Fachhochschule Nordwestschweiz
Hochschule für Technik

Software Architecture

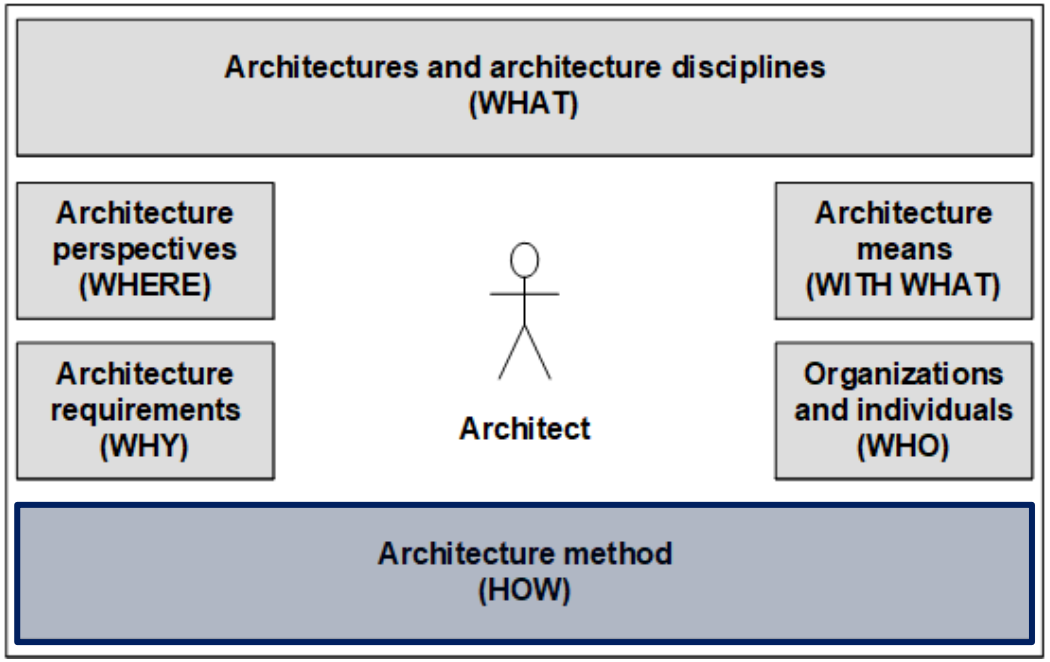
Architecture Methodology

BSc

Ingo Arnold

Lecture Opening
Architecture Orientation Framework

Architecture HOW discusses the process of systematically elaborating an architecture solution in response to a corresponding problem – in other words, with **architecture development models** [Vogel, Arnold et al 2011].



Lecture Opening
Motivation

By architecture we mean the **process of developing architecture as well as the result of this process**.

To the extent that architecture view models structure the manifestation of holistic architecture representation, **methodologies systematize the process of architecture planning, development and validation**.

Methodologies and view models go hand in hand, which is why the architecture frameworks referenced above also include process models in addition to view models.

Lecture Opening
Learning Objectives

You …

- gain an overview of the basic features of modern architecture methodologies.
- understand that methodologies are adapted to the contexts in which they are used.
- know the role of architecture mandates for architecture methodologies.
- understand the central components as well as attitudes (i.e., evolutionary approach) of solution architecture methodologies.

Lecture Agenda

- Methodology
- Architecture Mandate
- Solution Architecture Methodology

Methodology
Overview

A **methodology systematically directs efforts and activities toward defined objectives**. It specifies the valid state and transition space for its respective field of concern.

A methodology includes a method in which it **outlines a systematic, step-by-step procedure**. Each step in the procedure specifies what information is needed to execute the step, what information is generated by the step, and what role is responsible for performing the step.

A methodology embodies the **deep structure of its method** and other **complementary building blocks** regardless of their instantiation.

Methodologies **facilitate discussions about the appropriateness of designs** while abstracting from their material implementation aspects.

Methodologies are **inevitably linked to method-specific view models** – thus practically inseparable from them.

Methodology
Overview

An architecture methodology increases the **systematicity and repeatability of architecture governance and performance activities** in an enterprise.

Methodologies **promote alignment, standardization, synergy, and economies of scale** – ultimately, methodologies promote an attitude of repetition rather than reinvention.

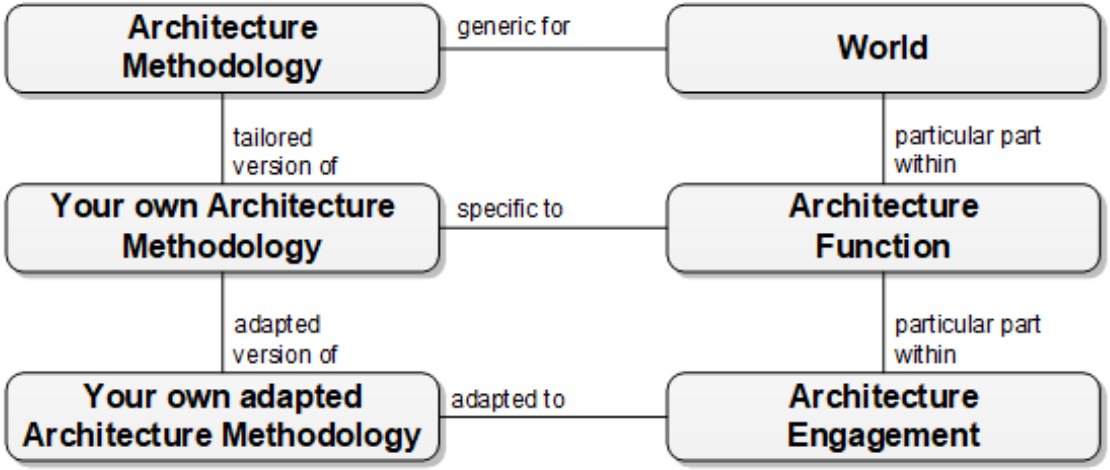
Architecture methodologies tangibly **explain the *why*, *what*, *who*, *when*, and *how* of states and state transitions** for their particular areas of responsibility.

Different types of architecture methodologies exist:

- **Solution Architecture Development** (e.g., Unified Process, TOGAF, ArchiMate,)
- **Enterprise and Domain Architecture Elaboration** (e.g., TOGAF, Zachman)
- **Reference Architecture Elaboration** (e.g., architecture pattern or roadmap methodologies)
- **Architecture Assessment Methodology** (e.g., ATAM).

Methodology
Method Adaption

In any case, methodologies **must always be adapted to the context** in which they are applied.



Methodology
Method attitude and approach

The **simplest solution architecture elaboration method** is comprised of (a) develop solution and (b) find deficits and resolve them.

More advanced methods …

- reduce failure cost
- improve repeatability of the solutioning process
- adopt an iterative attitude and approach to solution development

Lecture Agenda

- Methodology
- Architecture Mandate
- Solution Architecture Methodology

Architecture Mandate
Overview

The **architecture discipline collaborates** with other disciplines along the enterprise value chain.

Some collaborations are lightweight, informal by nature and part of normal day-today operations. Other collaborative relationships, however, require clarification of the expectations, assumptions, and commitments of all cooperating parties.

Inadequately aligned cooperation inevitably leads to ambiguity about scope, objective, deliverables, timelines, or resources.

The purpose of an a**rchitecture mandate is to encourage and require appropriate clarification** before any commitments are made. A well-defined mandate promotes clarity of expectations on all sides by fully describing, and formally agreeing on commitments.

Architecture Mandate
Overview

Why are architecture mandates so crucial?

More often than we would like to admit, **we act without having sufficiently clarified the goal of our actions** for ourselves.

The reason that this happens to us is that we are all too **often hastily satisfied with an initial understanding of a given problem**.

We **do not listen enough** or **reflect long enough** to distinguish between a *supposed understanding* of the problem (immature understanding) and an actual understanding of a given problem (mature understanding).

Architecture **mandates increase our awareness** as they force us to go through a dedicated reflection process.

Architecture Mandate
Supposed versus Actual Need

I am presenting a model that accompanies us practically all the time in life. In this conception, I distinguish needs into conscious versus unconscious and actual versus supposed needs.

A **supposed need is one we believe we have, when in fact we unconsciously need something else**.

An **actual need assumes the existence of an optimum, which at the same time is unknown to us**.

We can only approach our actual needs through a questioning and sceptical attitude, but at the same time we can never determine that we have achieved it.

While questioning the question risks infinite regression, pragmatic heuristics help you avoid philosophical rabbit holes.

Architecture Mandate
Supposed versus Actual Need

Below I distinguish between a **supposed goal** and an **actual goal** on the one hand, and the **successful** or **unsuccessful pursuit of the goal** on the other hand.

Discuss the corresponding combinations (A, B, C, D) and evaluate each in terms of how desirable they are.

		aim: supposed	aim: actual
pursuit of aim: successful	successful	A	B
pursuit of aim: unsuccessful	unsuccessful	C	D

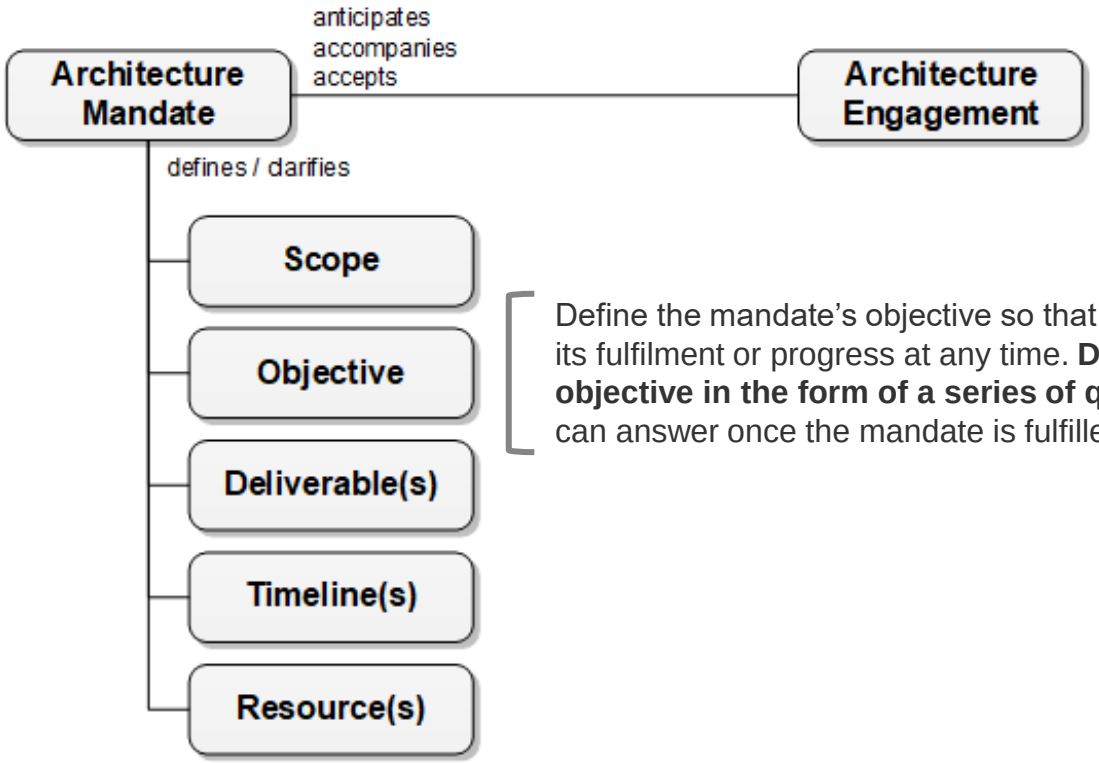
Architecture Mandate
Supposed versus Actual Need

Discuss the corresponding combinations (A, B, C, D) and evaluate each in terms of how desirable they are.

Architecture Mandate

Architecture Mandate Specification

Architecture mandates are used to precisely **anticipate and articulate the desired end state of an architecture activity** and guide the path to its achievement



Define the mandate's objective so that you can determine its fulfilment or progress at any time. **Describe the objective in the form of a series of questions** that you can answer once the mandate is fulfilled.

Architecture Mandate
Architecture Mandate Specification

Once defined, architecture mandates **support architects in performing the mandated engagements**.

While performing architecture activities in a corresponding engagement, architects use mandates to check whether their contributions have already achieved the desired goal (i.e., mandates serve architects as a *definition of done*).

Mandates, therefore, **accompany architecture engagements as a kind of yardstick**. This yardstick is continuously used to specifically plan and implement architecture contributions and to determine when they have been sufficiently delivered. Therefore, mandates are often incorporated in methods as a calibration companion.

Lecture Agenda

- Methodology
- Architecture Mandate
- Solution Architecture Methodology

Solution Architecture Methodology
Overview

Solution architecture methodologies standardize both the development and description of asset architectures.

That is, each solution methodology encompasses both a **development process and a view model**.

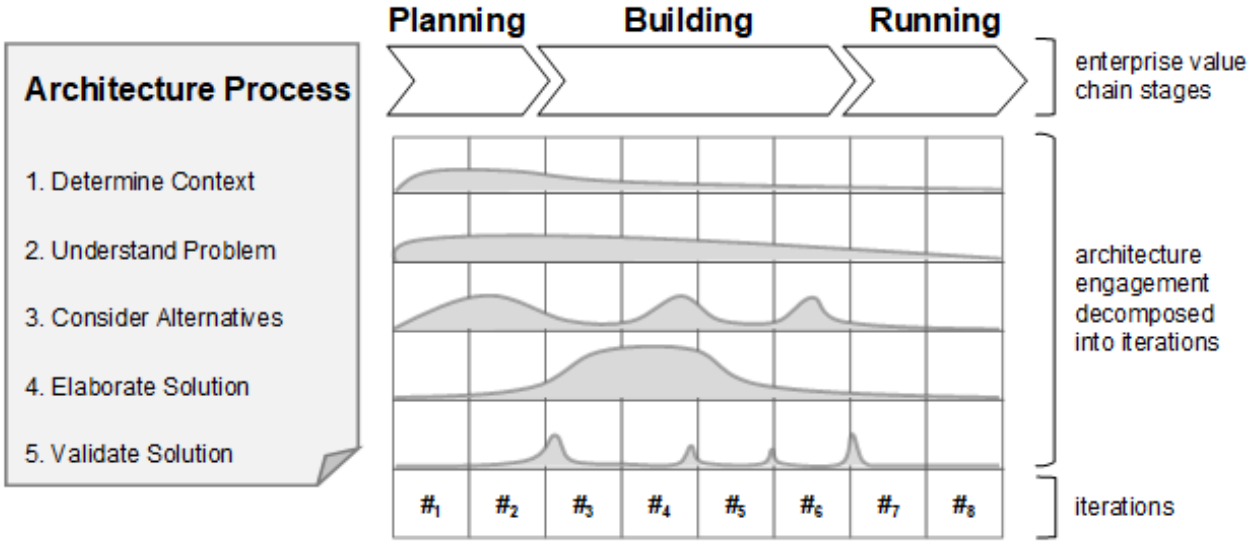
While a solution architecture development process defines the activities to be performed, a view model specifies what information is captured for which viewpoints to create an architecture description.

A well-designed solution methodology takes a **holistic approach to developing, capturing, and describing the solution design** in question. It avoids monocausal thinking and monodimensional capture of the architecture but strives to absorb the relevant dimensions of the given problem.

Furthermore, solution methodologies must consider that both **problems and their solutions evolve and change over time**. Thus, any meaningful methodology must address the evolutionary nature of problem-solving.

Solution Architecture Methodology
Overview

A **solution description is developed incrementally** through the repeated application of what appears to be a sequential process. Accordingly, one view of a solution view model is not completed at a time. Rather, all relevant views are also repeated again and again – and thus established incrementally.



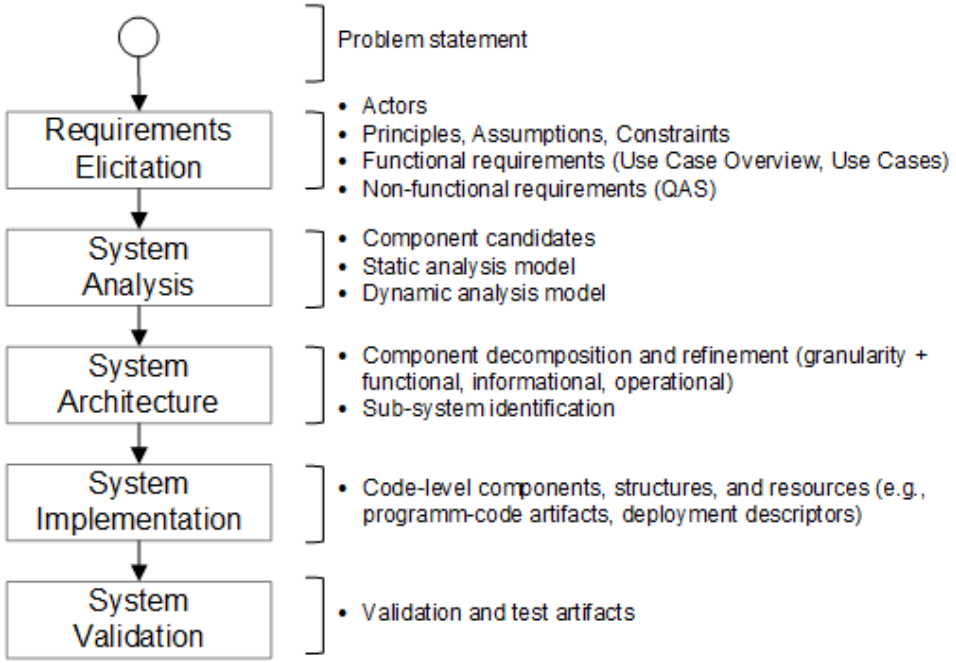
Solution Architecture Methodology
Overview

The procedure outlined below is certainly not sufficient in detail and scope, but it contains the genuinely essential ingredients of a solution architecture elaboration process.

- **Determine architecture context**. Determine the enterprise context of a given problem and envisioned solution. Understand how the enterprise context affects an envisioned solution (inbound dependencies) and, conversely, how a solution potentially affects its context (outbound dependencies).
- **Determine and refine architecture conditions**. Within the given conditions, identify those that are architecture-significant. Architecture conditions will significantly shape the design of your target architecture
- **Develop architecture alternatives and select alternative**. Investigate and develop alternative approaches to solve a given problem. Conduct this activity based on problem-specific selection metrics and select the most appropriate alternative based on them. In later iterations, repeat this activity in recursive descent for individual solution building blocks of the overall architecture.
- **Elaborate solution architecture**. Develop a solution design to respond to the identified architecture conditions and ensure that a given mandate is accomplished. Establish traceability across coherent views in such a way that you create a holistically navigable design description.
- **Validate solution architecture**. Perform architecture validation at the end of major progress, such as at the end of each iteration. What you specifically validate depends on where you are in your solution development.

Solution Architecture Methodology
Problem-solving core

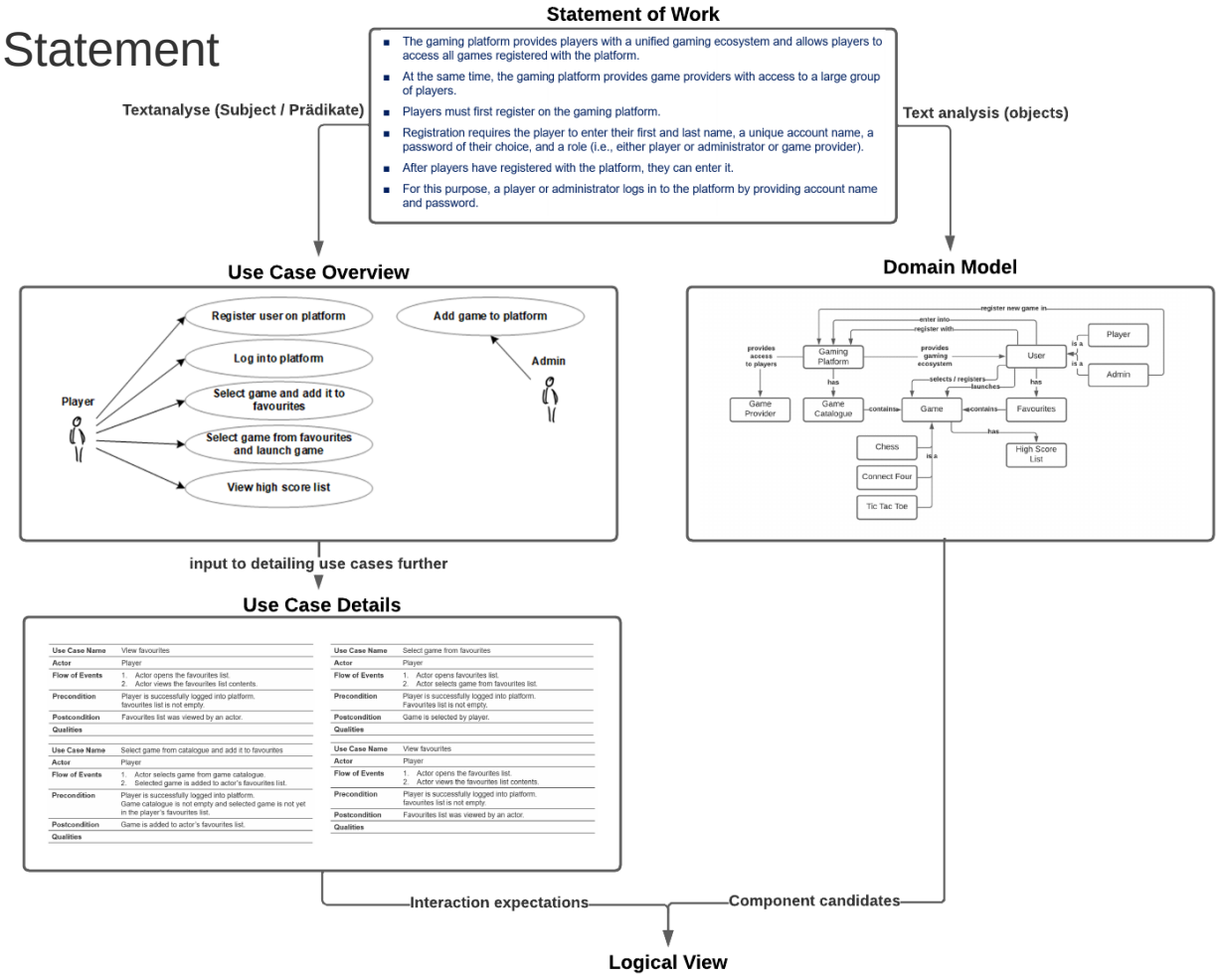
The core of the process spanning both, requirements and solutioning activities in is outlined in more detail, below.



Solution Architecture Methodology

From Problem to Solution

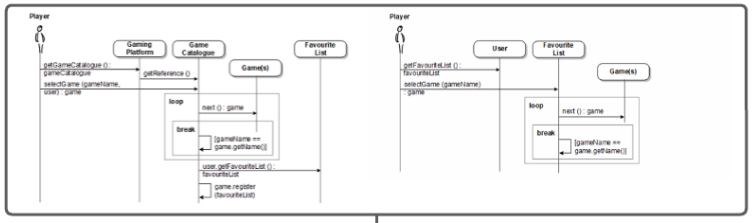
Problem Statement



Solution Architecture Methodology
From Problem to Solution

System Analysis

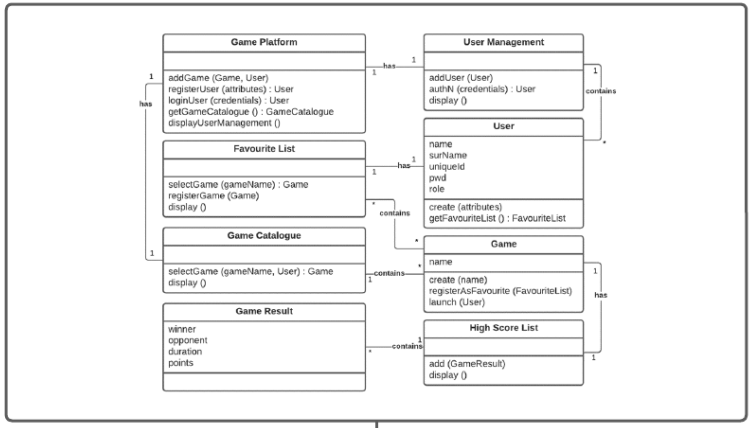
Logical View



In use case realisations, functional components and their methods are identified. The set of all these components is able to realise all functional expectations.

Generalising the components in classes and relationships between classes

Development View



If all essential use cases have been fully considered, the essential functional (business) components as well as their functional shells have been identified through this approach. Furthermore, it can be assumed that these components and their functional shells will not change significantly.

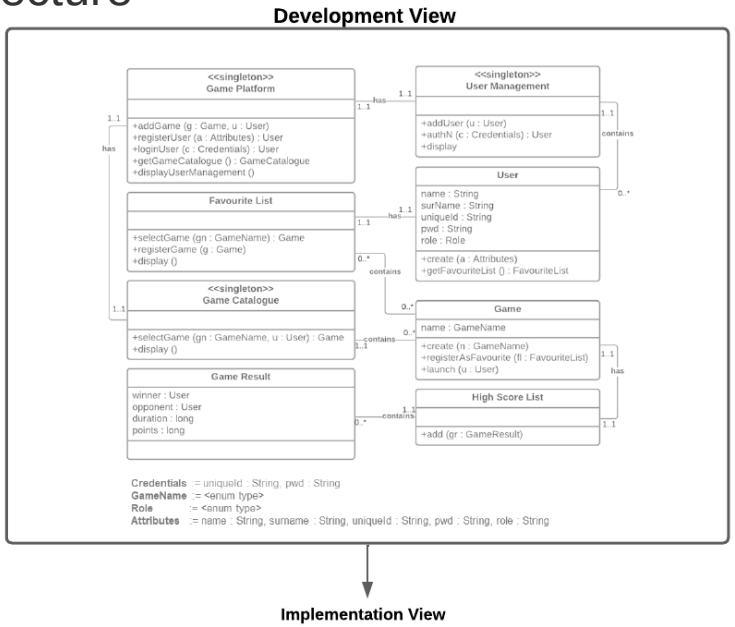
Elaborating System Architecture

Development View

Solution Architecture Methodology

From Problem to Solution

System Architecture



At the system architecture level, the analysis model is enriched with design-related details such as patterns, data types, further components.

Solution Architecture Methodology
From Problem to Solution

System Implementation

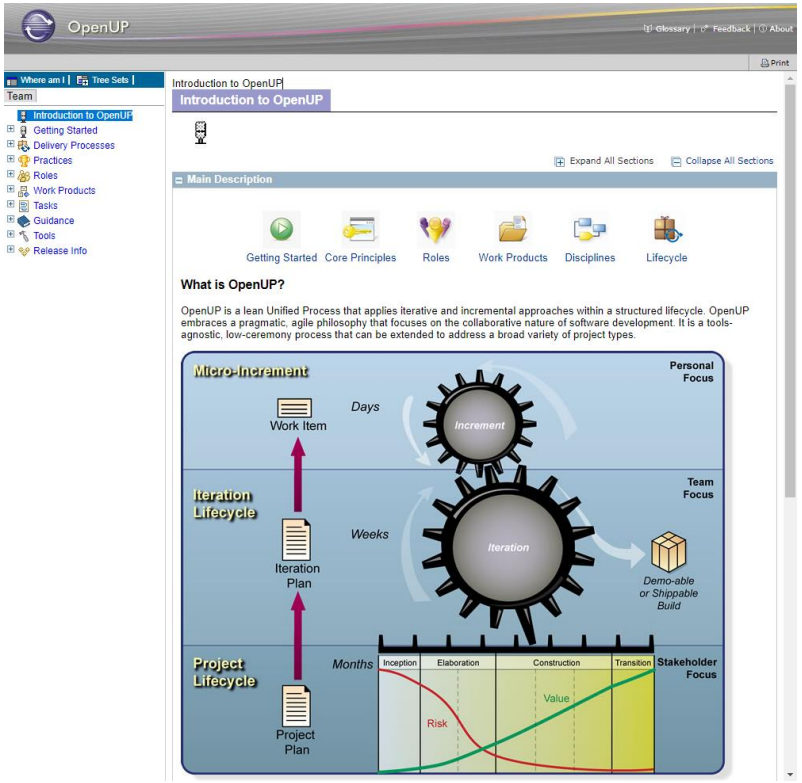
Implementation View

```
public class GamePlatform {
    private static GamePlatform singleton = null;

    // singleton factory
    private GamePlatform() {}
    public static GamePlatform getGamePlatform() {
        if(singleton == null)
            singleton = new GamePlatform();
        return singleton;
    }

    // instance methods
    public void registerUser(String name, String surname, String uniqueId, String pwd1, String pwd2, Role role) {
        UserManagement um = UserManagement.getUserManagement();

        if(pwd1.contentEquals(pwd2)) {
            User u = new MyUser(name,surname,uniqueId,pwd1,role);
            um.addUser(u);
        }
    }
    public void addGame(Game game, User user) {
        // is user authenticated?
        if (user != null) {
            // does user have admin role?
            if (user.getRole().equals(Role.ADMIN)) {
                GameCatalogue gc = GameCatalogue.getGameCatalogue();
                gc.add(game);
            }
        }
    }
```

On the basis of the models at system architecture level, a solution is implemented with suitable implementation language/technology. In the code artefacts "model" at implementation level, the software developer has degrees of freedom in design that are not predetermined by the system architecture.

Solution Architecture Methodology
Problem-solving framework

OpenUP [OpenUP 2021] is a process instance of the eclipse process framework (EPF) – an eclipse project.



Solution Architecture Methodology

Another perspective

Solution Architecture Methodology

Another perspective

$a = b$

Solution Architecture Methodology

Another perspective

$$
\begin{aligned}
a &= b && \mid \cdot b \\
a^2 &= ab && \mid -b^2 \\
a^2 - b^2 &= ab - b^2 && \mid \Sigma \to \Pi \\
(a+b)(a-b) &= b(a-b) && \mid : (a-b) \\
a + b &= b && \mid \text{remember } a = b \\
a + a &= a \\
2a &= a && \mid : a \\
2 &= 1 && \mid \text{Hm!}
\end{aligned}
$$

Solution Architecture Methodology
Exercise

*6-BSc_SWA_ServletEngine

Bibliography
Lecture

[Arnold 2021]
Arnold, Ingo, *Enterprise Architecture Function — a pattern language for planning, designing, and executing*, Springer Science and Business Media, Berlin Heidelberg, 2021

[OpenUP 2021]
Eclipse, Eclipse process framework – OpenUP, 2021, https://download.eclipse.org/technology/epf/OpenUP/published/openup_published_1.5.1.5_20121212/openup/index.htm

Questions