



Administrivia

- n Lifecycle Architecture (LCA) group assignment will go out later today.
- n Informal feedback meetings with LCO groups
 - n EasyShare: Mon, 2:45pm-3:15pm, room 624
 - n TeamForge: Mon, 3:15pm-3:45pm, room 624
 - n FantasySportsLeague: ??
 - n OpEnSpaCe: ??

03 Jul 2006

CSE403, Summer'06, Lecture06



Lecture 06: Lifecycle Architecture Review

Valentin Razmov

03 Jul 2006

CSE403, Summer'06, Lecture06



Outline

- n Team Conversations: a few remarks
- n Overview of Lifecycle Architecture (LCA) phase

03 Jul 2006

CSE403, Summer'06, Lecture06



Resources

- n "*Anchoring the Software Process*", Barry Boehm
 - n Especially pp. 1-10
- n "*Death March*", by Edward Yourdon
 - n Ch. 2: pp. 56-57

03 Jul 2006

CSE403, Summer'06, Lecture06



Team Conversations: Remarks

- n Goal is to establish shared understanding among teammates
 - n Done upfront or while projects are ongoing
- n Conversations to hold **before project starts**:
 - n Success criteria
 - n Done!
 - n Safety
 - n What conditions would make you feel welcome and safe to express your creativity and personality?
 - n Commitment
 - n What is your level of commitment to this project?
 - n See "*Death March*", pp.56-57.
 - n Feedback preference
 - n How would you like to be rewarded / criticized?

03 Jul 2006

CSE403, Summer'06, Lecture06



The Next Project Stage: Lifecycle Architecture (LCA) Review

- n Culmination of the detailed planning and design
- n An elaboration of the LCO review document
 - n Requires that the same five elements be addressed
 - n But more details and decisions are expected, and fewer open options
 - n LCO materials serve as a starting point
 - n But no reason to stay too close to the original idea if your expanded team wants to change/adapt some aspects
 - n Changes between LCO and LCA are frequently needed to improve focus and/or scope.
 - n Major risks are resolved or a management plan created.
- n Stakeholders must approve the decisions and the overall strategy before the project moves forward.

03 Jul 2006

CSE403, Summer'06, Lecture06



Basic LCA Elements

- n **Operational concept**
 - n Defines user community, environment, what it does and does not do, major benefits for all stakeholders
- n **System requirements**
 - n *All* features defined at this stage, including performance, reliability, security
- n **System architecture**
 - n Identifies any changes from original architecture, existing packages to be used, areas where changes are (proactively) anticipated in the design
- n **Lifecycle plan**
 - n Defines major milestones, maps tasks to resources, identifies roles and responsibilities
- n **Feasibility rationale**
 - n Remaining risks: likelihood, impact, risk mgmt plans

03 Jul 2006

CSE403, Summer'06, Lecture06




LCA Milestone Deliverables – at a High Level

- n Detailed requirements specification document
 - n Shows that you understand well what is being built
- n Detailed design document
 - n Shows that you know how to build it
- n Test plan document
- n Schedule and task assignments
- n Presentation

Note: These are all *evolving* documents, not set in stone. The LCA deliverables represent your latest understanding of what the issues are and what solutions you are envisioning.

03 Jul 2006

CSE403, Summer'06, Lecture06



Describing Specifications, Architectures, and Other Beasts

- n A specification or design document must be understandable not only to its author.
 - n Standard ways are needed to unambiguously express common relationships in a system.
- n Standard notations exist, some of which we will discuss in class
 - n Diagrams, description languages, etc.
 - n You will be expected to use those in your LCA documents.

03 Jul 2006

CSE403, Summer'06, Lecture06



LCA Milestone Deliverables – in a Bit More Detail (1/3)

- n **Detailed requirements specification document**
 - n From the point of view of the customer
 - n Techniques: use cases, commonality and variability analysis, prototyping
- n **Detailed architecture document**
 - n Technical, reasonably detailed description of:
 - n system modules and interfaces between them
 - n UIs
 - n Identifies assumptions and high-risk areas and proposes realistic solutions and/or alternatives
 - n Notations: state/dataflow diagrams, sequence diagrams, class diagrams; UML

03 Jul 2006

CSE403, Summer'06, Lecture06



LCA Milestone Deliverables – in a Bit More Detail (2/3)

- n **Test plan document**
 - n High level strategy: what will be tested, what won't, and why
 - n How you plan to test your product in a disciplined way
 - n If you can't test it easily, something is likely wrong with the design.
 - n A set of actual test cases, derived based on an established methodology
 - n Methodology:
Structure/Function/Data/Platform/Operations (SFDPPO)
- n **Schedule and task assignments**
 - n The actual plan of work
 - n How specifically will the project be split into milestones?
 - n Who will take on which piece of the puzzle?
 - n Tools of the trade: Microsoft Project; Excel

03 Jul 2006

CSE403, Summer'06, Lecture06



LCA Milestone Deliverables – in a Bit More Detail (3/3)

- n **Presentation**
 - n Before your (true or surrogate) customers
 - n Will allow more time to cover important aspects and for audience to dissect weaker points

03 Jul 2006

CSE403, Summer'06, Lecture06

Parallels between LCA Milestone Deliverables and Core LCA Elements

Milestone Deliverables LCA Elements

- n Overview presentation -> Operational concept
- n Specification document -> System requirements
- n Architecture document -> System architecture
- n Team structure, schedule, task assignments, risks -> Lifecycle plan, Feasibility rationale
- n Test plan -> *missing in Boehm's LCA*

03 Jul 2006

CSE403, Summer'06, Lecture06

Test Plan

- n "Failing to plan is planning to fail."
- n Describes what you want to test and how you will do it
 - n Must match the tests you will create (are creating)
- n Follows a disciplined methodology (e.g., SFDPO)
 - n Some types of tests may not fit nicely, so invent your own categories

Issues to consider:

- n Do you have tests in each of the categories?
- n Do your proposed tests cover all use cases?
- n Which tests are critical / high priority?

03 Jul 2006

CSE403, Summer'06, Lecture06

Test Planning using the SFDPO Methodology

Structure (what the product is):

- n What files does it have? Do I know anything about how it was built? Is it one program or many? What physical material comes with it? Can I test it module by module?

Function (what the product does):

- n What are its functions? What kind of error handling does it do? What kind of user interface does it have? Does it do anything that is not visible to the user? How does it interface with the operating system?

Data (what it processes):

- n What kinds of input does it process? What does its output look like? What kinds of modes or states can it be in? Does it come packaged with preset data? Is any of its input sensitive to timing or sequencing?

Platform (what it depends upon):

- n What operating systems does it run on? Does the environment have to be configured in any special way? Does it depend on 3rd party components?

Operations (how it will be used):

- n Who will use it? Where and how will they use it? What will they use it for? Are there certain things that users are more likely to do? Is there user data we could get to help make the tests more realistic?

03 Jul 2006

CSE403, Summer'06, Lecture06

What Comes Next?

- n Requirements gathering techniques
- n Architecture: descriptions, etc.
- n Design: strategies, principles

03 Jul 2006

CSE403, Summer'06, Lecture07