UNIT-III

Elaboration

MADHESWARI.K
AP/CSE
SSNCE

What is Elaboration?

- Elaboration often consists of two or more iterations(2 to 6 weeks duration)
- Each iteration is time-boxed(i.e. End Date fixed)
- Elaboration is not Design Phase(i.e. the model is not fully developed)
- Also it is not throw away Prototype;rather the code and design are production quality

- In other words, Elaboration is the initial series of iterations during which
  - The core ,risky software architecture is programmed and tested
  - The majority of requirements are discovered and stabilized
  - The major risks are mitigated or reduced

Define Elaboration

- Build the core architecture, resolve the high risk elements, Define most Requirements, and Estimate the overall schedule and resources

Define the first iteration is elaboration phase

- Iteration-1 of Elaboration Phase emphasizes fundamental and common OOA/D skills used in building OO Systems.

Example – NextGen POS

Iteration 1 Requirements

- The requirements for the first iteration of the NextGen POS application follow:
  - Implement a basic, key scenario of the *Process Sale* use case: entering items and receiving a cash payment.
  - Implement a *Start Up* use case as necessary to support the initialization needs of the iteration.
  - Nothing fancy or complex is handled, just a simple happy path scenario, and the design and implementation to support it.
  - There is no collaboration with external services, such as a tax calculator or product database.
  - No complex pricing rules are applied.
  - The design and implementation of the supporting UI ,database,are done(Not in detail)
  - Subsequent iterations will grow on this foundation.

Elaboration Overview

- Program and Test, core and critical parts of the project
- Achieve a stable version of the majority of the requirements
- Mitigate or retire major risks

2-3 iterations

Process: Inception and Elaboration

In UP terms and our case studies, imagine we have finished the inception phase and are entering the elaboration phase.

What Happened in Inception?

- The inception phase of the case studies may last only one week. Because this is not the requirements phase of the project, the artifacts created should be brief and incomplete, the phase quick, and the investigation light.
- Inception is a short step to elaboration. It determines basic feasibility, risk, and scope, to decide if the project is worth more serious investigation. Not all activities that could reasonably occur in inception have been covered; this exploration emphasizes requirements - oriented artifacts.

Process: Inception and Elaboration

Some likely activities and artifacts in inception include:

- a short requirements workshop
- most actors, goals, and use cases named
- most use cases written in brief format; 10 - 20% of the use cases are written in fully dressed detail to improve understanding of the scope and complexity
- most influential and risky quality requirements identified
- version one of the Vision and Supplementary Specification written
- risk list For example, leadership really wants a demo at the POS World trade show in Hamburg, in 18 months. But the effort for a demo cannot yet be even roughly estimated until deeper investigation.
- technical proof - of - concept prototypes and other investigations to explore the technical feasibility of special requirements ("Does Java Swing work properly on touch - screen displays?")
- user interface - oriented prototypes to clarify the vision of functional requirements
- recommendations on what components to buy / build / reuse, to be refined in elaboration
  - For example, a recommendation to buy a tax calculation package.
- high - level candidate architecture and components proposed

  This is not a detailed architectural description, and it is not meant to be final or correct. Rather, it is brief speculation to use as a starting point of investigation in elaboration. For example, "A Java client - side application, no application server, Oracle for the database, ..." In elaboration, it may be proven worthy, or discovered to be a poor idea and rejected.
- plan for the first iteration
- candidate tools list

On to Elaboration

Elaboration is the initial series of iterations during which, on a normal project:

- the core, risky software architecture is programmed and tested
- the majority of requirements are discovered and stabilized
- the major risks are mitigated or retired

On to Elaboration

- Elaboration is the initial series of iterations during which the team does serious investigation, implements (programs and tests) the core architecture, clarifies most requirements, and tackles the high - risk issues. In the UP, "risk" includes business value. Therefore, early work may include implementing scenarios that are deemed important, but are not especially technically risky.
- Elaboration often consists of two or more iterations; each iteration is recommended to be between two and six weeks; prefer the shorter versions unless the team size is massive. Each iteration is timeboxed, meaning its end date is fixed.

On to Elaboration

- Elaboration is not a design phase or a phase when the models are fully developed in preparation for implementation in the construction step - that would be an example of superimposing waterfall ideas on iterative development and the UP.
- During this phase, one is not creating throw - away prototypes; rather, the code and design are production - quality portions of the final system. In some UP descriptions, the potentially misunderstood term "architectural prototype"is used to describe the partial system. This is not meant to be a prototype in the sense of a discardable experiment; in the UP, it means a production subset of the final system. More commonly it is called the executable architecture or architectural baseline.

Some key ideas and best practices will manifest in elaboration

- do short timeboxed risk - driven iterations
- start programming early
- adaptively design, implement, and
- test the core and risky parts of the architecture
- test early, often, realistically
- adapt based on feedback from tests, users, developers
- write most of the use cases and other requirements in detail, through a series of workshops, once per elaboration iteration

Artifacts at the beginning of Elaboration

Artifact	Description
Domain Model	**Visual representation of the domain model**
Design Model	**Logical Design Diagrams**
Software Architecture Document	**Summary of key architectural issues and their resolution**
Data Model	**Database schemas, mapping strategies between object and non-object representation**
Use-Case Storyboards, UI Prototypes	**User interface description, usability models...**