Section 01: Life Cycle Objectives Review

Valentin Razmov

Outline

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

Life Cycle Objectives Review milestone

- Group assignment #1: Project Proposals
  - You need to work in pairs, so find a partner
  - Assignment is available on the course web
  - Due next Tuesday, June 28 @ 10pm
  - Project Proposal presentations in-class on Wednesday, June 29
    - 7-8 minutes per team, so that we can fit everyone
  - This lecture covers the necessary elements of a proposal (a.k.a. Life Cycle Objectives Review).

Life Cycle Objectives Elements

- Operational Concepts
  - What is it?
- System Requirements
  - What does it do for us?
- System and Software Architecture
  - How?
- Lifecycle Plan
  - Who wants it? Who’ll support it?
- Feasibility Rationale
  - Is this really true?

1: Operational Concepts

- Top-level system objectives and scope
  - What problem are you trying to solve? Why? For whom?
  - User community, environment, major benefits, goals and non-goals
  - What you would be expected to say in a 1-minute “elevator pitch”
2: System Requirements

- Essential features of the system
- What does the customer want from this system (avoid details at the start)?
  - Look from the user's perspective
- Discuss main capabilities, outcomes, reliability and performance levels, appearance
- Involving the customer at this stage is very beneficial, since they best know their interests and needs
  - Even if they may not always express those very well

2: System Requirements (cont.)

- Essential features of the system
- This will be your initial written specification
  - The customer can review it and sign off quickly or complain early.
    - If you put it in writing, it's less ambiguous than if you just say it.
    - Forces you to think of major functional areas, major architectural defects
  - "Failing to write a spec is the single biggest unnecessary risk you take in a software project" -- Joel Spolsky
- Be concise yet complete
  - People get attached to their work even if it is no longer of value.
  - A picture/diagram is (often) worth 1000 words.
  - Scenarios/stories help, but don't be verbose.

3: System and Software Architecture

- High-level description but with enough detail to allow feasibility analysis
  - Can this really be built with the available resources?
  - Try to come up with several (at least 3) alternative architectural designs
  - Architectural flaws will only deepen as you go forward, so look for alternatives while it's still early.
  - This is necessarily a technical discussion (unlike the prior two elements).

4: Life Cycle Plan

- WWWWHH: Why/What/When/Who/Where/How/How
  - Objectives - Why is the system being developed?
  - Schedules - What will be done, When?
  - Responsibilities - Who will do it? Where are they?
  - Approach - How will the job be done?
  - Resources - How much of each resource?

5: Feasibility Rationale

- Conceptual integrity and compatibility
  - What assumptions are you making? Any unwarranted ones among them?
    - "If you make one or two ridiculous assumptions, you'll find everything I say or do totally justified"
      -- Ashleigh Brilliant, 1671