LCO ARB Feedback and MBASE LCO Wrap-Up

CS577A

2002

Outline

- Where are you at?
- LCO ARB Feedback
- Exemplars
- LCO ARB Common Problems
- Where are we going?

Congratulations…

You are (mostly) engaged to your project!

Congratulations…

You are (mostly) engaged to your project!

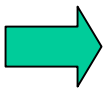
Lets review where we came from, where we are now, and where were headed

"Software Engineering": The disciplines which distinguish the coding of a computer program from the development of a software product.

	Requirements, Architecture	Design, Code	Implement, Maintain
Computer Science		CS Focus	
User Applications			
Economics			
People			

- **Accommodate new tools and techniques**
  - **Web browsers, GUI prototypers, WinWin, Spiral processes**
- **Integrate all these considerations**
  - **Via integrated models (MBASE)**

Major Class Project Milestones



September 9	--	All teams formed
September 20	--	Initial Prototypes/Storyboards
September 27	--	WinWin Negotiation Results
October 9	--	LCO Drafts on Web Site
October 15-21	--	LCO Architecture Reviews
October 28	--	LCO Package Due
November 13	--	LCA Drafts on Web Site
November 20-26	--	LCA Architecture Reviews
December 2	--	LCA Package Due
December 9	--	Individual Critiques Due

Class Project Artifacts

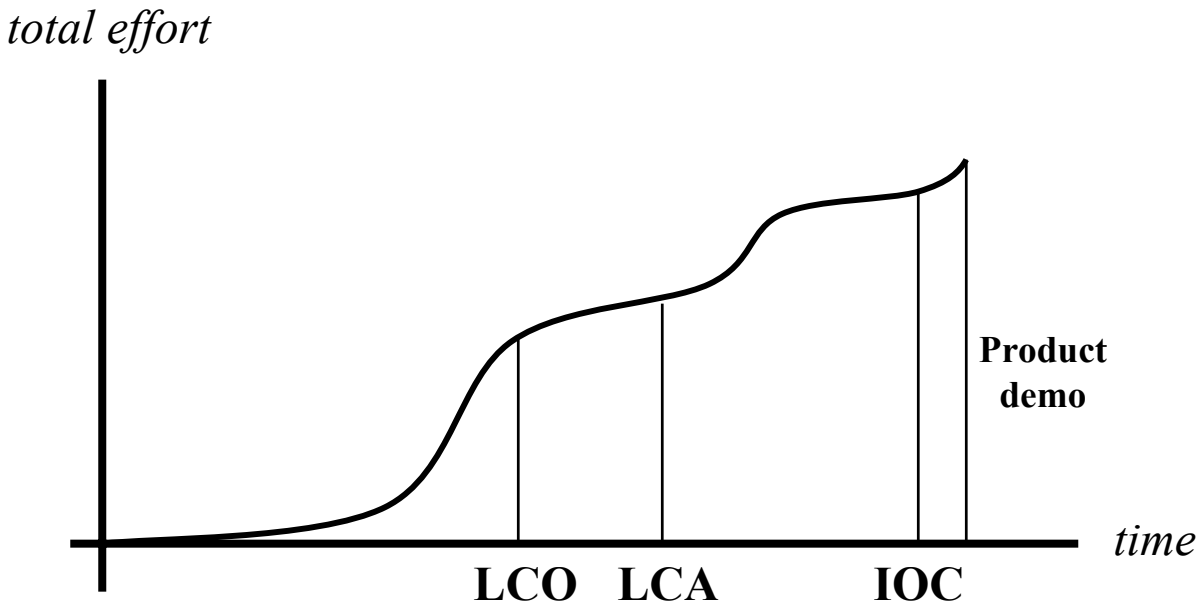
1. **An Operational Concept Definition**
2. **A Prototype of Key System Features**
3. **A System Requirements Definition**
4. **A System and Software Architecture Definition**
5. **A Life Cycle Plan**
6. **A Feasibility Rationale, assuring the consistency and feasibility of items 1-5**

For 577b projects

7. **A Detailed Construction Plan consistent with 1-6**
8. **The Test Results and Quality Assurance for 1-7**
9. **An Initial Operational Capability of the system derived from 1-8**

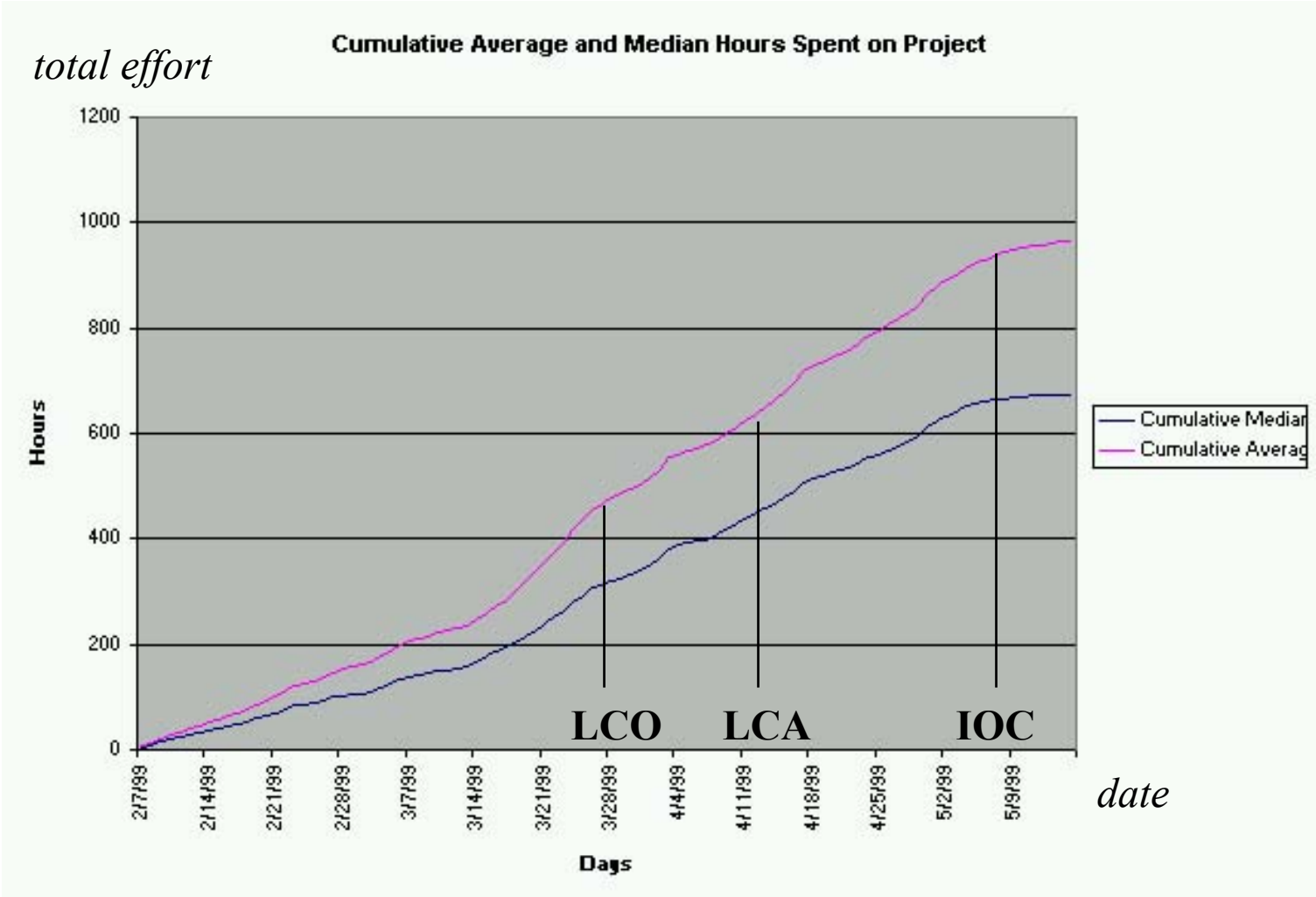
Are You Putting in the Right Effort?

Effort Graph (conceptual) for 1-semester Columbia University Course



Effort Graph CS3156 S99

http://www.columbia.edu/~cl363/research/images2.html



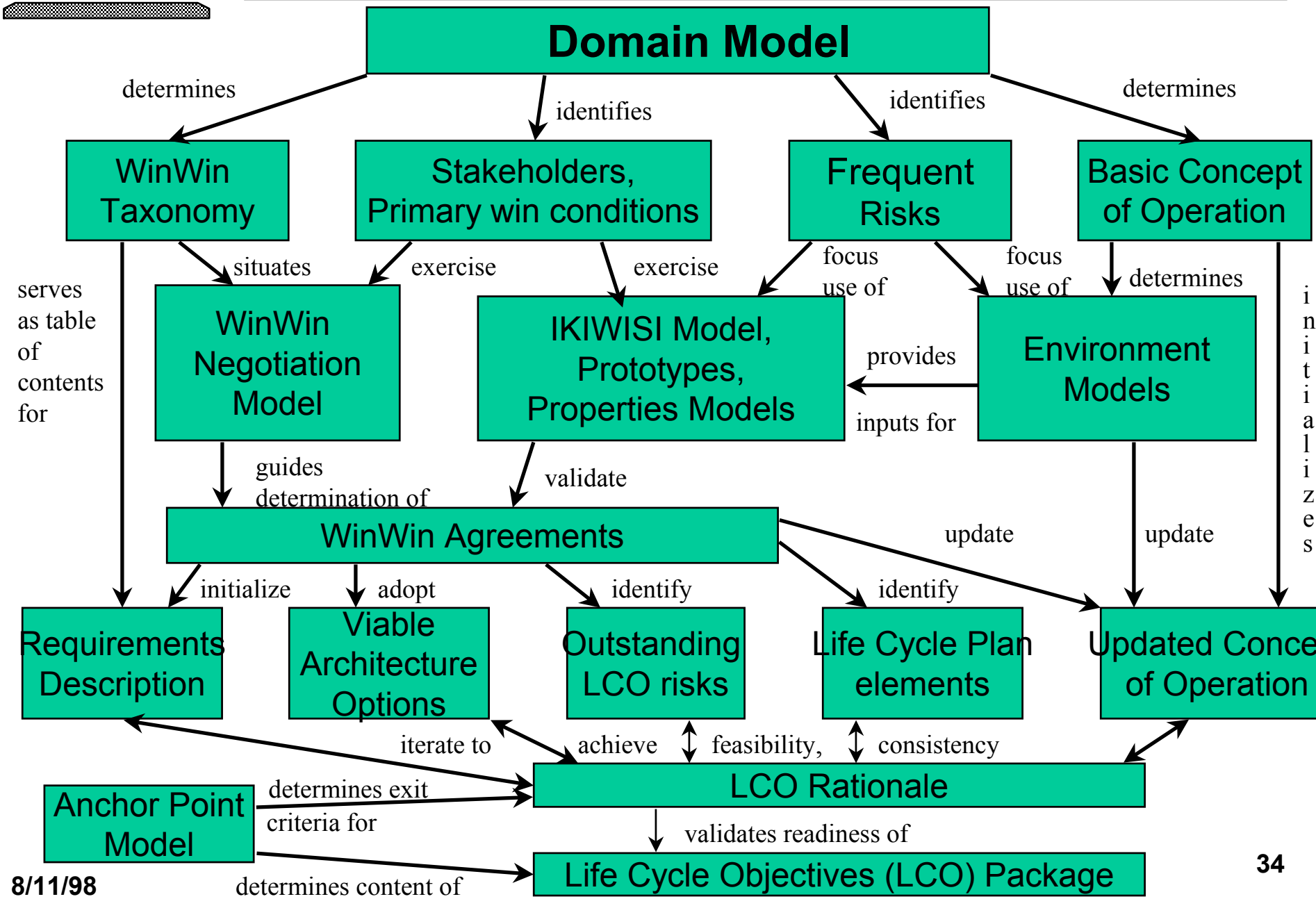
LCO – where are you at?

Success criteria:

Identify at least one feasible architecture

(***subject to risk and project considerations***)

MBASE Model Integration: LCO Stage



MBASE Milestone Elements

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	• Top-level system objectives and scope - Shared vision of expected initiatives and outcomes [COTS strategy for reuse and legacy elements] - System boundary [major COTS component boundaries within system] - Environment parameters and assumptions [CBS success factors] - Evolution parameters [major COTS vendor product availability, upgrade cycles] • Operational concept - Operations and maintenance scenarios and parameters [COTS maintenance] - Organizational life-cycle responsibilities (stakeholders) [major COTS vendors]	• Elaboration of system objectives and scope by increment • Elaboration of operational concept by increment [COTS operation and use summary, vendor support strategies, COTS adoption impact on organization]
System Prototype(s)	• Exercise key usage scenarios [demonstrate key COTS capabilities and validate interfaces, scope use and value of major COTS products] • Resolve critical risks [COTS selection and evaluation plan]	• Exercise range of usage scenarios [complete COTS evaluations, initial COTS configuration and training, initial COTS tailoring] • Resolve major outstanding risks [resolve COTS integration issues and level of use]
Definition of System Requirements	• Top-level functions, interfaces, quality attribute levels, including: - Growth vectors - Priorities - Legacy systems and environments - [Establish key COTS evaluation attributes and screening parameters] • Stakeholders' concurrence on essentials [mandated COTS packages, suppliers and vendors, identify negotiable and non-negotiable COTS requirements, COTS evaluation win-conditions and constraints]	• Elaboration of functions, interfaces, quality attributes by increment - Identification of TBDs (to-be-determined items) [mapping of COTS capabilities to requirements] • Stakeholders' concurrence on their priority concerns [concurrence on COTS imposed requirements and 90% re-worked requirements tradeoffs]

MBASE Milestone Elements (2)

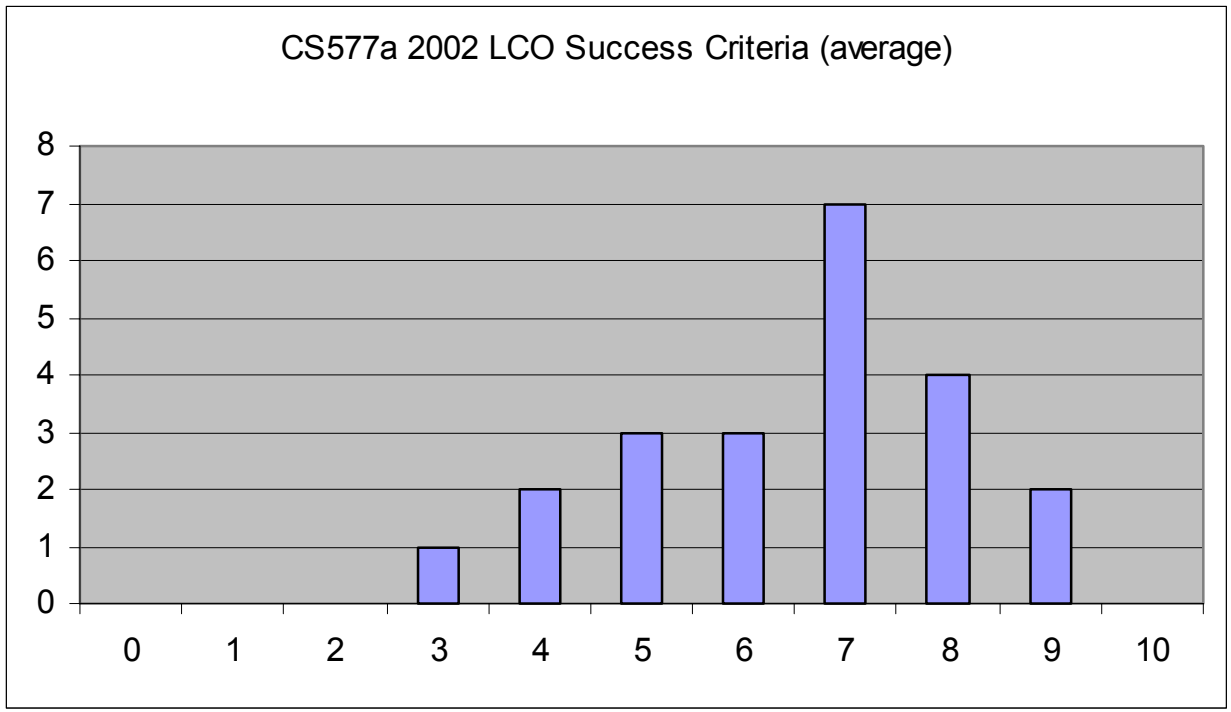
Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of System and Software Architecture	• Top-level definition of at least one feasible architecture - Physical and logical elements and relationships, including top-level domain factoring and identification of possible design patterns - Choices of COTS and reusable software elements [mapping of critical system components to COTS products] • Identification of infeasible architecture options	• Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices [CBS design (mapping of components and system factors to COTS), COTS package configurations] - Domain-architecture and architectural style choices [design for COTS glue code and wrappers] • Architecture evolution parameters
Definition of Life-Cycle Plan	• Identification of life-cycle stakeholders - Users, customers, developers, maintainers, interpreters, general public, others [COTS suppliers and vendors] • Identification of life-cycle process model - Top-level stages, increments [COTS deployment – prototyping, evaluation, schedule, risks impact of schedule and cost, stakeholders, license negotiations] • Top-level WWWWWHH* by stage [COTS product lifecycles (releases, delivery dates, costs, training needs)	• Elaboration of WWWWWHH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBDs for later increments - [COTS upgrades, patches] - [License managament (non-strandard provisions)] - [COTS users, maintainer, and developer skill set attributes] - [COTS training plan] - [COTS risks realized; integration, cultural, economic]
Feasibility Rationale	• Risk assessment and mitigation plans [COTS vendor issues – product availability, stability, costs] • Assurance of consistency among elements above - Via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures [buy versus build rationale, justification of CBS strategy, CBS tradeoffs]	• Assurance of consistency among elements above [feasibility of COTS vendor relationships, COTS maintenance costs, organization readiness to implement COTS] • All major risks resolved or covered by risk management plan [completed license contracts, COTS product alternatives and marketplace issues, COTS organization culture issues, vendor commitments]

Outline

- Where are you at?
- LCO ARB Feedback
- Exemplars
- LCO ARB Common Problems
- Where are we going?

LCO ARB Assessments

	LCO SC	OCD	Proto	SSRD	SSAD	LCP	FRD
mean	6.24	7.57	7.42	6.77	6.09	4.98	5.75
var	2.74	4.00	4.59	3.73	2.13	2.10	2.29
std	1.66	2.00	2.14	1.93	1.46	1.45	1.51



Teams failing LCO success at ARB: 3

Teams with unknown ARB success: 2

USC and Columbia Projects

Metric	USC 1996-97	USC 1997-98	USC 1998-99	USC 1999-00	Columbia U-grad. S99	Columbia Grad. S99	Columbia U-grad. F99	Columbia Grad. F99
Fall Semester:								
LCA Package								
Teams	15	16	19	22	20	13	10	7
Students	86	80	102	100	107	59	44	26
Applications	12	15	17	22	10	10	10	7
Teams failing LCO review	4	4	1	1	10	6	5	1
Teams failing LCA review	0	0	0	0	0	1	1	0
Pages, LCO package	160	103	114	-	124	116	107	95
Pages, LCA package	230	154	167	-	142	142	140	109
Client Evaluation (1-5, 5 best)	4.46	4.67	4.74	4.48	-	-	-	-
Spring Semester: IOC Package								
Teams	6	5	6	8	Remained the same since projects were only one semester long			
Students	28	23	28	35				
Applications	8	5	6	8				
Teams failing IOC acceptance review	0	0	0	1	0	0	1	0
Applications satisfying clients (*teams)	5	5	6	7	20*	12*	10*	7*
Applications not overtaken by events	6	4	4	4	10	9	10	6
Applications continued	3	3	4	4	2	3	1	2
Applications used	1	3	3	5	10	5	7	3
Client evaluation	-	4.15	4.3	4.75	4.44	4.21	3.9	4.38

How To Interpret Your ARB Feedback

Element	Eval	Things done well	Needs review
The MBASE section and the presenter and their presentation score	*How well the section (as presented) meets the LCO milestone element criteria (see MBASE milestone elements list). The values are 0 = "no elements satisfied", 10 = "all elements satisfied" and 5 = "minimum satisfactory degree of elements satisfied". Elements with low scores are at risk and should be reviewed before submitting final LCO package. A high score does not ensure the element is satisfactory (based on ARB review).*	*Comments here should be used to identify areas that were effective. Understand why and continue on this tract. Some good areas may have been overlooked.*	*Comments here indicate areas that may have problems and need to be re-considered carefully. Some good areas may have been overlooked.*
LCO Success Criteria	*Score indicates to what degree overall package satisfies LCO success criteria "identify at least one feasible architecture"*		

Outline

- Where are you at?
- LCO ARB Feedback
- Exemplars
- LCO ARB Common Problems
- Where are we going?

LCO ARB Exemplars (with respect to project)

- Team 1 (deLA)
  - Excellent OCD, complex COTS assessment project
- Team 3 (UML2Web)
  - Excellent OCD, COTS intensive
- Team 6 (Geotechnical Data Center)
  - Good SSRD, good FRD
- Team 8 (Leavey Scheduling)
  - Solid prototype, nice SSAD, good LCP
- Team 11 (Borehole Data)
  - Excellent Prototype, nice SSRD
- Team 13 (Quality Management)
  - Excellent OCD, solid results chain, prioritization
- Team 20 (Budget System)
  - Excellent prototype, solid OCD
- Team 22 (N.A. Customer Relation Management)
  - Very good overall, very good LCP, FRD

Outline

- Where are you at?
- LCO ARB Feedback
- Exemplars
- LCO ARB Common Problems
- Where are we going?

OCD Common Problems

- Results chains often missing initiatives outside of building proposed system
- Org. goals, results chains, benefits realized sometimes inconsistent
  - System capabilities sometimes presented as organizational goals
  - Results chains over focused on software
  - Benefits realized and results chain outcomes should specifically address organizational goals
- Add software maintainer as key stakeholder
- L.O.S. goals usually need performance baselines and quantitative measures, “how well” certain system capabilities are desired to perform, must reference!
- Reconcile Sys. Caps. With SSRD Sys. Behaviors

Prototype Common Problems

- Continue to explore usage issues with clients via prototype extension, end-user exercising
  - Focus on risk items, not easy-to-do things (password processing)
  - Must involve representative users
- For some projects, explore COTS level-of-service issues via prototyping, benchmarking, exercise
- Prototypes and prototyping can be used to elaborate domain description (e.g. org. activities) in addition to requirements
- Prototyping activities must be planned and managed (see LCP)
- Risky items should be prototyped if possible as soon as possible

SSRD Common Problems

- Superfluous requirements, developer initiated requirements
- Not prioritizing requirements
- Only specify elements that are actually required through WW agreements, IKIWISI prototyping, org. mandate, etc. Indicate how became required
- Validate all requirements with client
  - Levels-of-service, programming language, etc. particularly
- Specification of requirements that do not indicate how they are implementable and testable (usually vague specifications)
- Differentiate between OCD Sys. Caps. And SSRD Sys. Caps (e.g "core" requirements)
- MR, MRS, MARS integrations incorrect
  - Project refers to goals, L.O.S. refer to capabilities
- Incorrect specification of requirements
  - Used MARS for Sys. Caps., did not use MARS for L.O.S. and Project
- Some level-of-service requirements overly vague
  - Easy to use, high availability, "good" performance
  - Specify MARS criteria, 'R' for relevant Sys. Caps. specifically 'S'
- Evolution requirements to specified as "implementable and testable"
  - use them to constrain architecture

SSAD Common Problems

- Some OCD-SSRD-SSAD traceability problems and model clashes
  - OCD Sys. Caps ← SSAD Behaviors, SSRD Sys. Caps. ← SSAD Operations
- Design elements in Component Model ("MySQL")
- Ineffective description of designs with COTS
- Too little or too much design for LCO
  - Depends on project type (1-semester needs some design, COTS assessment needs top-level only with alternatives), where and when will the project be completed?
- If you see same items in the design as components then those items are design elements
- If you see something that is both a component and a behavior then you have a model clash
- Component relationships not labeled
- No alternative architectures specified
- Confusion over Enterprise class model

LCP Common Problems

- Missing project specific details
  - Avoid copy of MBASE guidelines, general stuff, detail of class stuff)
  - Need “plans for plans” and placeholders for uncertain activities
- Be concise about deliverables (do not expand LCO, LCA package elements, reference guidelines if needed)
- Process choice arbitrary and not planned, no elaboration on activity element plans, milestones, and schedule
- Stakeholder responsibilities need project specifics
- Be much more specific about 577b plans
  - Roles, skills, risks, tasks, milestones, increments, test and transition planning, preparation, and execution
  - CTS activity planning must be included now!
- Risk mitigation and contingencies not planned
  - All risks affect schedule, plans, resources (costs)
- More thorough risk management plans
  - Contingencies, missing risks, project specific risks
- Difference between FRD and LCP risks
  - Must be consistent and complementary
  - LCP “what, management activities, impact on schedule and resources”, FRD “how bad, worst case and typical case, why management plans will succeed”

FRD Common Problems

- FRD Business case
  - Cost estimates need to consider non-$ costs such as effort (esp. developer), utils. etc.
  - Quantify costs and value (for ROI), not qualitative
  - Include project success criteria
  - Value estimates need to tie to OCD Org. Goals, Results Chain, Benefits Realized
  - ROI summaries need to consider initial costs, ongoing costs, and future costs as compared to initial value, ongoing value, and future value
  - Quantify business-case cost and value via representative scenarios
  - Include COCOMO cost driver rationale when applicable
- Focus on 577b risks: skills, schedule, COTS
  - Big risks (timely or adequate COTS) should have fallbacks
- Process rationale should identify specific core capability
  - And propagate it to LCP milestones
- Operations and maintenance effort by people already in organization is not free!! (re-allocation of tasks)

General LCO Package Feedback

- Confusion on applying MBASE guidelines to COTS
  - Uses-COTS, COTS assessment, COTS Tailoring, COTS Glue code
- Confusion over use of guidelines for different project types
  - Some projects need to tailor guidelines!! (feasibility, adv. Proto., etc.)
- Use of old version of MBASE guidelines and templates
- More consistency across documents
  - System definition across OCP, SSRD
  - Risks in LCP, FRD
  - Levels-of-service in OCD, SSRD, FRD
- Avoid duplicating text
  - Better to cross-reference; makes update easier
- Use client-oriented, not abstract terms
  - Option 1, 2, 3; data items a, b, c\
- Some projects need to start consideration of CTS plans now

Outline

- Where are you at?
- LCO ARB Feedback
- Exemplars
- LCO ARB Common Problems
- Where are we going?

The Road Ahead: LCA

Success criteria:

Commit to one feasible architecture

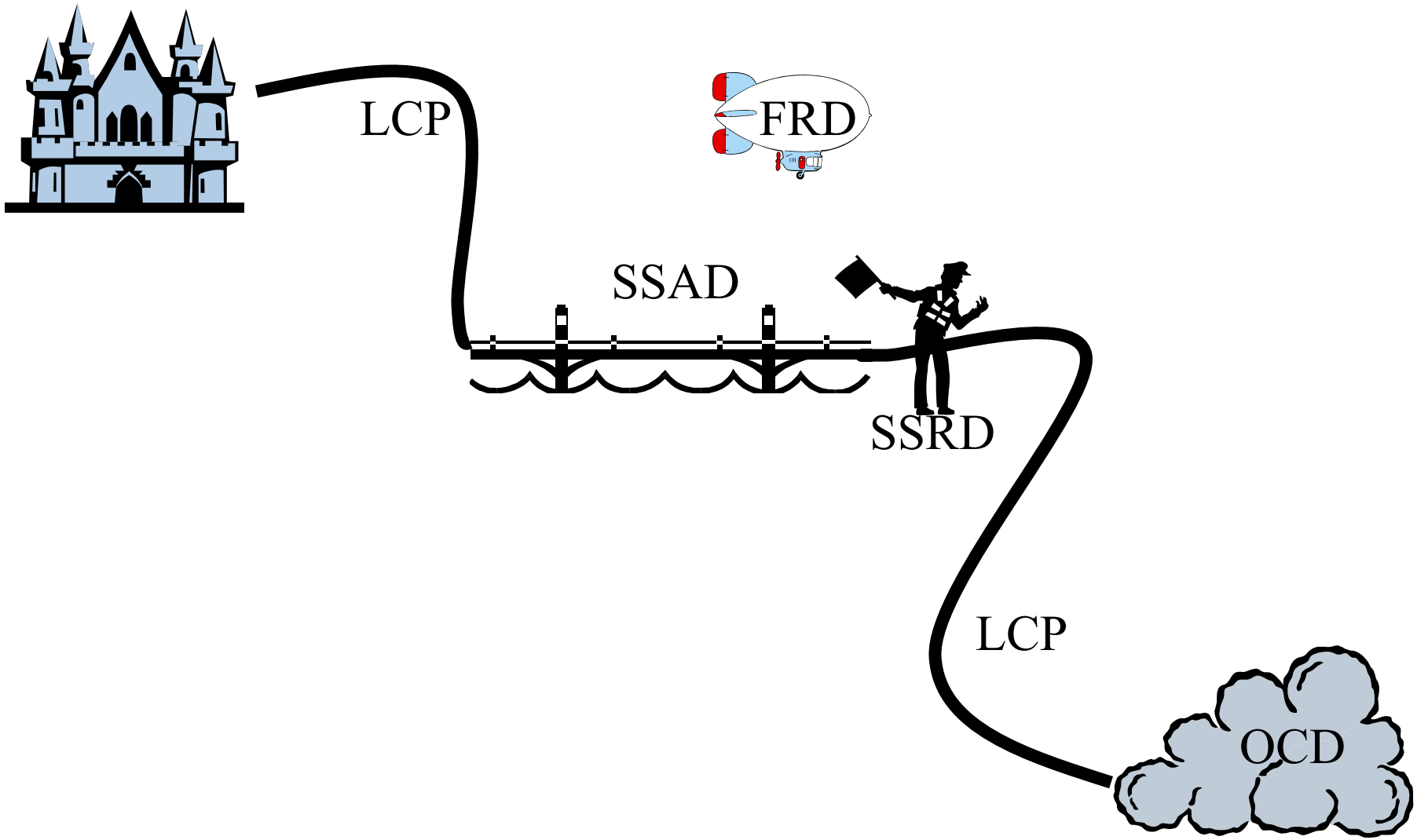
Pre-wedding plans

- What do you need to do between LCO and LCA?
  - Design
    - Resolve all significant architectural issues
    - Advanced prototyping
  - Refine
    - Resolve model clashes
    - Add significant details
    - Remove non-significant details - no “fluff”
    - Schedule, roles, commitments, effort estimates, etc.
  - Justifine! (Justify)
    - Assure architecture is faithful to concept
    - Assure value of system vs. stakeholder investment
    - Reduce risk exposure

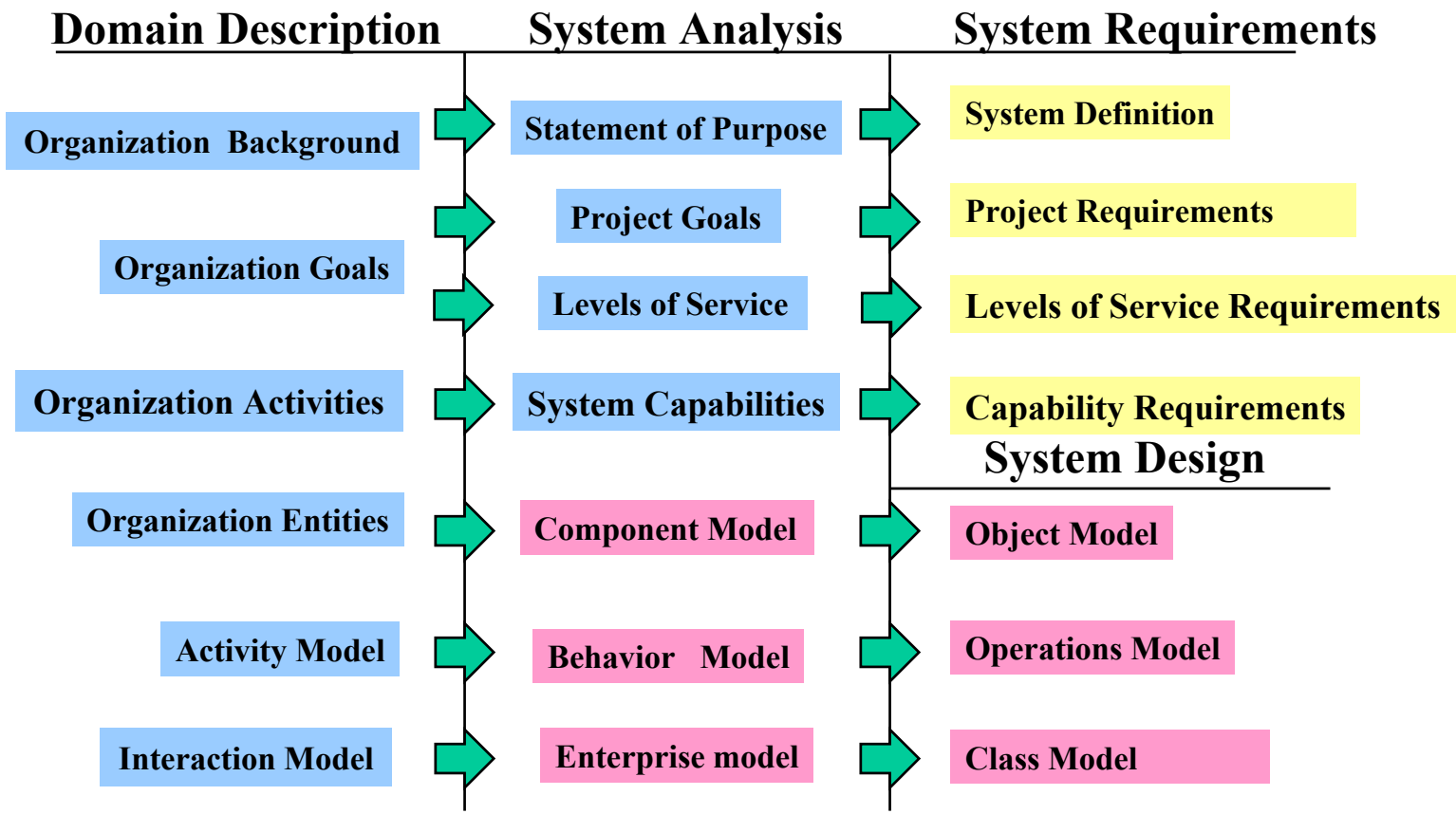
What does this involve?

- Several iterations through MBASE models
  - Model integration and integrity paramount
- Refine WinWin negotiations
  - Close out old issues
  - Cover all win conditions
  - Identify and deal with new win conditions
- Writing code
  - Advanced prototyping to resolve risky architectural issues
  - Head start on implementing critical requirements (for assurance, schedule, etc.)

Getting From A to B



Integration of MBASE System Definition Elements



Operational Concept Description (OCD)

System and Software Requirements Definition (SSRD)

System and Software Architecture Description (SSAD)

Modeling versus Documenting

- MBASE describes models, how to build them, and ways to communicate them.
- Documentation is a necessary consequence of modeling
  - why?
  - Also, the act of writing something down helps you solidify a model
- Communication is an essential part of the collaborative development approach
- Avoid documenting for documentation sake!
  - Everything you document should be the result of modeling
  - everything you document should have value and meaning to stakeholders (think about risk here)

The documentation are the models!!!

The Models are the documentation!!!

Upcoming lectures

- Analysis and design
  - More in depth about components, objects, behaviors, operations
- Scheduling
  - Effort estimation
- Model clash resolution
- COTS Process
- Project Management
- Construction plans
- Specialty areas

Life Cycle Architecture (LCA)

- more formal, with everything tracing upward and downward
- no major unresolved issues or items, and closure mechanisms identified for any unresolved issues or items (e.g., “detailed data entry capabilities will be specified once the Library chooses a Forms Management package on February 15”)
- no more TBD's
- there should no longer be any "possible" or "potential" elements (e.g., Entities, Components, …)
- no more superfluous, unreferenced items: each element (e.g., Entities, Components, …) either should reference, or be referenced by another element. Items that are not referenced should be eliminated, or documented as irrelevant

The Road Ahead Ahead: IOC

Success criteria

The initial operational capabilities of the system as constructed satisfy the architecture models

* This includes all the MBASE models, not just SSAD

** Completeness is not as important as soundness