

Object-Oriented Software Analysis and Design

School of Computer Science
University of Windsor

Inception is Not the Requirements Phase

Inception is Not the Requirements 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, and
 - ▶ preparation of the development environment

so that programming can start in the following elaboration phase.

What is Inception?

Most projects require a **short initial step** in which 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?

What is Inception? (contd.)

- ▶ The **purpose of the inception phase** is not to define all the requirements, or generate a believable estimate or project plan.
- ▶ Inception, is **not the time do all requirements or create believable estimates or plans**. That happens during elaboration.
- ▶ Most requirements analysis occurs during the elaboration phase, in parallel with early production-quality programming and testing.

How Long is Inception?

- ▶ The inception phase will be especially **brief** if it has been decided beforehand that **the project will definitely be done, and it is clearly feasible**.
- ▶ It may include the first requirements workshop, planning for the first iteration, and then quickly moving forward to elaboration.

Sample inception artifacts

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 have 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.

Isn't That a Lot of Documentation?

- ▶ Recall that artifacts should be considered **optional**. Choose to create only those that really add value for the project, and drop them if their worth is not proved.
- ▶ And since this is evolutionary development, the point is **not** to create complete specifications during this phase, but initial, rough documents, that are refined during the elaboration iterations, in response to the invaluable feedback from early programming and testing.

You Know You Didn't Understand Inception When...

- ▶ It is more than “a few” weeks long for most projects.
- ▶ There is an attempt to define most of the requirements.
- ▶ Estimates or plans are expected to be reliable.
- ▶ You define the architecture (this should be done iteratively in elaboration).
- ▶ You believe that the proper sequence of work should be: 1) define the requirements; 2) design the architecture; 3) implement.
- ▶ There is no Business Case or Vision artifact.
- ▶ All the use cases were written in detail.

How Much UML During Inception?

- ▶ The **purpose of inception** is to collect just enough information to establish a common vision, decide if moving forward is feasible, and if the project is worth serious investigation in the elaboration phase.
- ▶ As such, perhaps 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.

Evolutionary Requirements

Evolutionary Requirements

- ▶ Requirements are capabilities and conditions to which the system and more broadly, the project must conform.
- ▶ One of the **best practices** of UP is *manage requirements*.
 - ▶ which means
 - ... a systematic approach to finding, documenting, organizing, and tracking the changing requirements of a system. [RUP]
- ▶ In short, doing it iteratively and skillfully, and not being sloppy.
- ▶ A **prime challenge** of requirements analysis is to find, communicate, and remember (that usually means write down) what is really needed, in a form that clearly speaks to the client and development team members.

Evolutionary vs. Waterfall Requirements

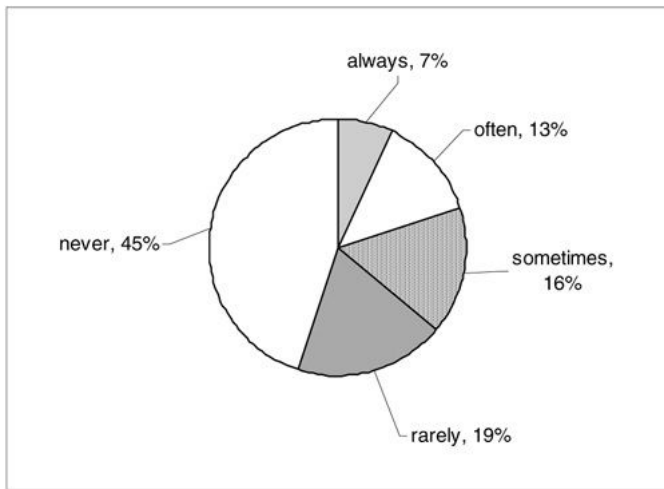


Figure: *Actual use of waterfall-specified features.*

What are the Types and Categories of Requirements?

In the UP, requirements are categorized according to the FURPS+ model

- ▶ **Functional**—features, capabilities, security.
- ▶ **Usability**—human factors, help, documentation.
- ▶ **Reliability**—frequency of failure, recoverability, predictability.
- ▶ **Performance**—response times, throughput, accuracy, availability, resource usage.
- ▶ **Supportability**—adaptability, maintainability, internationalization, configurability.

Types and Categories of Requirements in UP (contd.)

The “+” in FURPS+ indicates ancillary and sub-factors, such as:

- ▶ **Implementation**—resource limitations, languages and tools, hardware, . . .
- ▶ **Interface**—constraints imposed by interfacing with external systems.
- ▶ **Operations**—system management in its operational setting.
- ▶ **Packaging**—for example, a physical box.
- ▶ **Legal**—licensing and so forth.

Types and Categories of Requirements in UP (contd.)

- ▶ Use FURPS+ model as a checklist for requirements coverage, in order to reduce the risk of not considering some important facet of the system.
- ▶ A more general categorization of requirements:
 - ▶ Functional requirements
 - ▶ Non-functional requirements

Functional and Non-functional Requirements

- ▶ Functional requirements
 - ▶ These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
 - ▶ In some cases, the functional requirements may also explicitly state what the system should not do.
- ▶ Non-functional requirements
 - ▶ These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards.
 - ▶ Non-functional requirements often apply to the system as a whole rather than individual system features or services.

How are Requirements Organized in UP Artifacts?

The UP offers several requirements artifacts. As with all UP artifacts, they are optional. Key ones include:

- ▶ **Use-Case Model:** set of typical scenarios of using a system. CAPTURE functional (behavioral) requirements.
- ▶ **Supplementary Specification:** Basically, everything not in the use cases. CAPTURE all non-functional requirements, such as performance or licensing.
- ▶ **Glossary:** Glossary defines noteworthy terms. The data dictionary
- ▶ **Vision:** Summarizes high-level requirements that are elaborated in the Use-Case Model and Supplementary Specification, and summarizes the business case for the project.
- ▶ **Business Rules:** Business rules (also called Domain Rules) typically describe requirements or policies that transcend one software project - they are required in the domain or business, and many applications may need to conform to them. An excellent example is government tax laws.

What is the Correct Format for these Artifacts?

- ▶ In the UP, all artifacts are information abstractions; they could be stored on **Web pages** (such as in a Wiki Web), **wall posters**, or any variation imaginable.
- ▶ The online RUP documentation product contains templates for the artifacts, but these are an optional aid, and can be ignored.

It's Quiz Time

1. Inception is the initial short step to establish a common vision and basic scope for the project. (True or False)
2. The “+” in FURPS+ indicates ancillary and sub-factors, such as: (Choose the one correct answer)
 - 2.1 Implementation, interface, operations, packaging, and legal
 - 2.2 Performance, usability, operations, packaging, and legal
 - 2.3 Implementation, performance, reliability, packaging, and legal
 - 2.4 Implementation, supportability, functional, packaging, and usability
3. Non-functional requirements state the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. (True or False)