

Other Requirements

COMP-3831

Larman: Chapter 7

Other Requirement Artifacts

- Vision
- Supplementary Specifications
- Glossary
- Business Rules

Vision

- The vision serves to communicate the big ideas regarding:
 - Why the project was proposed?
 - What the problems are?
 - Who the stakeholders are?
 - What they need?
 - What the proposed solution looks like?
- Vision provides the contractual basis for the more detailed technical requirements.

System Features

The Vision document should include a list of System Features

- An externally observable service provided by the system which directly fulfills a stakeholder need
- They should pass the linguistic test:

The system shall do xxxxxx

Supplementary Specifications

Document requirements not captured in Use Cases.

- FURPS+ requirements - functionality, usability, reliability, performance, and supportability
- Reports
- Hardware and software constraints
- Development constraints
- Other design and implementation constraints
- Internationalization concerns
- Licensing and other legal issues
- Packaging
- Standards (technical, safety, quality)
- Physical environment (e.g. heat or vibration)
- Operational concerns (e.g. error handling or backups)
- Domain or business rules
- Any other domain-related information

Supplementary Specifications - Constraints

- Constraints are not behaviors, but some other kind of restriction on the design or project.
- Restrictive in nature
- Early constraints are always a bad idea

E.g.

❖ Must use Oracle

❖ Must run on Linux

Supplementary Specifications - Quality Attributes

- Quality of the system other than functionality
- Usability, reliability, etc ...
- Two types:
 - Observable at execution (usability, reliability, performance, etc ...)
 - Not observable at execution (supportability, testability, ...)

Supplementary Specs - Domain (Business) rules

- Rules that dictate how a domain or business may operate
 - E.g. company policies, physical laws, government laws, etc

Glossary

- Also plays the role of a data dictionary
- Terms attributes could include:
 - Aliases
 - Description
 - Format (type, length, unit)
 - Relationship to other elements
 - Range of values
 - Validation rules
- Glossary not only for “atomic” terms (product price), but also for “composite” terms (payment authorization request).

Suggested sequence of Inception phase artifacts

1. Write a brief draft of the vision.
2. Start the Glossary.
3. Identify user goals and the supporting use cases.
4. Write some use cases and start the Supplementary Specification.
5. Refine the Vision, summarizing information from the previous documents.

Overview of activities in inception

- Decide if moving forward is feasible
- Collect just enough information to establish a common vision
- Decide if project is worth serious investigation in the elaboration phase
- Simple UML use case diagrams
- Understand basic scope and 10% of requirements

Questions

Elaboration

Iteration 1 - Basics

Larman: Chapter 8

What happened in inception

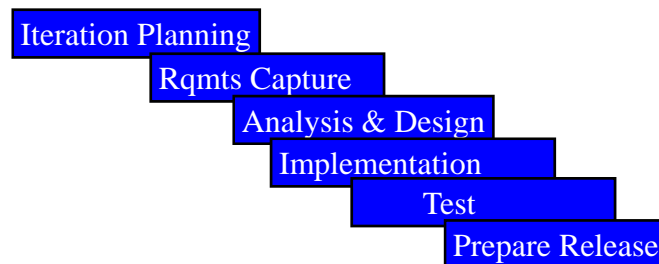
- A short requirements workshop
- Most actors, goals, and use cases named
- 10-20% of use cases written in fully dressed format
- Most influential and risky quality requirements identified
- Version 1.0 of the Vision and Supplementary Specifications written
- Risk list (demo for Comdex show)
- Proof-of-concept prototypes
- UI prototypes
- Recommendations on buy / build / reuse
- High-level architecture and proposed components
- Plan for first elaboration iteration
- Candidate tools list

Use Cases Drive the Iteration Process

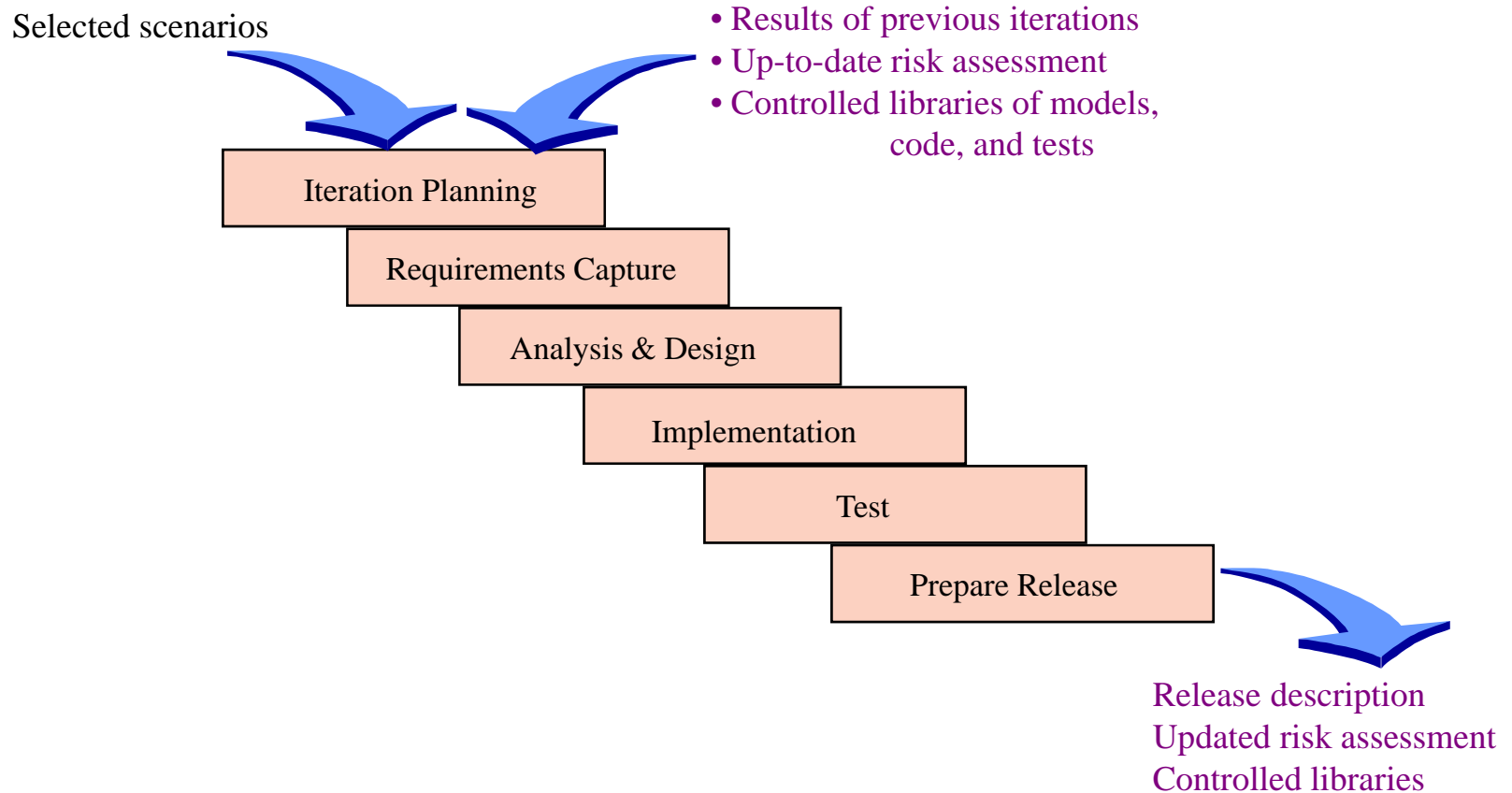
Inception → Elaboration → Construction → Transition

Iteration 1 → Iteration 2 → Iteration 3

“Mini-Waterfall” Process



The Iteration Life Cycle: A Mini-Waterfall



Elaboration phase

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

Elaboration (continued ...)

Initial series of iterations during which:

- Carry out serious investigation
- Implement core architecture
- The majority of requirements are discovered and stabilized
- Major risks are mitigated or retired
- Core architectural elements are implemented and proven

Elaboration (continued ...)

- Elaboration usually 2-4 iterations.
 - Each iteration typically between 2-6 weeks depending on team size.
 - Each iteration is time-boxed (i.e. end date is fixed)
 - If team is unable to meet date, requirement is moved to next iteration
- During this phase, prototypes are not throw-away

Elaboration - best practices

- 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, and realistically
- Adapt based on feedback from tests, users, and developers
- Write most of the use cases and other requirements in detail, through a series of workshops, once per elaboration iteration

Planning the next iteration

Organize requirements and iterations by:

- **Risk:** technical complexity, uncertainty of effort, ...
- **Coverage:** ensure all major parts of the system are at least touched on in early iterations
- **Criticality:** functions of high business value

Artifacts that may start in Elaboration

Artifact	Comment
Domain Model	visualization of domain concepts; similar to static information model of domain entities
Design Model	Set of diagrams that describe logical design (class diagrams, object interaction diagrams, package diagrams, etc...)
Software architecture document	key architectural issues and their resolution in the design
Data Model	database schemas
Test Model	what will be tested and how
Implementation Model	source code, executables, database, etc...
Use-case storyboards UI Prototypes	UI, paths of navigation, usability models, etc ...

Questions and Conclusions