



Hugbúnaðarverkefni 1 / Software Project 1

3. Project Inception

HBV501G – Fall 2018

Matthias Book



HÁSKÓLI ÍSLANDS
VERKFRÆÐI- OG NÁTTÚRUVÍSINDASVIÐ
IÐNAÐARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

Team Consultations (Final Schedule)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	X			X			X			X			X							X			X			X				
D		X			X			X			X					X							X				X		X	
M			X			X			X			X			X				X						X			X		X
#								2						0			0	0			0	0								

■ Consultation time slots

- Thursday 08:30-11:30, V-152
 - Teams 1-10: 08:30-09:30
 - Teams 11-20: 09:30-10:30
 - Teams 21-30: 10:30-11:30

■ Seating arrangement

- Andri's teams: last rows
- Daníel's teams: middle rows
- Matthias' teams: first rows

In-Class Quiz Prep

- Please prepare a scrap of paper with the following information:

■ ID: _____@hi.is Date: _____

■ a) _____ e) _____

■ b) _____ f) _____

■ c) _____ g) _____

■ d) _____ h) _____

- During class, I'll show you questions that you can answer with numbers
 - No further elaboration necessary
- Hand in your scrap at end of class
- All questions in a quiz weigh same
- All quizzes (ca. 10 throughout semester) have the same weight
 - Your worst 2 quizzes will be disregarded
- Overall quiz grade counts as optional question worth 7.5% on final exam



Quiz #1 Solution: Unified Process Phases

1. Inception

- Approximate vision
- Business case (b)
- Scope
- Vague estimates (h)

Not a “requirements phase”, but feasibility analysis after which we can decide whether it makes sense to proceed with project

2. Elaboration

- Refined vision (f)
- Identification of most requirements (c)
- More realistic estimates
- Iterative implementation of core architecture (d)
- Resolution of high risks (g)

Not a “design phase”, but iterative development of core architecture and mitigation of high risks

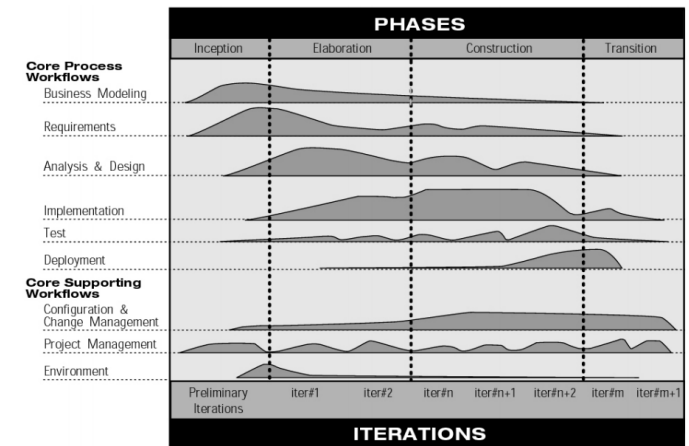
3. Construction

- Iterative implementation of lower-risk and other elements (e)
- Preparation for deployment

4. Transition

- Beta tests (a)
- Deployment

Not an “implementation phase”, but incremental design and development of sets of features



Recap: An Important Distinction

- **During Inception and Elaboration**, the project still is a relatively light-and-fast, low-risk operation.
 - If the project fails to reach one of these milestones,
 - it may be aborted due to infeasibility
 - or it needs to be considerably re-thought.
- **During Construction and Transition**, the project becomes a high-cost, high-risk operation with substantial organizational inertia.
 - If the project fails to reach one of these milestones,
 - deployment may have to be postponed by at least one release.
- Note: The UP is not the *cause* for the high complexity. Rather, it provides the *structure* that makes such highly complex projects manageable at all.

Construction: Primary Objectives

- Analyze, design, develop and test all required **functionality**.
- Achieve useful **versions** (alpha, beta, other test releases) of adequate quality.
- Minimize cost by optimizing **resources**, avoiding unnecessary scrap & rework.
- Achieve some degree of **parallelism** in the work of development teams.
 - Even on smaller projects, there are typically components that can be developed independently of one another, allowing for natural parallelism between teams (resources permitting).
 - Parallelism can accelerate the development activities significantly; but it also increases the complexity of resource management and workflow synchronization.
 - A robust architecture is essential if any significant parallelism is to be achieved.
- Decide if the software, sites, and users are all **ready** for application deployment.

Construction: Essntl. Activities

- Complete component **development and testing** against the defined evaluation criteria.
- **Resource management**, control and process optimization.
- **Assessment** of product releases against acceptance criteria for the vision.

Project Management

Manage Iteration
Monitor and Control Project
Reevaluate Project Scope and Risk
Plan for Next Iteration
Refine Software Development Plan

Requirements

Manage Changing Requirements

Analysis and Design

Design Components
Design the Database
Refine the Architecture

Implementation

Plan the Integration
Implement Components
Integrate Each Subsystem
Integrate the System

Test

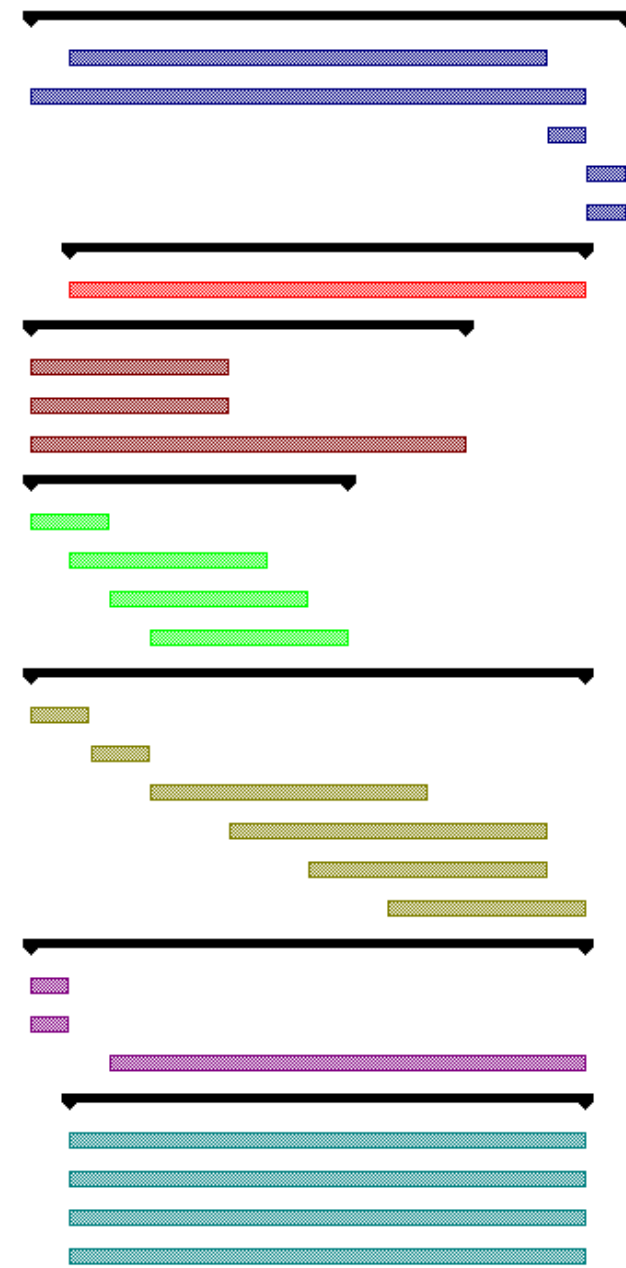
Define Evaluation Mission
Verify Test Approach
Validate Build Stability
Test and Evaluate
Achieve Acceptable Mission
Improve Test Assets

Environment

Prepare Environment for an Iteration
Prepare Guidelines for an Iteration
Support Environment During an Iteration

Configuration & Change Management

Change and Deliver Configuration Items
Manage Baselines and Releases
Manage Change Requests
Monitor and Report Configuration Status



Construction: “Initial Operational Capability” Milestone

“Ensure that it’s ready to beta-test.”

Evaluation Criteria

- Is this **product release stable and mature** enough to be deployed in the user community?
- Are all the **stakeholders ready for the transition** into the user community?
- Are actual resource **expenditures** versus planned still **acceptable**?

Artifacts

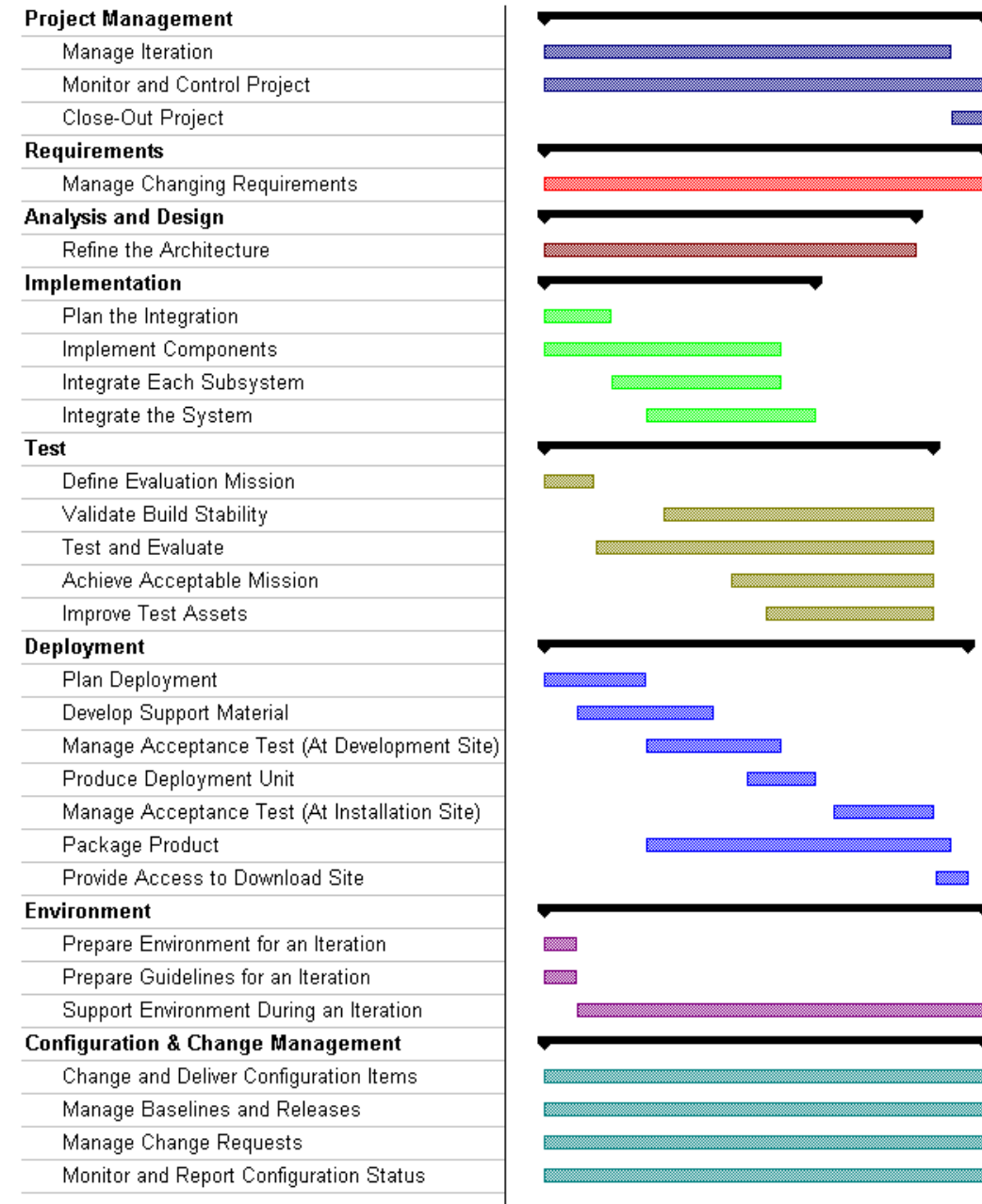
- The System
- Deployment Plan
- Implementation Model
- Test Suite
- Training Materials
- Iteration Plan
- Design Model
- Development Case
- Tools
- Data Model
- Project-specific Templates
- Supplementary Specifications
- Use Case Model

Transition: Primary Objectives

- **Beta-testing**
 - Validate the new system against user expectations
 - Parallel operation relative to a legacy system that shall be replaced
- **Tuning** activities such as bug-fixing, enhancement for performance and usability
- Achieving **user self-supportability**
- Assessment of the deployment baselines against the complete vision and the **acceptance criteria** for the product; achieving stakeholder concurrence that
 - deployment baselines are complete
 - deployment baselines are consistent with the evaluation criteria of the vision
- Executing **deployment** plans

Transition: Essential Activities

- **Deployment-specific engineering**
 - such as commercial packaging and production
- **Converting operational databases**
- **Testing** deliverable product at deployment site
- Finalizing end-user **support material**
- Creating a product **release**
- **Training** of users and maintainers
- Getting user **feedback**
- **Fine-tuning** the product based on feedback
- **Rollout** to marketing, distribution, sales forces



Transition: “Product Release” Milestone

“Ensure it’s ready for production use.”

Evaluation Criteria

- Is the **user satisfied**?
- Are actual resources **expenditures** versus planned expenditures **acceptable**?

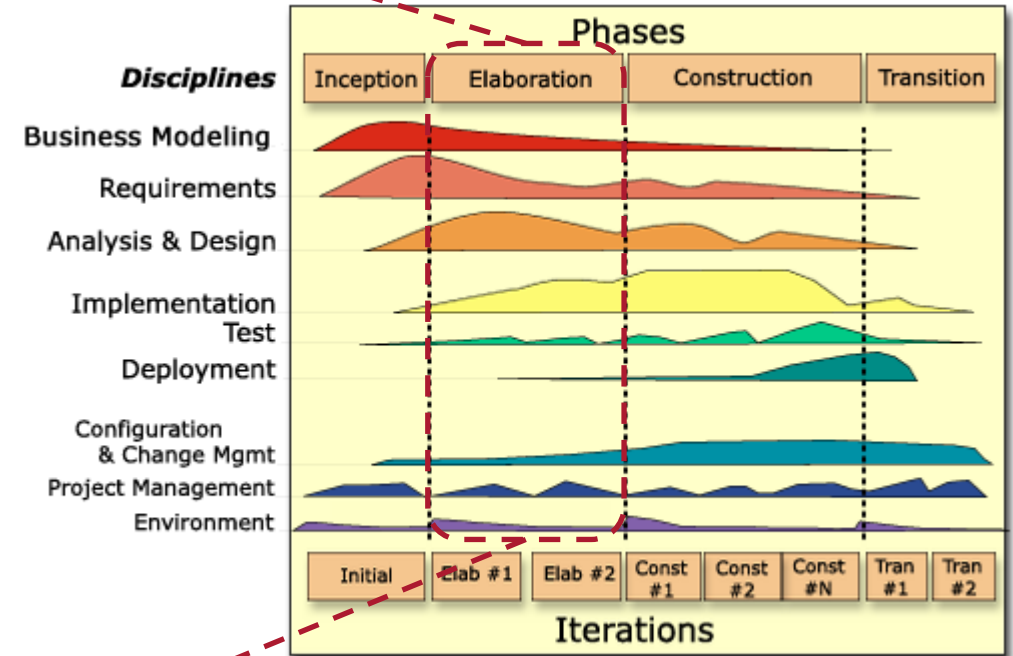
Artifacts

- Product Build
- Release Notes
- Installation Artifacts
- Training Material
- End-User Support Material
- Test Suite
- Product Packaging

Example: Definition of Activities and Artifacts in the RUP



Phase: Elaboration

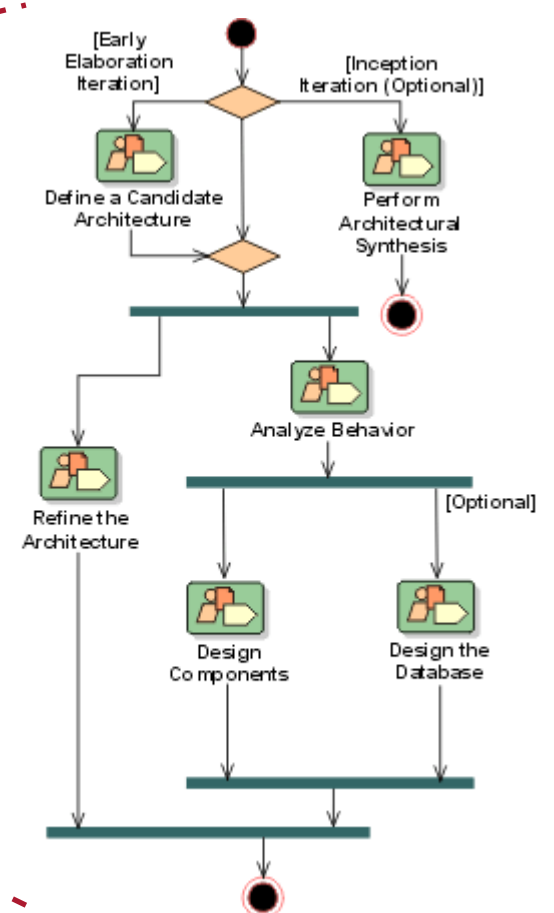


Rational Unified Process

Example: Definition of Activities and Artifacts in the RUP

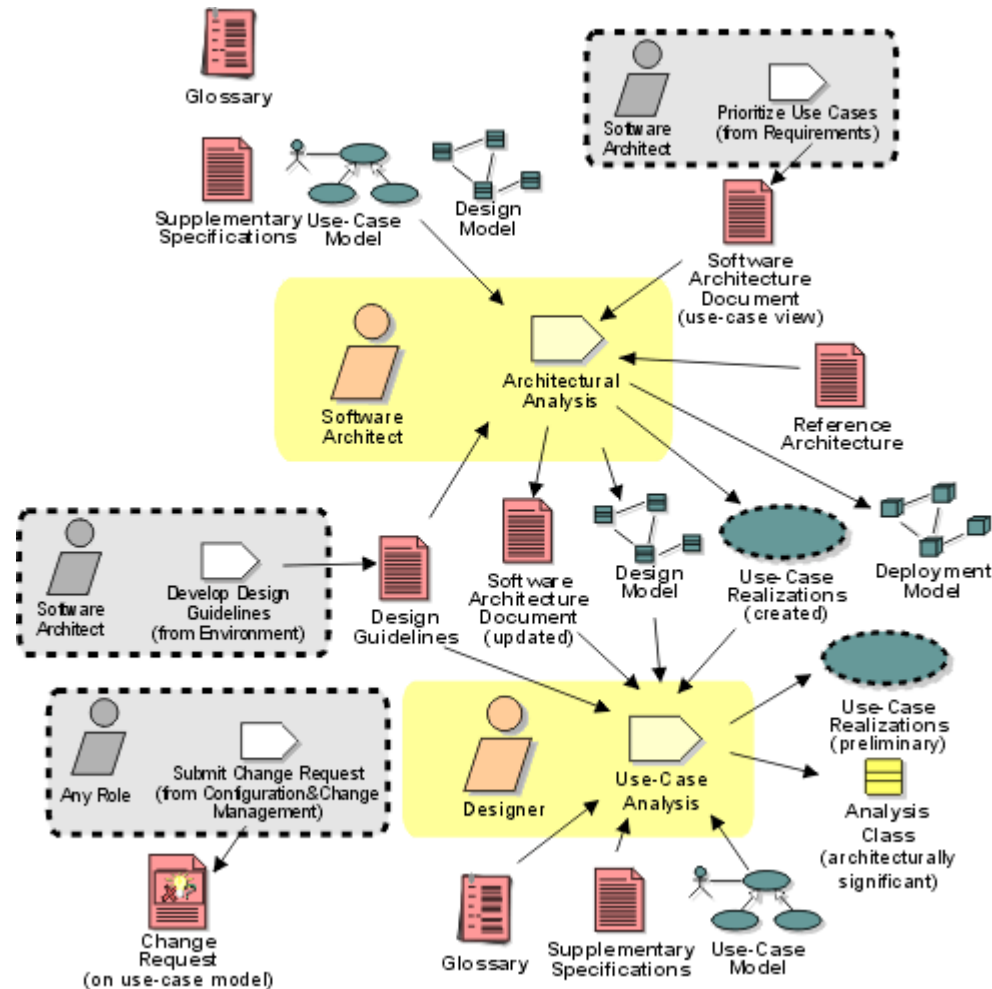


Phase: Elaboration

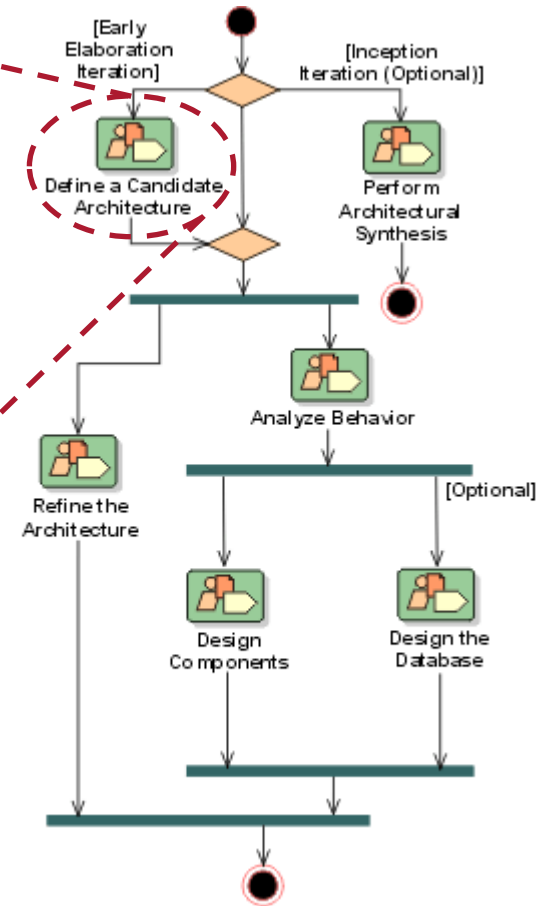


Discipline Workflow:
Analysis & Design

Example: Definition of Activities and Artifacts in the RUP

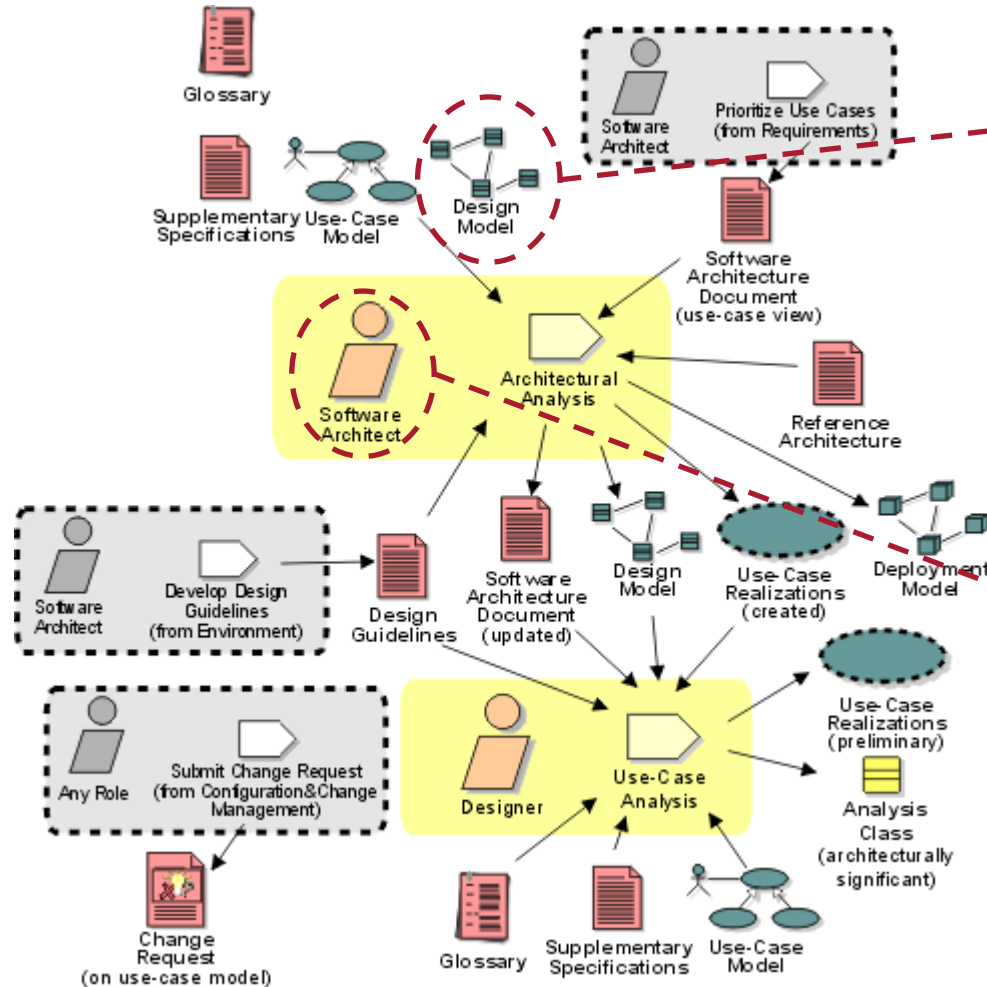


Workflow Detail:
Define a Candidate Architecture



Discipline Workflow:
Analysis & Design

Example: Definition of Activities and Artifacts in the RUP



Workflow Detail:
Define a Candidate Architecture

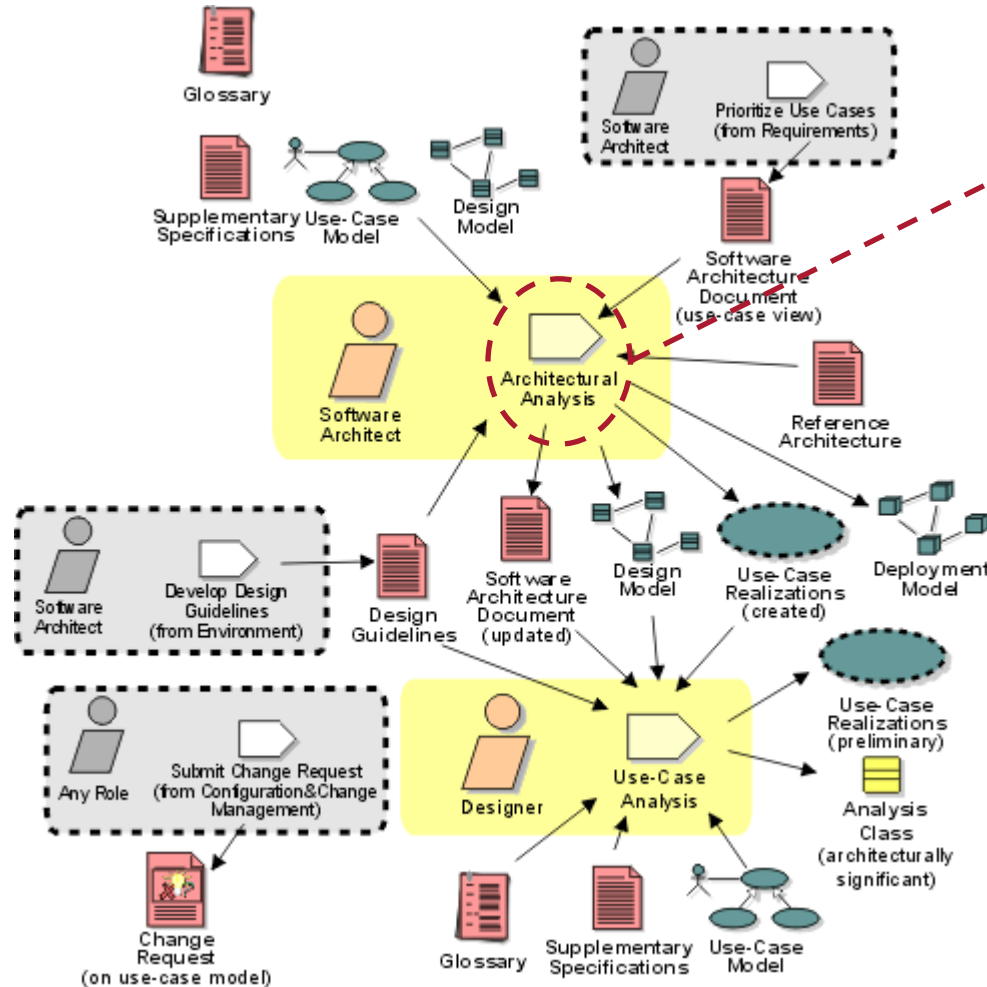
■ Artifact: Design Model

- UML Class Diagram
- UML Sequence Diagram
- UML Collaboration Diagram
- UML State Diagram
- ...

■ Role: Software Architect

- Responsibilities
- Staffing Guidelines

Example: Definition of Activities and Artifacts in the RUP

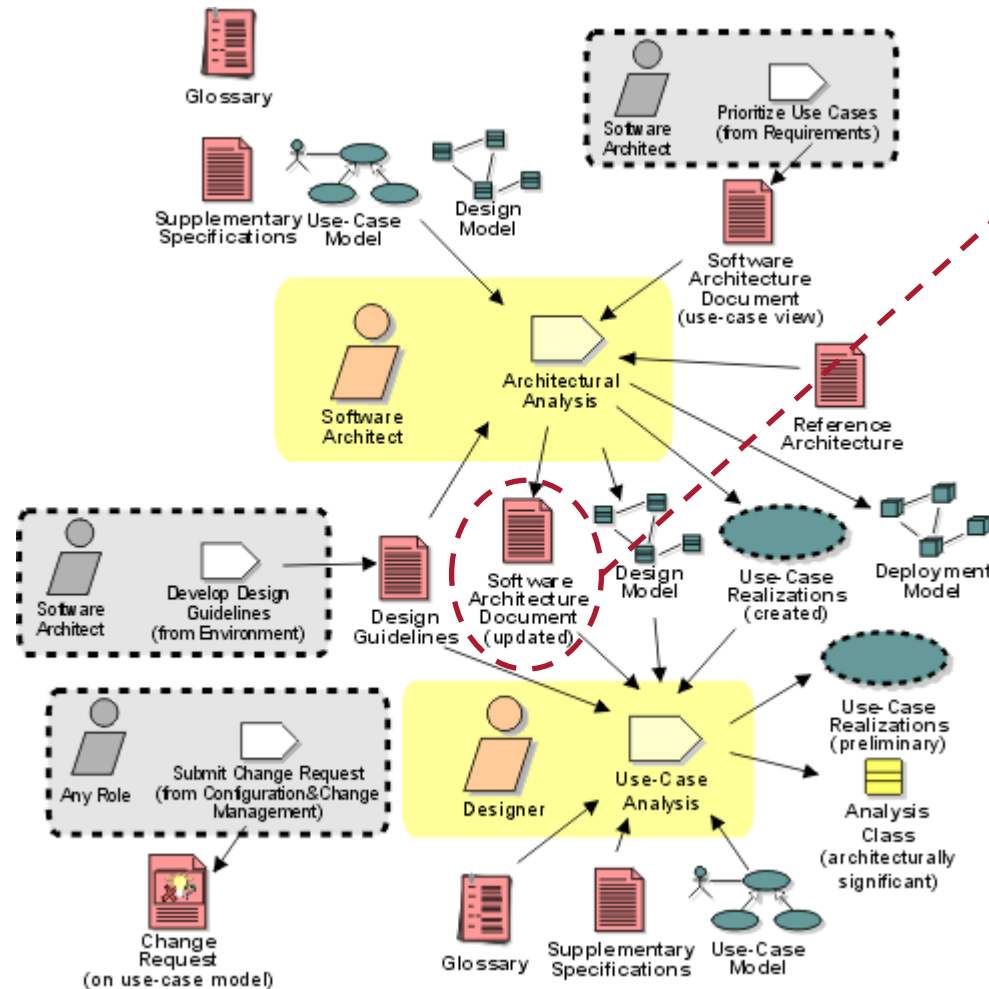


Workflow Detail:
Define a Candidate Architecture

■ **Activity:** Architectural Analysis

- Develop architecture overview
- Survey available assets
- Define high-level subsystem organization
- Identify key abstraction
- Develop high-level deployment model
- Identify analysis mechanisms
- Create use-case realizations
- Identify stereotypical interactions
- Review the results

Example: Definition of Activities and Artifacts in the RUP



Workflow Detail:
Define a Candidate Architecture

■ **Artifact: Software Architecture Doc.**

- Introduction
- Architectural Representation
- Architectural Goals and Constraints
- Use-Case View
- Logical View
- Process View
- Deployment View
- Implementation View
- Data View
- Size, Performance
- Quality

Table of Contents	
1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 References	1
2. Architectural Representation	2
2.1 Architectural Goals and Constraints	2
2.2 Use-Case View	2
2.3 Logical View	2
2.4 Process View	2
2.5 Deployment View	2
2.6 Implementation View	2
2.7 Data View	2
2.8 Size and Performance	2
2.9 Quality	2
3. Architectural Goals and Constraints	3
3.1 Architectural Goals	3
3.2 Architectural Constraints	3
4. Use-Case View	4
4.1 Use-Case Diagram	4
4.2 Use-Case Description	4
4.3 Use-Case Realization	4
4.4 Use-Case Realization Diagram	4
4.5 Use-Case Realization Description	4
4.6 Use-Case Realization Diagram	4
4.7 Use-Case Realization Description	4
4.8 Use-Case Realization Diagram	4
4.9 Use-Case Realization Description	4
4.10 Use-Case Realization Diagram	4
4.11 Use-Case Realization Description	4
4.12 Use-Case Realization Diagram	4
4.13 Use-Case Realization Description	4
4.14 Use-Case Realization Diagram	4
4.15 Use-Case Realization Description	4
4.16 Use-Case Realization Diagram	4
4.17 Use-Case Realization Description	4
4.18 Use-Case Realization Diagram	4
4.19 Use-Case Realization Description	4
4.20 Use-Case Realization Diagram	4
4.21 Use-Case Realization Description	4
4.22 Use-Case Realization Diagram	4
4.23 Use-Case Realization Description	4
4.24 Use-Case Realization Diagram	4
4.25 Use-Case Realization Description	4
4.26 Use-Case Realization Diagram	4
4.27 Use-Case Realization Description	4
4.28 Use-Case Realization Diagram	4
4.29 Use-Case Realization Description	4
4.30 Use-Case Realization Diagram	4
4.31 Use-Case Realization Description	4
4.32 Use-Case Realization Diagram	4
4.33 Use-Case Realization Description	4
4.34 Use-Case Realization Diagram	4
4.35 Use-Case Realization Description	4
4.36 Use-Case Realization Diagram	4
4.37 Use-Case Realization Description	4
4.38 Use-Case Realization Diagram	4
4.39 Use-Case Realization Description	4
4.40 Use-Case Realization Diagram	4
4.41 Use-Case Realization Description	4
4.42 Use-Case Realization Diagram	4
4.43 Use-Case Realization Description	4
4.44 Use-Case Realization Diagram	4
4.45 Use-Case Realization Description	4
4.46 Use-Case Realization Diagram	4
4.47 Use-Case Realization Description	4
4.48 Use-Case Realization Diagram	4
4.49 Use-Case Realization Description	4
4.50 Use-Case Realization Diagram	4
4.51 Use-Case Realization Description	4
4.52 Use-Case Realization Diagram	4
4.53 Use-Case Realization Description	4
4.54 Use-Case Realization Diagram	4
4.55 Use-Case Realization Description	4
4.56 Use-Case Realization Diagram	4
4.57 Use-Case Realization Description	4
4.58 Use-Case Realization Diagram	4
4.59 Use-Case Realization Description	4
4.60 Use-Case Realization Diagram	4
4.61 Use-Case Realization Description	4
4.62 Use-Case Realization Diagram	4
4.63 Use-Case Realization Description	4
4.64 Use-Case Realization Diagram	4
4.65 Use-Case Realization Description	4
4.66 Use-Case Realization Diagram	4
4.67 Use-Case Realization Description	4
4.68 Use-Case Realization Diagram	4
4.69 Use-Case Realization Description	4
4.70 Use-Case Realization Diagram	4
4.71 Use-Case Realization Description	4
4.72 Use-Case Realization Diagram	4
4.73 Use-Case Realization Description	4
4.74 Use-Case Realization Diagram	4
4.75 Use-Case Realization Description	4
4.76 Use-Case Realization Diagram	4
4.77 Use-Case Realization Description	4
4.78 Use-Case Realization Diagram	4
4.79 Use-Case Realization Description	4
4.80 Use-Case Realization Diagram	4
4.81 Use-Case Realization Description	4
4.82 Use-Case Realization Diagram	4
4.83 Use-Case Realization Description	4
4.84 Use-Case Realization Diagram	4
4.85 Use-Case Realization Description	4
4.86 Use-Case Realization Diagram	4
4.87 Use-Case Realization Description	4
4.88 Use-Case Realization Diagram	4
4.89 Use-Case Realization Description	4
4.90 Use-Case Realization Diagram	4
4.91 Use-Case Realization Description	4
4.92 Use-Case Realization Diagram	4
4.93 Use-Case Realization Description	4
4.94 Use-Case Realization Diagram	4
4.95 Use-Case Realization Description	4
4.96 Use-Case Realization Diagram	4
4.97 Use-Case Realization Description	4
4.98 Use-Case Realization Diagram	4
4.99 Use-Case Realization Description	4
4.100 Use-Case Realization Diagram	4

Tailoring the Unified Process

- Remember: Most elements of the UP are optional – tailor to project's needs!

For a lightweight project:

- Use only a small set of UP activities and artifacts
 - Avoid creating them unless they add value
- Don't try to complete all requirements and designs before implementation
 - In an iterative process, requirements will emerge based on feedback on increments
- Use the UML for visual modeling
 - Use simple notation and focus on those parts of the design that are not yet well understood
- Have a general Phase Plan covering just the milestones, and have a detailed Iteration Plan covering one iteration in advance
 - Detailed planning is done adaptively from iteration to iteration

Key Practices of the Unified Process

- Perform iterative-incremental development with short time-boxes
- Tackle high-risk and high-value issues in early iterations
- Continuously engage users for evaluation, feedback, and requirements
- **Build a cohesive core architecture in early iterations**
- Continuously verify quality, test early, often, and realistically
- **Apply use cases where appropriate**
- **Model visually with the UML**
- Carefully manage requirements
- Practice change request and configuration management
- **Avoid superimposing a waterfall model over the Unified Process!**

Inception in a Nutshell

- Goal: Envision the product vision, scope, and business case
- **Do the stakeholders have basic agreement on the vision of the project, and is it worth investing in serious investigation?**
- It's not Inception if...
 - it takes more than very few weeks
 - there is an attempt to define most requirements
 - the architecture is being set in stone
 - there is no Vision and Scope document
 - estimates or plans are expected to be reliable
- Most requirements analysis will occur during Elaboration, in parallel with early production-quality programming and testing.

Vision and Scope

see also:

- Larman: Applying UML and Patterns, Ch. 4
- Wiegers, Beatty: Software Requirements, Ch. 5



Product Vision and Project Scope

- Based on **business requirements**

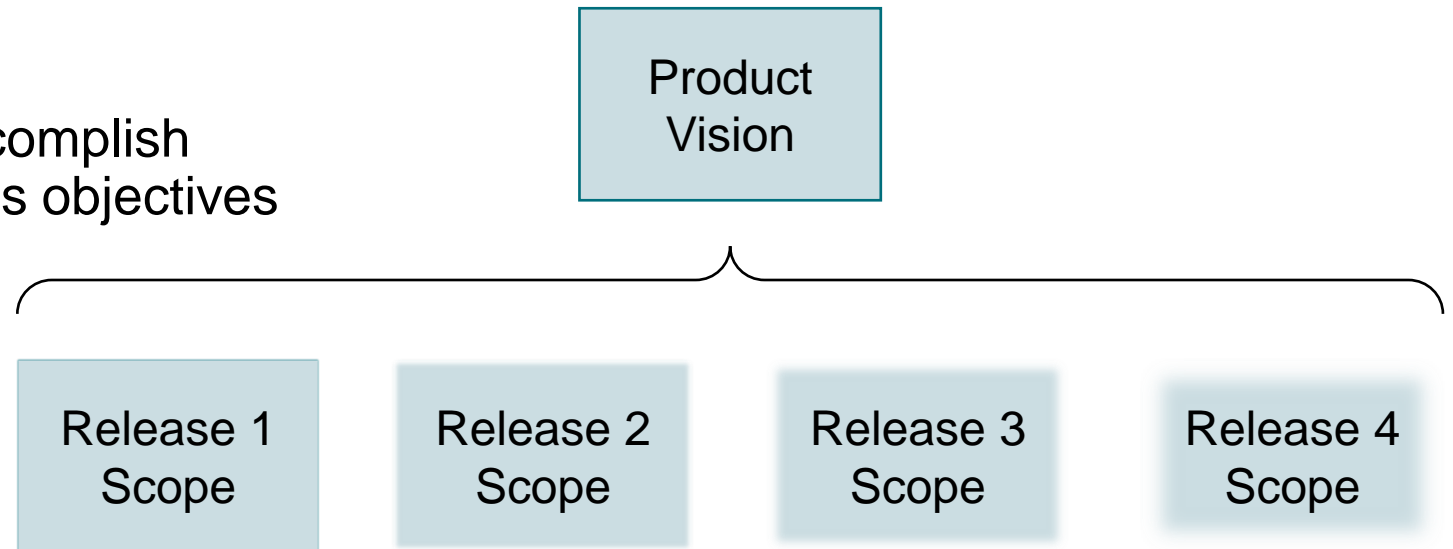
- Needs that lead to one or more projects to deliver desired business outcomes.
- Come from funding sponsors, executives, marketing, product visionaries...
- Provide a reference for making decisions on requirements, designs, changes...
- Set context for / enable measurement of benefits the project hopes to achieve.

- **Product vision**

- What we ultimately want to accomplish in order to achieve the business objectives

- **Project scope**

- What portion of the vision we will address in the upcoming project/release



Vision and Scope Document Template

1. Business requirements

1. Background
2. Business opportunity
3. Business objectives
4. Success metrics
5. Vision statement
6. Business risks
7. Business assumptions and dependencies

2. Scope and limitations

1. Major features
2. Scope of initial release
3. Scope of subsequent releases
4. Limitations and exclusions

3. Business context

1. Stakeholder profiles
2. Project priorities
3. Deployment considerations

- You can leave out sections that are not applicable to your project...
- ...but it's worth it to consider carefully
 - if they are maybe relevant after all
 - how you would fill them
 - what that means for your project

Vision and Scope Document: 1. Business Requirements

- **1.1 Background (“What triggered this?”)**
 - Rationale and context for new product/system or changes made to an existing one
 - History or situation that led to decision to build this product/system
- **1.2 Business opportunity (“Why does it look like we can be successful?”)**
 - Problem being solved / process being improved / market opportunity being targeted
 - Comparative evaluation of existing products/solutions and their inadequacy
 - Alignment with market trends, technology evolution, corporate strategy etc.
 - Needs of typical customers / users / target market
- **1.3 Business objectives (“Which benefits do we expect out of this?”)**
 - Measurable, realistic goals that the product/solution is expected to help achieve
- **1.4 Success metrics (“How can we tell whether we are successful?”)**
 - Measurable, quantifiable criteria that indicate whether the project was worth undertaking

Vision and Scope Document: 1. Business Requirements

- **1.5 Vision statement (“What will the product accomplish for whom?”)**
 - Summary of long-term purpose and intent of the product
 - Reflection of the expectations of all stakeholders
 - Idealistic – but grounded in realities of markets, technologies, architecture, resources etc.
 - Template:
 - For [target customer(s)]
 - who [need/opportunity],
 - the [product name]
 - is [product category]
 - that [major capabilities, key benefit, compelling reason to buy/use].
 - Unlike [primary competitive alternative, current system/process],
 - our product [primary differentiation and advantage of new product].

Vision and Scope Document: 1. Business Requirements

■ 1.5 Vision statement (“What will the product accomplish for whom?”)

Example:

- **For** scientists
- **who** need to request containers of chemicals,
- **the** Chemical Tracking System
- **is** an information system
- **that** will provide a single point of access to the chemical stockroom and to vendors.
 - The system will store the location of every chemical container within the company, the quantity of material remaining in it, and the complete history of each container’s locations and usage.
 - This system will save the company 25% on chemical costs in the first year of use by allowing the company to fully exploit chemicals that are already available within the company, dispose of fewer partially used or expired containers, and use a standard chemical purchasing process.
- **Unlike** the current manual ordering processes,
- **our product** will generate all reports required to comply with federal and state government regulations that require the reporting of chemical usage, storage, and disposal.

Vision and Scope Document: 1. Business Requirements

- **1.6 Business risks (“What could jeopardize the product’s success?”)**
 - Major business risks associated with developing – or not developing – the product
 - e.g. competition, timing, user acceptance, potential negative business impacts...
 - Not the same as technical project risks such as resource availability, technology factors etc.
 - Business risks can hit you even if your project runs smoothly from a technical perspective
 - Estimated likelihood and potential loss from each risk
 - Potential prevention/mitigation actions
- **1.7 Business assumptions/dependencies (“What are our plans based on?”)**
 - Assumptions / expectations / uncertainty about the development of external factors that form the basis for the preceding discussion of context, objectives, success metrics etc.
 - External factors that the execution / success of the project relies on (e.g. other projects’ deliverables, third party suppliers, legal regulations etc.)
 - Impacts of false assumptions and broken dependencies (these turn into risks!)

Vision and Scope Document: 2. Scope and Limitations

- **2.1 Major features (“What key things should the product be capable of?”)**
 - Major features or user capabilities (rule of thumb: the most important 10%)
 - especially those distinguishing it from previous / competing products
 - Describe e.g. as use cases
- **2.2 Scope of initial release (“What should be rolled out first?”)**
 - Beside features, scope can also comprise quality attributes (e.g. performance)
 - Focus on features that will provide the most value at the most acceptable cost to the broadest community in the earliest time frame
- **2.3 Scope of subsequent releases (“What can be rolled out later?”)**
 - Scope (and possibly rough timing) for later releases (more and more fuzzy towards future)
- **2.4 Limitations and exclusions (“What are we *not* going to do?”)**
 - Stakeholder expectations that will not be satisfied
 - Reasons for cutting aspects from the product scope

Vision and Scope Document: 3. Business Context

- **3.1 Stakeholder profiles (“Who has what interest in this project?”)**
 - People, groups, organizations that are
 - actively involved in the project
 - affected by its outcome
 - able to influence its outcome
 - Profile should include
 - Major value or benefit the stakeholder will receive from the product
 - e.g. in terms of improved productivity, reduced rework and waste, cost savings, process optimization, automation, ability to perform new tasks, compliance with regulations, improved usability, etc.
 - Likely attitude toward the product
 - Major features and characteristics of interest
 - Known constraints to be accommodated
 - Possibly: names of key stakeholders who can serve as representatives / contacts

Vision and Scope Document: 3. Business Context

■ 3.2 Project priorities (“What room is there for compromise?”)

- Unexpected events may require changes to a project’s scope, deadlines, resources, etc.
 - Example: If the product must be released a month ahead of schedule, can/should you...
 - ...defer certain requirements to a later release?
 - ...shorten planned system test cycle?
 - ...demand overtime from staff or hire additional contractors?
 - ...shift resources from other projects to help out?
- A general framework of priorities helps to decide how to best respond to such events.

- Five dimensions that can be managed: **Features, Quality, Schedule, Cost, Staff**

- **Constraint:** limiting factor that must be observed

- **Driver:** significant success factor that can’t be waived

- **Degree of freedom:** some adjustment possible here

<u>Example:</u>	Features	Quality	Schedule	Cost	Staff
Constraint				X	X
Driver		X	X		
Degree of freedom	X				

- (Not a substitute for talking to key stakeholders about best course of action when needed!)

Vision and Scope Document: 3. Business Context

- **3.3 Deployment considerations (“How will users obtain the product?”)**
 - Not a technical deployment specification
 - Key requirements that need to be taken into account in this earliest of project stages to enable efficient deployment later on:
 - Access required by users
 - Distribution of users
 - Platforms employed by users
 - Infrastructure requirements
 - Training requirements
 - Anything affecting the preparation of successful roll-out of the product that is not a feature/characteristic of the software itself (and thus easily overlooked at first)

In-Class Quiz #2: Vision and Scope Document



- **Business requirements**

- ...
- 3. Business objectives
- ...
- 5. Vision statement
- 6. Business risks
- ...

- **Scope and limitations**

- 8. Major features
- 9. Scope of initial release
- 10. Scope of subsequent releases
- 11. Limitations and exclusions

- **Business context**

- 12. Stakeholder profiles
- ...

- **Which section answers which question?**

- a) What are we *not* going to do?
- b) What can be rolled out later?
- c) What could jeopardize the product's success?
- d) What key things should the product be capable of?
- e) What should be rolled out first?
- f) What will the product accomplish for whom?
- g) Which benefits do we expect out of this?
- h) Who has what interest in the project?

Team Assignments 1 & 2

- Two of the most important artefacts created in the Inception phase are:

1. The **Vision and Scope document**

- Describing what you want to build, what its key features/capabilities will be, who will use it...
- Imagine having to convince your boss or an investor to provide funding for the project
 - What would they want to know to be convinced?

2. The initial **Use Case document**

- Describing the most important use cases of your product
 - i.e. the primary things that your system is supposed to be able to do
-
- Producing these documents and explaining the considerations that went into them will be your job in Team Assignments 1 and 2.

Team Assignment 1: Content

- By **Sun 23 Sep**, submit in Uglu:
 - A **Vision and Scope document** for your project
 - Following the template on slide 23 – most importantly, sections 1.5, 2.1
 - Use *brief* use case format for the 2-4 most important use cases in section 2.1
 - Write a paragraph on one 1.x section and one 2.x section of your choice as well
 - (“Convince your boss” – there are e.g. business considerations for games, too!)
- On **Thu 27 Sep**, present and **explain** your document to your tutor:
 - What considerations are behind your vision and scope document?
 - Why did you leave out certain sections of the template?
 - How did you come up with your use cases?
 - etc.

Team Assignment 1: Grading Criteria

- **Vision and Scope document**

- Precise vision statement in Sect 1.5 (25%)
- Precise formulation of 2-4 key use cases in brief format in Sect. 2.1 (25%)
- Plausible argument in chosen Sect. 1.x (25%)
- Plausible argument in chosen Sect. 2.x (25%)

Recap: General Assignment Format

- Required **artifacts** must
 - be produced by all team members together
 - be submitted in one PDF document by specified deadline in Uglu
 - contain your team number, the names and kennitölur of all team members
 - indicate who will present the assignment
- **Submissions** are due at 23:59 on Sundays
 - **No individual extensions** – missing artifacts receive a grade of 0!
 - But undefined grace period until whenever tutors download submissions from Uglu
 - Only the team member who will present must submit a document for the whole team
- The **presentation** must
 - be given by one representative of the team (a different one for each assignment)
 - be based on the submitted document (don't prepare extra slides)
 - take around 5-10 minutes (plus some questions asked by the tutor)

Course Schedule (updated)

Week	Thu: Team Consultations	Fri: Lectures	Sun: Assignments due
1		Introduction	
2	Phase 1: Inception	Rational Unified Process	Team Formation (9 Sep)
3	Phase 1: Inception	Requirements Engineering	
4	Phase 1: Inception	Requirements Engineering	#1: Vision & Scope Doc (23 Sep)
5	Vision & Scope Present. (27 Sep)	Spring Web Applications	#2: Use Case Document (30 Sep)
6	Use Cases Presentation (4 Oct)	Spring Web Persistence	
7	Phase 2: Elaboration	Guest Lecture	
8	Phase 2: Elaboration	Behavior Models	
9	Phase 2: Elaboration	Domain Models	#3: Behavior Model (28 Oct)
10	Behavior Model Present. (1 Nov)	Design Models	
11	Phase 3: Construction	Design Patterns	Exchange code w other team(11 Nov)
12	Phase 3: Construction	Design Patterns	#4: Code Review Report (18 Nov)
13	Code Review Present. (22 Nov)	Web Presentation Layer	
14	Project Presentation (29 Nov)	Final Exam Prep	#5: Final Project (2 Dec)

Use Cases

see also:

- Larman: Applying UML and Patterns, Ch. 6



Types of Requirements

Functional Requirements

- Features
- Capabilities
- Business rules
- Security measures

In RUP, described in Use Case Model, Business Rules, Software Requirements Specification documents

In RUP, described in Supplementary Specification document

Quality (“non-functional”) Requirements

- Usability
 - Human factors, help, documentation
- Reliability
 - Failure rates, recoverability, predictability
- Performance
 - Response times, throughput, accuracy...
- Supportability
 - Adaptability, maintainability, internationalization...
- Constraints
 - Technology, external interfaces, operational setting, legal conditions...

Use Cases

- In the Unified Process, we **discover and record functional requirements** by writing **text stories of an actor using a system to meet goals**.
 - **Simple example: “Process Sale” use case (brief format)**
 - 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 the inventory. The customer receives a receipt from the system and leaves with the items.
 - **Emphasis on users’ goals and perspectives:**
 - Who is using the system?
 - What are their typical scenarios of use?
 - What are their goals?
- More user-centric than just coming up with a list of system features



Use Case Terminology

■ Actor

- Something with a behavior, e.g. a person (role), system or organization
- **Primary actor:** Has user goals fulfilled through using services of system under development
 - Identify these to find user goals, which drive use cases
- **Supporting actor:** Provides a service/information to the system under development
 - Identify these to clarify external interfaces and protocols
- **Offstage actor:** Has interest in behavior of use case, but is not active in its execution
 - Identify these to ensure necessary interests are satisfied and conditions are met

■ Scenario (or: use case instance)

- A specific sequence of (inter)actions between actors and the system
- One particular story of using a system, or one path through the use case
 - e.g. a successful path (success scenario) or a path involving an error/failure (alternate scenario)

■ Use case

- A collection of related success and failure scenarios describing an actor using a system to support a goal

Use Case Content

➤ Official definition in the RUP:

- A use case is a **set of scenarios** where each scenario is a **sequence of actions** a system performs that yields an **observable result** of **value** to a particular **actor**.

■ Key writing challenge: Capturing the requirements precisely

- Understand and formulate the essence of the requirement
- Don't be too detailed, don't be too high-level
- Don't combine several alternatives in one scenario
- Don't forget to consider and describe alternate scenarios
- Don't interpolate what you're not sure about – ask!
- Don't preempt technical design decisions

Use Case Formats

- **Brief format**

- Terse one-paragraph summary, usually of the main success scenario
- Written during early requirements analysis to get a quick sense of subject and scope

- **Casual format**

- Multiple informal paragraphs that cover various scenarios
- Typically including at least the main success scenario and most relevant alternate scenarios
- Written during early requirements analysis to get a quick sense of subject and scope

- **Fully dressed format**

- All steps and variations described in detail, with supporting sections on preconditions etc.
- During first requirements workshop, about 10% of architecturally significant, high-value use cases are written out in this detail; rest may follow in further iterations as needed.

- **UML use case diagrams**

- Secondary, supplementary illustrations only – essential information is contained in the text!

Example: Casual Use Case Format



“Handle Returns” use case

■ Main success scenario

- A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item. [...]

■ Alternate scenarios

- If the customer paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash.
- If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).
- If the system detects failure to communicate with the external accounting system, [...]