



Section 01: Life Cycle Objectives Review

Valentin Razmov

23 Jun 2006

CSE403, Summer'06, Section 01



Resources

- n "*Anchoring the Software Process*", Barry Boehm

23 Jun 2006

CSE403, Summer'06, Section 01



Outline

- n Life Cycle Objectives Review milestone – group assignment #1
- n The five constituent elements of a Life Cycle Objectives Review

23 Jun 2006

CSE403, Summer'06, Section 01



Life Cycle Objectives (LCO) Review milestone

- n Group assignment #1: Project Proposals
 - n You need to work in pairs, so look for a partner
- n Assignment is available on the course web
- n Due next Wednesday, June 28 @ 10pm
- n Project Proposal presentations in-class on Thursday, June 29
 - n ~10 minutes per presentation, so we can hear all
- n Today we cover the necessary elements of a proposal (a.k.a. Life Cycle Objectives Review).

23 Jun 2006

CSE403, Summer'06, Section 01



Life Cycle Objectives Elements

- n **Operational Concepts**
 - n What is it? (High-level scope and objectives)
- n **System Requirements**
 - n What does it do for us? (Lower-level actual planned deliverables)
- n **System and Software Architecture**
 - n How? (Technically)
- n **Lifecycle Plan**
 - n Who wants it? Who'll support it? (Resources needed)
- n **Feasibility Rationale**
 - n Given the constraints, is this realistic (can it be built)?

23 Jun 2006

CSE403, Summer'06, Section 01



1: Operational Concepts

- n **Top-level system objectives and scope**
 - n What problem are you trying to solve? Why? For whom?
 - n User community, environment, major benefits?
 - n Goals and non-goals
 - n To set realistic expectations in the audience

Tip: This is what you should be able to explain in a 1-minute pitch (if you didn't have more time) – a.k.a. "an elevator pitch."

Tip: It takes practice to refine this, so start early.

23 Jun 2006

CSE403, Summer'06, Section 01



2: System Requirements

Essential features of the system

- n What does the customer want from this system?
 - n Look from the user's perspective
 - n **Tip:** Avoid details at the start; there's time to evolve.
- n Discuss main capabilities, outcomes, reliability and performance needs, appearance
- n *Customer involvement* is important and beneficial
 - n They know best what their interests and needs are, including what fits in their daily work and life patterns
 - n ... even if they may not always express it very well
 - n They understand the domain better than developers do.
 - n Working jointly and openly with customers helps build trust, so any necessary changes are more acceptable.

23 Jun 2006

CSE403, Summer'06, Section 01



2: System Requirements (cont.)

Essential features of the system

- n This will be your *initial* written specification
 - n Customers can review and sign off quickly or complain early.
 - n Putting it in writing makes it less ambiguous than saying it.
 - n Forces you to think of major functional areas and seek architectural defects early
 - n "Failing to write a spec is the *single biggest unnecessary risk* you take in a software project" -- Joel Spolsky
- n Be concise yet complete
 - n People get attached to their work even if it is no longer of value.
- n **Tip:** A picture / diagram is (often) worth 1000 words.
- n **Tip:** Scenarios and stories help, but avoid being verbose.

23 Jun 2006

CSE403, Summer'06, Section 01



3: System and Software Architecture

High-level technical description but with enough detail to allow feasibility analysis

- n Unlike the previous two elements, this is technical.
- n Architectural flaws will only deepen as you go forward, so look for alternatives while it's still early.
- n **Tip:** Try to come up with several (at least 3) alternative architectural designs.
- n **Tip:** Identify clients, servers, major software components, external 3rd party software, and the interactions between them.
- n **Tip:** Pictures say 1000 words.

23 Jun 2006

CSE403, Summer'06, Section 01



4: Life Cycle Plan

Identify stakeholders and their roles

- n Users, architects, developers, testers, managers, etc.
- n WWWWWHH:
Why / What / When / Who / Where / How / How
 - n Objectives: Why is the system being developed?
 - n Schedules: What will be done, When?
 - n Responsibilities: Who will do it? Where are they?
 - n Approach: Whow will the job be done?
 - n Resources: Whow much of each resource?
- n **Tip:** Make your best (educated) guess. Some of this will necessarily change. This is *not* a contract.

23 Jun 2006

CSE403, Summer'06, Section 01



5: Feasibility Rationale

Conceptual integrity and compatibility

- n Can this really be built with the available resources?
- n Identify project risks
- n What are the assumptions? Any unwarranted ones?
 - n "If you make one or two ridiculous assumptions, you'll find everything I say or do totally justified."
-- Ashleigh Brilliant, 1671
- n **Tip:** Keep asking "why" until the assumptions emerge.

- n E.g., "Why do we need this?", "Why is this good?"

23 Jun 2006

CSE403, Summer'06, Section 01