

System Analysis and Design

L08. Elaboration Basics

Topics

- Review the inception Phase
- Review the elaboration phase
- Iteration 1—Elaboration Basics



Inception Phase

- Inception is the initial short step to establish a **common vision and basic scope for the project**. It will include
 - analysis of perhaps **10% of the use cases**
 - analysis of the **critical non-functional requirement**
 - creation of a **business case**
 - preparation of the **development environment**
- so that programming can start in the following elaboration phase.

What have done in Inception

- Short requirements workshop
- Most actors, goals, and use cases named
- Most use cases in brief format, and 10-20% in full dress
- Most influential and risky requirements identified
- Version 1 of vision and Supplementary Specification written
- Risk list
- Proof of concept prototypes
- User interface prototypes
- Recommend on buy/build
- High-level candidate architecture
- Plan for first iteration
- Possible tools

复习

What gets defined in Inception?

初始阶段
完成的工作

Artifact	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements. During inception, the names of most use cases will be identified, and perhaps 10% of the use cases will be analyzed in detail.
Supplementary Specification	Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that will have a major impact on the architecture.
Glossary	Key domain terminology, and data dictionary.
Risk List & Risk Management Plan	Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

复习 Requirements Organization in UP Artifacts

用以组织需求no UP工件

- **Use-case model** 用例模型
A set of typical scenarios of using a system.
- **Supplementary specification** 附加说明
Basically, everything not in the use cases.
- **Glossary** 词汇表
the Glossary defines noteworthy terms.
- **Vision** 愿景(设想)
 - Summarizes high-level requirements that are elaborated in the Use-Case Model and Supplementary Specification, and summarizes the business case for the project.
 - A short executive overview document for quickly learning the project's big ideas.
- **Business rules (also called Domain Rules)** 业务规则(领域规则)
Policies that apply to all software projects. E.g. Tax laws, data security, etc.

On to Elaboration Phase 步入细化阶段

- **Elaboration in one sentence:**

Build the **core architecture**, resolve the **high-risk** elements, define **most requirements**, and estimate the **overall schedule and resources**.

- Elaboration is the *initial series of iterations* during project
- Elaboration often consists of *two or more iterations*; each iteration is recommended to be 2~6 weeks
- Elaboration is not a design phase or a phase when the models are fully developed in preparation for implementation. 非设计阶段 (模型没有完整开发)

Some Key Ideas and Best Practices Will Manifest in Elaboration

一些重要的思想和经验

- do short time boxed 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

Elaboration Tasks 细化阶段的任务.

- Core, risky software architecture programmed and tested
 - Executable architecture/Architectural baseline/Architectural prototype to describe the partial system.
 - It is a production subset of the final system.
- Majority of requirements discovered and stabilized
- Major risks mitigated or retired 规避主要风险.
- Estimate overall schedule and resources
- Test early and often
- Adapt based on user feedback 基于客户的反馈调整.

Start These Artifacts in Elaboration

Artifact	Comment
Domain Model	This is a visualization of the domain concepts; it is similar to a static information model of the domain entities.
Design Model	This is the set of diagrams that describes the logical design. This includes software class diagrams, object interaction diagrams, package diagrams, and so forth.
Software Architecture Document	A learning aid that summarizes the key architectural issues and their resolution in the design. It is a summary of the outstanding design ideas and their motivation in the system.
Data Model	This includes the database schemas, and the mapping strategies between object and non-object representations.
Use-Case Storyboards, UI Prototypes	A description of the user interface, paths of navigation, usability models, and so forth.

You Didn't Understand Elaboration

下面这些理论是不对的

- More than a few months long
- It has only one iteration
- Most requirements defined before elaboration
- Risks are not being tackled
- No executable code
- Considered a requirements or design phase
- Full, careful design
- No early testing

第1个迭代

Iteration 1: Elaboration Basics

- Summarizes the iteration-1 requirements
- Understand what's being tackled in this iteration for chosen requirements
- You may need reading the remainder depends on your need or interest in iterative process issues.

POS Iteration 1: Requirements and Emphasis

- 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**, and so forth, would also be done

Iteration 1: Iterative and Incremental Development

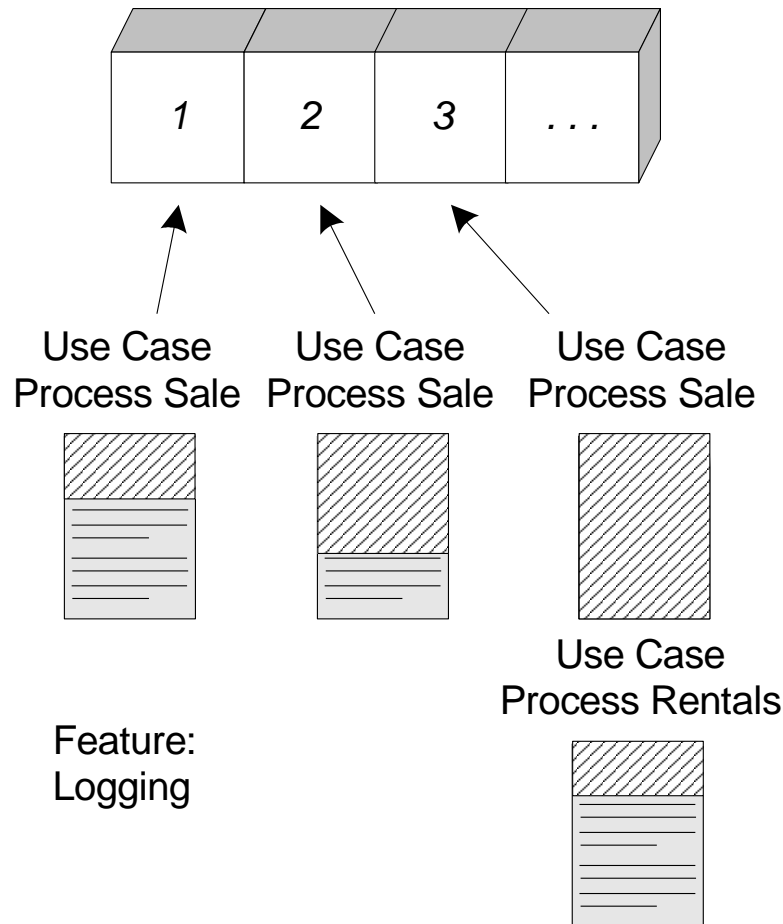
- We Don't Implement All the Requirements at Once
- We start production-quality programming and testing for a subset of the requirements,
- We start that development *before* all the requirements analysis is complete.
- Simple use cases may be completed within one iteration.

Iteration 1:

Incremental Development for the Same Use Case Across Iterations

- Not all requirements in the *chosen* use case are being implemented in iteration-1.
- It is common to work on varying scenarios of the same use case over several iterations
- Gradually extend the system to ultimately handle all the functionality required.

Use Case Implementation May Be Spread Across Iterations



A use case or feature is often too complex to complete in one short iteration.

Therefore, different parts or scenarios must be allocated to different iterations.

将不同的场景分配到不同的迭代去完成
(同一个use case)

Planning the Next Iteration

- Use these criteria to rank requirements (Organize requirements and iterations by ***risk, coverage, and criticality.***):
 - Risk includes technical complexity as well as uncertainty of effort or usability
 - Coverage implies that all major parts of the system are known in early iterations
 - Criticality is functions the client considers of high business value

POS Rank List

Rank	Requirement (Use Case or Feature)	Comment
High	Process Sale Logging ...	Scores high on all rankings. Pervasive. Hard to add late. ...
Medium	Maintain Users ...	Affects security subdomain. ...
Low

选择高优先级的 use case 做为下一个迭代可选内容。