System Analysis and Design

L04. Inception and Evolutionary Requirements

Topics

- Inception
- Evolutionary Requirements

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.

Inception Phase

- In such a short initial inception phase, the following kinds of questions are explored:
  - What is the **vision and business case** for this project?
  - **Feasible**?
  - Buy and/or build?
  - *Rough* unreliable **range of cost**: Is it $10K100K or in the millions?
  - Should we proceed or stop?

Purpose of Inception

- Purpose of Inception is **feasibility investigation**
- *The purpose of the inception phase is not*
  - *to define all the requirements*, or
  - to generate a believable estimate or
  - to generate project plan.

Inception in one sentence…

- Inception in one sentence:
  - Envision the project scope, vision, and business case
- The main problem solved in one sentence:
  - Do the stakeholders have basic agreement on the project and is it worth further investigation?

Activities and Artifacts in Inception

- Some activities and artifacts in inception
  - A short requirements workshop
  - Most actors, goals, and use case 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 risk and quality requirements identified
  - Version one of the Vision and Supplementary Specification written
  - Risk list
  - Technical proof-of-concept prototypes and other investigations to explore the technical feasibility of special requirements (e.g. Does Java Swing work properly on touch-screen displays?)

Activities and Artifacts in Inception

- Some activities and artifacts in inception
  - User interface-oriented prototype to clarify the vision of functional requirements
  - Recommendations on what components to buy/build/reuse, to be refined in elaboration (a tax calculation package)
  - High-level candidate architecture and components proposed
  - Plan for the first iteration
  - Candidate tools list

Sample Inception Artifacts (What gets defined?)

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.

How long does it take?

Inception phase should be relatively short for most projects

- Can be as short as a day or two, as long as a few weeks
- On many projects, if it is more than a week long, then the point of inception has been missed: It is to decide if the project is worth a serious investigation

You didn’t get it when…

- It takes more than a few weeks
- You attempt to define most of the requirements
- Estimates are expected to be reliable
- You define the architecture
- There is no business case or vision
- All of the use cases were written
- No use cases were written

How Much UML During Inception?

- Purpose of Inception is feasibility investigation
- Beyond simple UML use case diagrams, not much diagramming is warranted.
- There is more focus in inception on understanding the basic scope and 10% of the requirements, expressed mostly in text forms.
- In practice, most UML diagramming will occur in the next phase elaboration

Chapter 5

Evolutionary Requirements

Requirements

- **Requirement:** Capabilities and conditions to which the system (and more broadly the project) must conform
- **Requirement Management (UP):**
  - not the waterfall attitude of attempting to fully define and stabilize the requirements in the first phase
  - But a systematic approach to finding, documenting, organizing the changing requirements of a system.
  - In short, doing it iteratively and skillfully, and not being sloppy.

Evolutionary vs. Waterfall Requirements

- 25% of the requirements change on a project
- Attempting to define all requirements was the main problem in 82% of projects Waterfall
- When attempting to specify all features in advance, 45% of such features were never used Waterfall.
- The UP embraces change in requirements as a fundamental driver on projects. UP
- Do **iterative and evolutionary requirements analysis** combined with early timeboxed iterative development and frequent stakeholder participation, evaluation, and feedback on partial results.

Skillful Means to Find Requirements

找出需求的技巧

- The UP encourages skillful elicitation via techniques such as

探求需求的技术.

  - writing use cases with customers,
  - requirements workshops that include both developers and customers,
  - focus groups with proxy customers, and
  - a demo of the results of each iteration to the customers, to solicit feedback.
- The UP welcomes any requirements elicitation method that can add value and that increases user participation.

可以采用任何的探求方法

Types and Categories of Requirements

- In common usage, requirements are categorized as **functional** (behavioral) or **non-functional** (everything else).
- In the UP, requirements are categorized according to the **FURPS+ model**.
  - Some of these requirements are collectively called the **quality attributes, quality requirements**, or the "-ilities" of a system.
  - These include **usability**, **reliability**, **performance**, and **supportability**.
  - The quality attributes have a strong influence on the architecture of a system.

FURPS+ model

- Functional
- Usability – human factors
- Reliability – failure rate and recovery
- Performance – response time throughput, etc.
- Supportability – Adaptability, maintainability, internationalization, configurability

The “+”: ancillary and sub-factors

- Implementation – resource limitations, languages and tools, hardware
- Interface – constraints imposed by interfacing with external systems
- Operations – System management in the operational setting
- Packaging – mostly for external-use systems
- Legal – licensing, copyright, etc.

Requirements Organization in UP Artifacts

- **Use-case model**
  A set of typical scenarios of using a system.
- **Supplementary specification**
  Basically, everything not in the use cases.
- **Glossary**
  the Glossary defines noteworthy terms.
- **Vision**
  - 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)**
  Policies that apply to all software projects. E.g. Tax laws, data security, etc.

Example vision & business case

Introduction Vision 设想的目标

We envision a next generation fault-tolerant point-of-sale (POS) application, NextGen POS, with the flexibility to support varying customer business rules, multiple terminal and user interface mechanisms, and integration with multiple third-party supporting systems.

Business Case 业务案例（现在系统不满足要求的各种情况）

Existing POS products are not adaptable to the customer's business, in terms of varying business rules and varying network designs (for example, thin client or not; 2, 3, or 4-tier architectures). In addition, they do not scale well as terminals and business increase. None can work in either on-line or off-line mode, dynamically adapting depending on failures. None easily integrate with many third-party systems. None allow for new terminal technologies such as mobile PDAs. There is marketplace dissatisfaction with **this inflexible state of affairs**, and demand for a POS that rectifies this.

用例的例子.

Brief format Use Case

A use case is a story of an actor using a system to meet a goal.

Process Sale: 处理销售的用例

A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item.
The system presents a running total and line-item details. The customer enters payment information, which the system validates and records.
The system updates inventory.
The customer receives a receipt from the system and then leaves with the items.

Glossary

Term	Definition	Format	Aliases
item	A product or service for sale		
Payment authorization	Validation by an external payment authorization service that they will make or guarantee the payment to the seller.		
UPC	Numeric code that identifies a product. Usually symbolized with a bar code placed on products.	12-digit code of several subparts	Universal Product Code