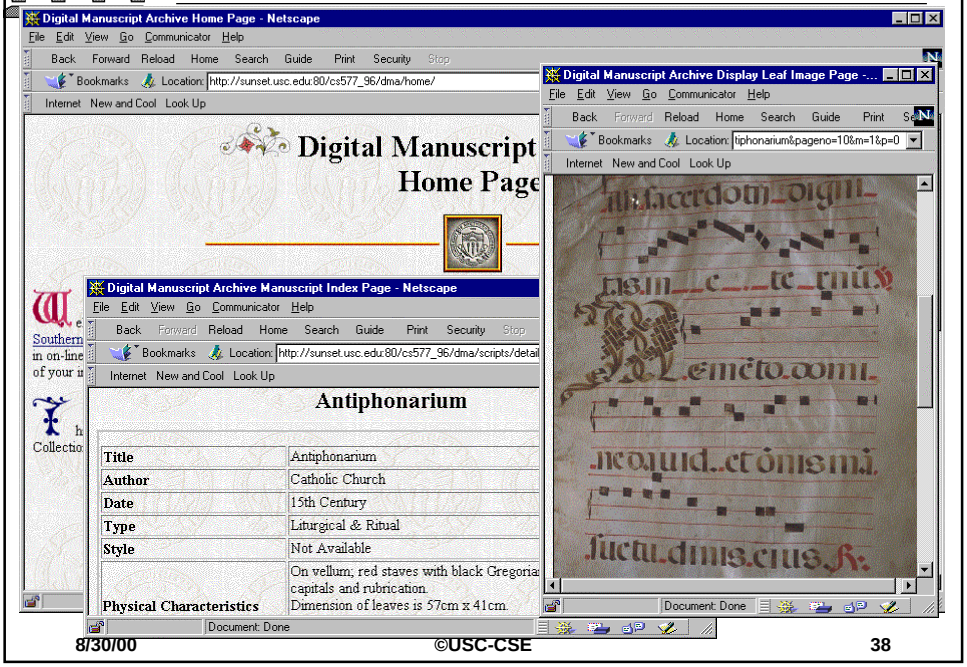
MBASE Model Integration: LCO Stage



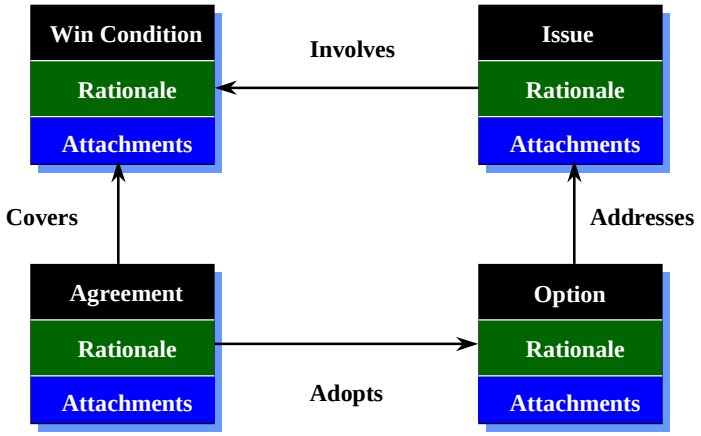
Domain Model: Block Diagram



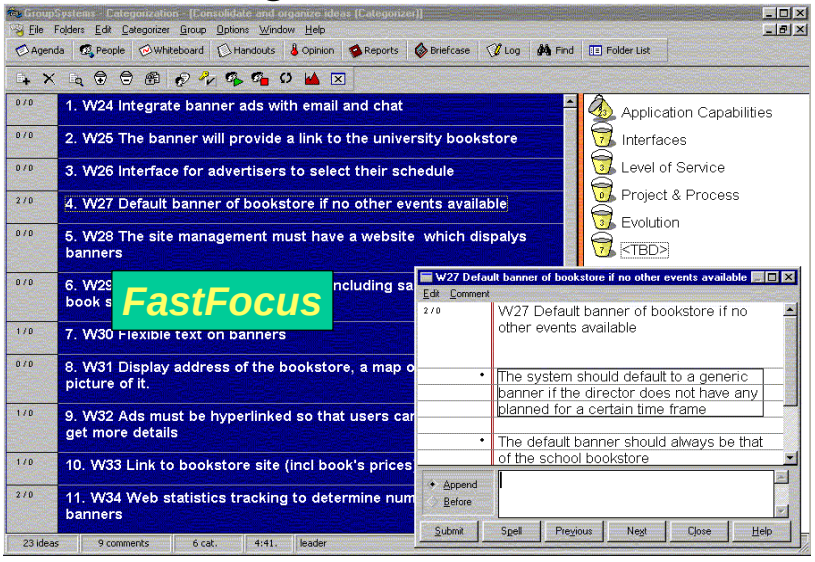
IA: Information Archive



WinWin Negotiation Model



Converge on Win Conditions



Transition Between Cycle 2 and 3

- **Smaller class (30-40 vs. 100 students)**
  - **Not a MS-CS core course**
- **Mixed team backgrounds from Cycle 1 and 2**
  - **Teams, clients need to "jell" again**
- **Library assumptions often change**
  - **Priorities, infrastructure, operations, organization**
- **Perform risk-driven rebaselining of LCA packages**
  - **First 3 weeks, with ARB reviews**

Project Results: 1996-97

- All products completed on schedule
- Librarians generally enthusiastic about products

  3-Committed to implementation
  - Cinema-TV, Business School, Tech. Reports

  2-Investing in further effort
  - Latin American Pamphlets, Medieval Manuscripts

  1-Awkward synthesis of 3 applications
  - 3 photo archives not equivalent
- Continuing in 1997-98
  - 20 candidate Library projects; mostly new

1997-98 Results

- **Better integrated LCO, LCA packages**
  - **LCO: 103 vs. 160 pages**
  - **LCA: 154 vs. 230 pages**
  - **Stronger ARB reviews**
- **Higher client satisfaction ratings**
  - **4.7 vs. 4.4 on scale of 5**
- **More effective transition into practice**
- **Improvements still needed for 1998-99**
  - **Rational Rose and OO training**
  - **Better integrated LCO, LCA packages**
  - **Better integrated architecture, product elements**

MBASE Project Experience at USC/Columbia

Metric	USC 1996-97	USC 1997-98	USC 1998-99	Columbia U-grad. 99	Columbia Grad. 99
Fall Semester: LCA Package					
Teams	15	16	20	20	13
Students	86	80	102	107	59
Applications	12	15	17	10	10
Teams failing LCO review	4	4	1	10	6
Teams failing LCA review	0	0	0	0	1
Pages, LCO package	160	103	114	124	116
Pages, LCA package	230	154	167	142	142
Client Evaluation (1-5, 5 best)	4.46	4.67	4.74	-	-
Spring Semester: IOC Package				Remained the same	
Teams	6	5	6	since projects were only	
Students	28	23	28	one semester long	
Applications	8	5	6		
Teams failing IOC acceptance review	0	0	0	0	0
Applications satisfying clients	5	5	6	20*	12*
Applications not overtaken by events	6	4	4	10	9
Applications continued	3	3	TBD	-	-
Applications used	1	3	TBD	10	5
Client evaluation	-	4.15	4.3	4.44	4.21

Unmet Expectations Problems

- **LCO success condition**
  - Describes at least one feasible architecture
  - Satisfying requirements within cost/schedule/resource constraints
  - Viable cost-effective business case
  - Stakeholder concurrence on key system parameters
- **Projects That Failed LCO Criteria**
  - 1996: 4 out of 16 (25%)
  - 1997: 4 out of 15 (27%)

why?

Requirements and Expectations: Domain Model Clashes

- **Easy/hard things for software people**

  "If you can do queries with all those ands, ors, synonyms, data ranges, etc., it should be easy to do natural language queries."

  "If you can scan the document and digitize the text, it should be easy to digitize the figures too."

- **Easy/hard things for librarians**

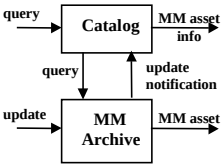
  "It was nice that you could add this access feature, but it overly (centralizes, decentralizes) control of our intellectual property rights."

  "It was nice that you could extend the system to serve the medical people, but they haven't agreed to live with our usage guidelines."

1998 Simplifier/Complicator Experiment

- Identify application simplifiers and complicators
  - For each digital library sub-domain
  - For both developers and clients
- Provide with explanations to developers and clients
  - Highlight relation to risk management
- Homework exercise to analyze simplifiers and complicators
  - For two of upcoming digital library projects
- Evaluate effect on LCO review failure rate

Example S&C's

Type of Application	Simple Block Diagram	Examples	Simplifiers	Complicators
Multimedia Archive		1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 31, 32, 35, 36, 37, 39	• Use standard query languages • Use standard or COTS search engine • Uniform media formats	• Natural language processing • Automated cataloging or indexing • Digitizing large archives • Digitizing complex or fragile artifacts • Rapid access to large Archives • Access to heterogeneous media collections • Automated annotation/description/ or meanings to digital assets • Integration of legacy systems

The Results

- **Projects That Failed LCO Criteria**
  - **1996: 4 out of 16 (25%)**
  - **1997: 4 out of 15 (27%)**
  - **1998: 1 out of 20 (5%)**
- **40% of Student critiques cited S&C's as helpful**
  - **In focusing on achievable requirements set within tight schedule**
  - **In understanding project risks and tradeoffs**

MBASE Laboratory

- **20 software engineering projects/year**
  - **5-person USC Digital Library applications, others**
- **Rapidly developing successful applications**
  - **Multimedia, virtual assistants, data acquisition**
- **Integrating models and tools**
  - **DARPA-EDCS architecture and WinWin tools**
  - **Rational Rose, Unified Modeling Language**
- **Rapidly improving artifact integration**
  - **Now using SEI Electronic Process Guide tool**
- **Results transitioning to early adopters**
- **Ultimate goal: Model-integrated SW Engr. agents**

Primary 1999-2000 Changes

- **Integration of LCO, LCA, IOC Guidelines**
- **Reworked LCO/LCA Guidelines**
  - **OCD : Shared Vision**
  - **SSRD, SSAD elaborations**
  - **LCP : less CMMI orientation**
  - **Better traceability**
  - **General guidelines with 577 addenda**
- **Example Application : Easy WinWin**
- **New Tools : Easy WinWin, MBASE Deliverables Manager**

Case Study: CCPDS-R Project Overview

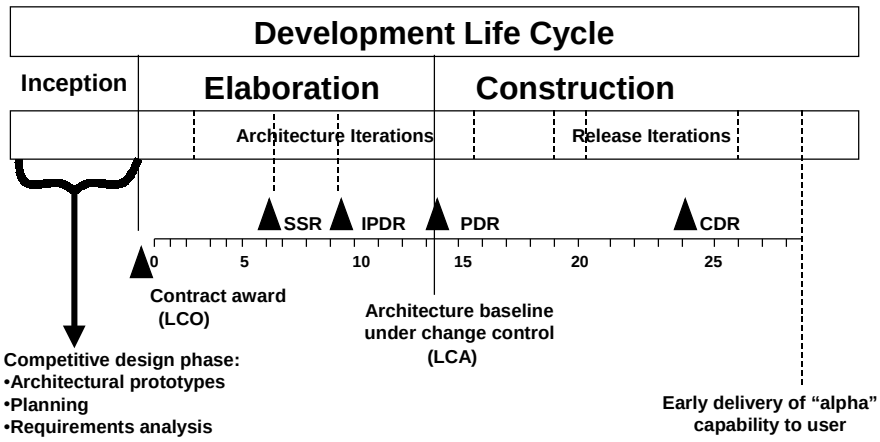
Characteristic	CCPDS-R
Domain	Ground based C3 development
Size/language	1.15M SLOC Ada
Average number of people	75
Schedule	75 months
Process/standards	DOD-STD-2167A Iterative development
Environment	Rational host DEC host DEC VMS targets
Contractor	TRW
Customer	USAF
Current status	Delivered On-budget, On-schedule

RATIONAL
Software Corporation

CCPDS-R MBASE Models

- **Success Models**
  - **Reinterpreted DOD-STD-2167a; users involved**
  - **Award fee flowdown to performers**
- **Product Models**
  - **Domain model and architecture**
  - **Message-passing middleware (UNAS)**
- **Process Models**
  - **Ada process model and toolset**
  - **Incremental builds; early delivery**
- **Property Models**
  - **COCOMO cost & schedule**
  - **UNAS - based performance modeling**
  - **Extensive progress and quality metrics tools**

Common Subsystem Macroprocess



RATIONAL Software Corporation

MBASE Conclusions

- Successfully used on Digital Library projects
  - And CCPDS-R MBASE precursor
- Key to reducing cycle time (USC RAD Workshop)
  - Top people and teambuilding
  - Prepositioning assets (people, tools, architectures, components, models)
- Key to mastering increasingly complex systems
- Complementary to, integrates existing partial models
  - CMM's, J-STD-016, ISO/IEC 12207, Architecture-based models
- Avoids many current model clashes
  - Due to uncoordinated model-element choices (mandated, legacy, default, arbitrary)