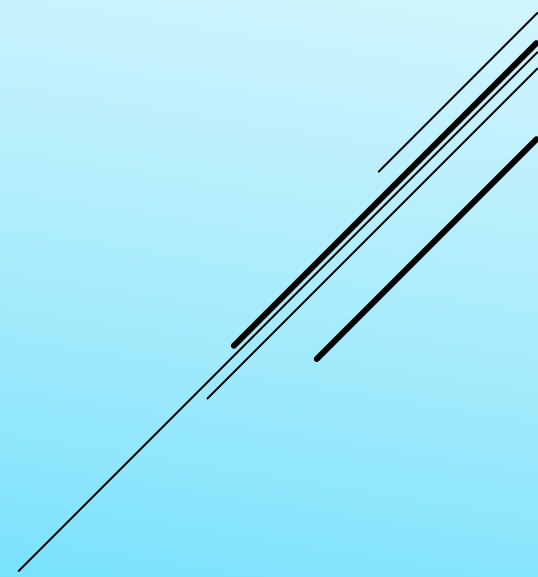



Development of Information Systems

Overview of topics and tasks

2021-22



Teaching information

- RNDr. Ing. Martin Radvanský, M.Sc., Ph.D.
 - EA 439, +420 597 325 877
 - homel.vsb.cz/~rad0028
 - martin.radvansky@vsb.cz
 - Lectures
 - Knowledge
 - Competence
 - Guided exercises
 - Discussion
 - Presentation
 - Notes on the development and use of technology
 - Design and programming
- 

Key goals of this subject

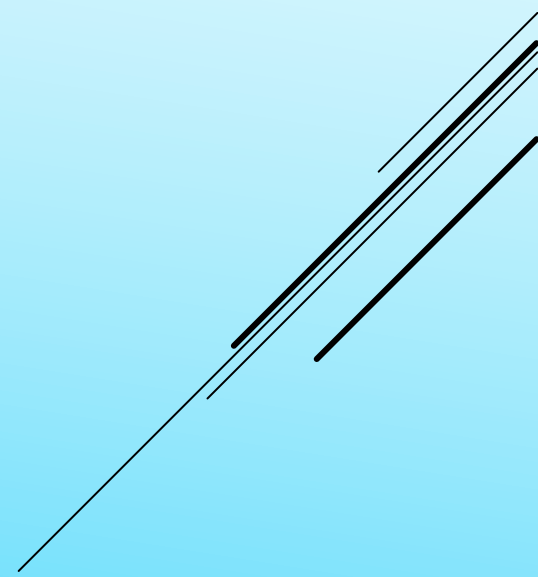
- Understanding what, why and how it's done.
To do well is to do as others do successfully.
- Know technology from the other side.

Stakeholder, Architecture, project
management, developer, analytics

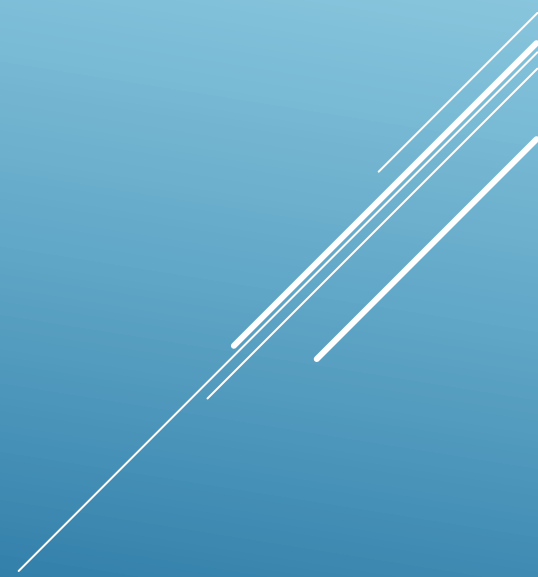
A series of parallel diagonal lines in the bottom right corner, consisting of a thin black line, a medium grey line, and a thick black line, all slanting upwards from left to right.

Input knowledge

- Object-oriented approach
- UML
- Design patterns (GoF)
- Programming C#, JAVA
- Database design



Basic terminology



Information system

“Information systems (IS) is the study of complementary networks of hardware and software that people and organizations use to collect, filter, process, create, and distribute data.” [Wikimedia]

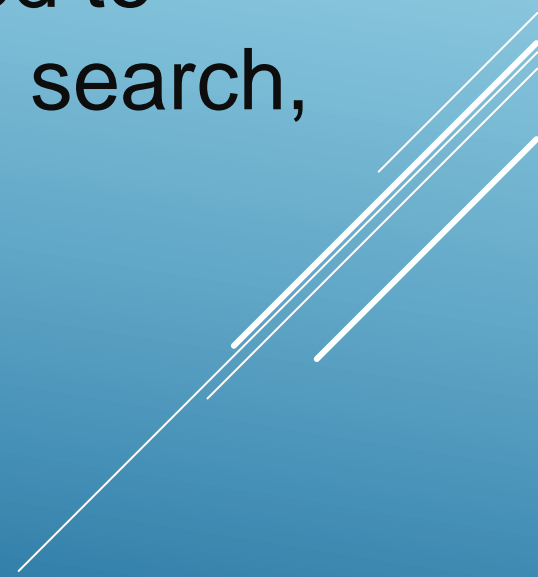
“Information systems are combinations of hardware, software, and telecommunications networks that people build and use to collect, create, and distribute useful data, typically in organizational settings.” [*Information Systems Today*]

“Information systems are interrelated components working together to collect, process, store, and disseminate information to support decision making, coordination, control, analysis, and viualization in an organization.” [*Management Information Systems*]

Information system

In a wide sense, it is about the interaction between people, processes and data.

An information system is designed to process (retrieve, transmit, store, search, manipulate, display) information.

Several thin, white, parallel diagonal lines are located in the bottom right corner of the slide, extending from the middle of the right edge towards the bottom left.

Components of Information systems

Five main components:

- Hardware
- Software
- Data

Technology

- People
- Process

*separate the idea of
information systems
from more technical
fields*

Components of Information systems

Technology can be thought of as the application of scientific knowledge for practical purposes.

- **Hardware** is the part of an information system you can touch – the physical components of the technology.
- **Software** is a set of instructions that tells the hardware what to do.
- **Data** you can think of data as a collection of facts. For example, your street address, the city you live in, and your phone number are all pieces of data.

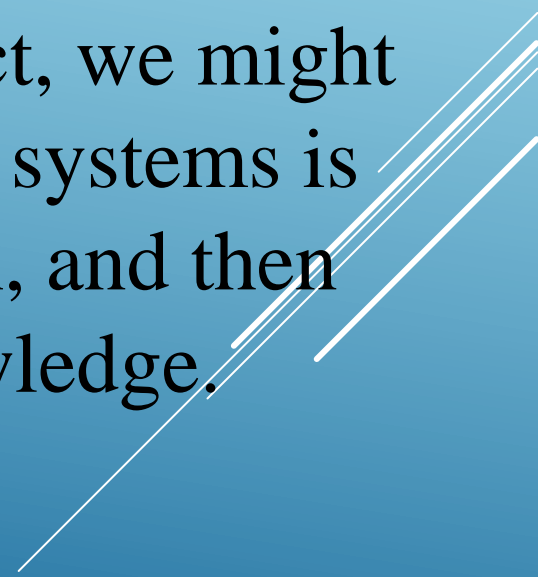
Components of Information systems

- **People** when thinking about information systems, it is easy to get focused on the technology components and forget that we must look beyond these tools to fully understand how they integrate into an organization. **A focus on the people involved in information systems is the next step**
- **Process** is a series of steps undertaken to achieve a desired outcome or goal. Information systems are becoming more and more integrated with organizational processes, bringing more productivity and better control to those processes.

Information systems purpose

What do these components actually do for an organization?

From previous definitions, we see that these components collect, store, organize, and distribute data throughout the organization. In fact, we might say that one of the roles of information systems is to take data and turn it into information, and then transform that into organizational knowledge.


Several white lines of varying thickness and length are drawn diagonally across the bottom right corner of the slide, extending from the right edge towards the bottom left.

History

Era	Hardware	Operating System	Applications
Mainframe (1970s)	Terminals connected to mainframe computer.	Time-sharing (TSO) on MVS	Custom-written MRP software
PC (mid-1980s)	IBM PC or compatible. Sometimes connected to mainframe computer via expansion card.	MS-DOS	WordPerfect, Lotus 1-2-3
Client-Server (late 80s to early 90s)	IBM PC “clone” on a Novell Network.	Windows for Workgroups	Microsoft Word, Microsoft Excel
World Wide Web (mid-90s to early 2000s)	IBM PC “clone” connected to company intranet.	Windows XP	Microsoft Office, Internet Explorer
Web 2.0 (mid-2000s to present)	Laptop connected to company Wi-Fi.	Windows 7	Microsoft Office, Firefox
Post-PC (today and beyond)	Apple iPad	iOS	Mobile-friendly websites, mobile apps

Classification of IS


Different types of agendas

- Economic
 - Personnel
 - Stock
 - Documentary (e.g. file service)
 - School
-
- ERP, CRM, CMS, DMS, Project management...
- 

Software architecture

There are many definitions of software architecture, easily discoverable with a web search, but the one we like is this one:

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.

Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

Software architecture

- Architecture is a set of software structures. (modules, components, allocations)
- Architecture is an abstraction. (selects certain details and suppresses others)
- Every software system has a software architecture.
- Architecture includes behavior (how elements interact with each other)

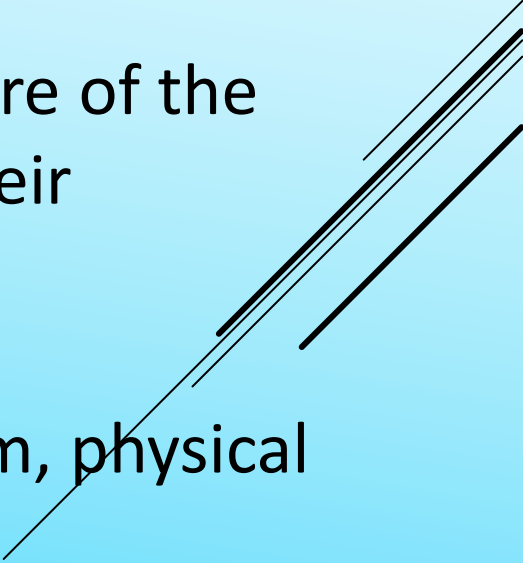
Not all architectures are good architectures

Architecture types

- **System architecture** is a representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and a concern for the human interaction with these components.
- **Enterprise architecture** is a description of the structure and behavior of an organization's processes, information flow, personnel, and organizational subunits, aligned with the organization's core goals and strategic direction.

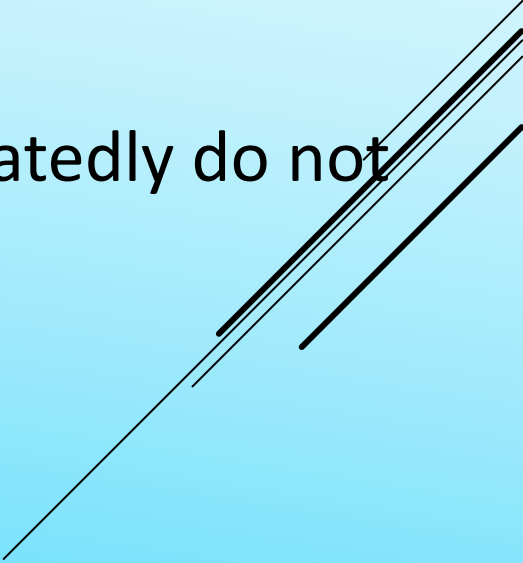
Architecture of IS

The architecture of an information system lies at a higher level of abstraction by including:

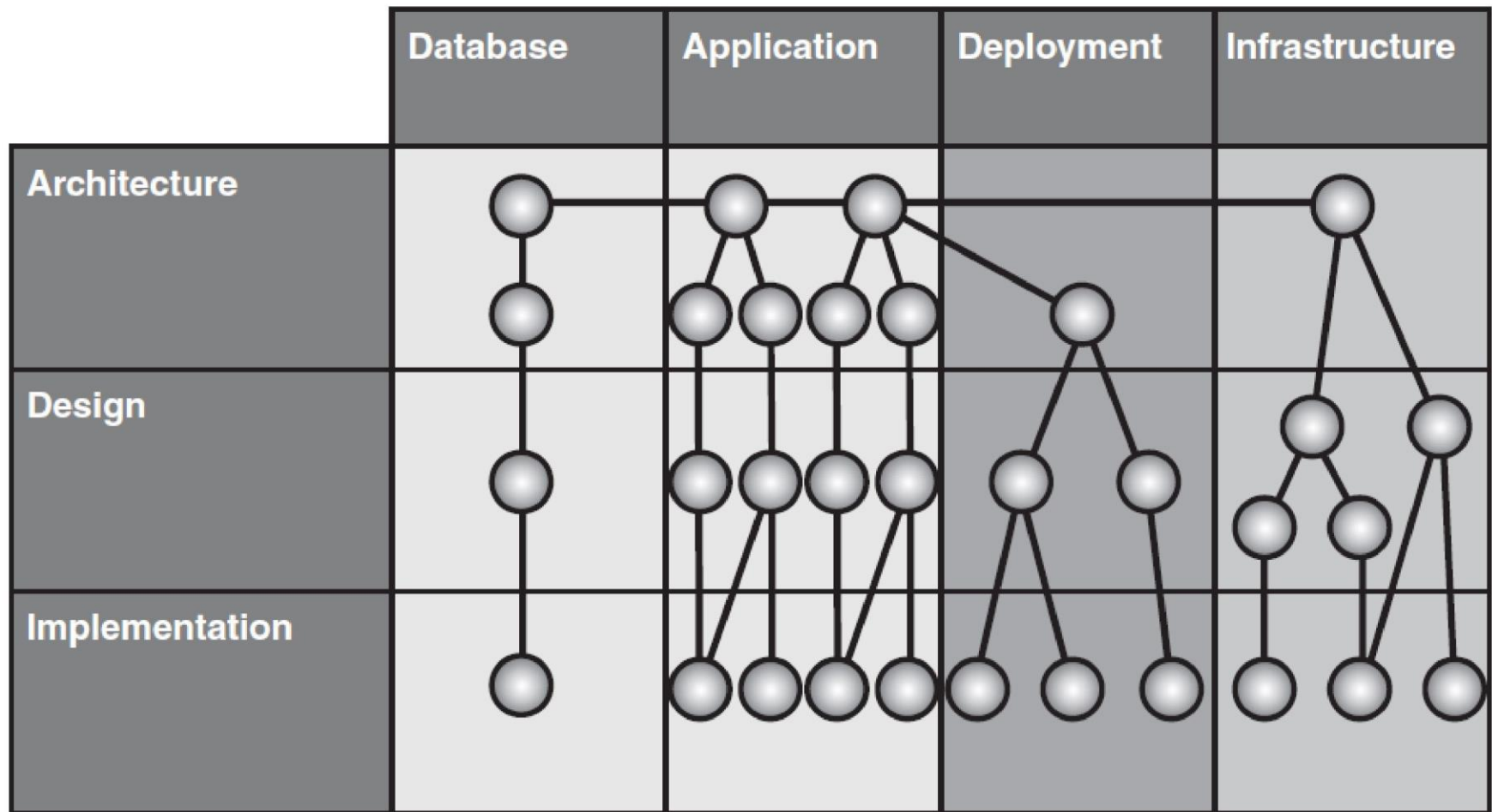
- a domain view of the information system (a group of related "things" from the customer's perspective);
 - a developer's view of the global structure of the system and the behavior of its parts, their interconnection and synchronization;
 - data access and data flows in the system, physical deployment of components;
- 
- A series of three parallel diagonal lines in the bottom right corner of the slide, consisting of a thin black line, a medium-thick grey line, and a thick black line.

	What	How	Where	Who	When	Why	
Scope							Strategists
Business							Executive Leaders
System							Architects
Technology							Engineers
Component							Technicians
Operations							Workers
	Inventory	Process	Network	Organization	Timing	Motivation	

Patterns – guidelines for common tasks

- GoF (creation, structure, behaviour).
 - Patterns and languages of patterns.
 - The pattern is a reusable solution to a problem that works.
 - There are also anti-patterns (they repeatedly do not work).
- 
- A decorative graphic consisting of several parallel diagonal lines in black and light blue, extending from the bottom right towards the center of the slide.

Architect and developer view (levels of abstraction)



Three-Layered Application

Context

You are building a business solution using layers to organize your application

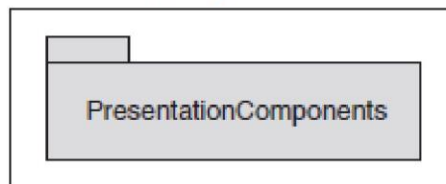
Problem

How do you organize your application to reuse business logic, provide deployment flexibility and conserve valuable resource connections?

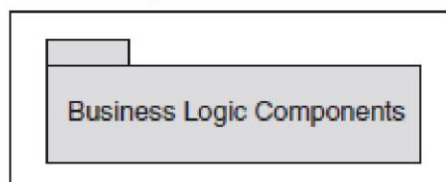
Solution

Create three layers: presentation, business logic (domain), and data access. Place all components responsible for the view in the presentation layer. Encapsulate all business logic in domain layer components that implement well-known component interfaces. Locate all database-related code, including database client access and utility components, in the data access layer. Require the data access layer to be responsible for connection pooling when accessing resources. Make sure you eliminate the dependencies between data access components and business layer components. Either eliminate dependencies between the business layer and the presentation layer or manage the dependencies here using the *Observer* pattern.

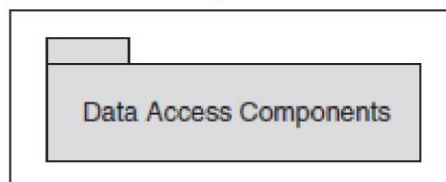
Presentation Layer



Domain Layer



Data Access Layer



Three-Layered Services Application

Context

You are building a business solution that uses presentation, business, and data access layers to organize your application. You want to expose some of the core functionality of your application as services that other applications can consume and enable your application to consume other services.

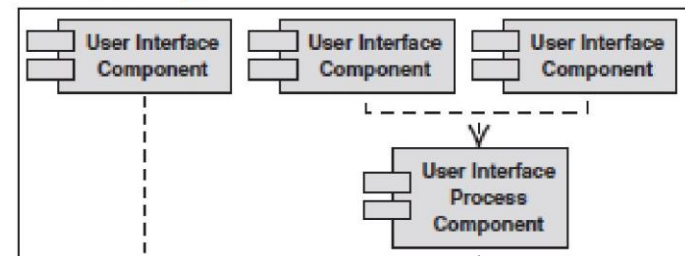
Problem

How do you organize your application to provide and consume granular data and logical elements from highly variable sources?

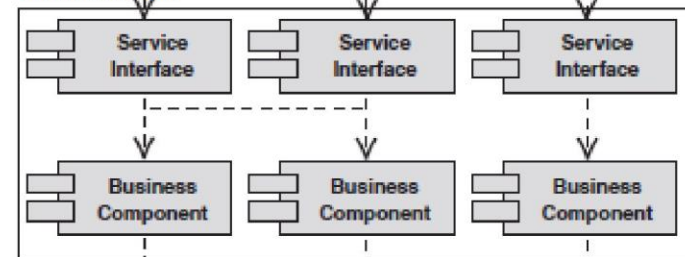
Solution

Decompose your application logic into a collaborating set of services that provide parts of the overall system functionality. Next, in the domain layer, identify a Service Interface for each service that is independent of the underlying implementation. Finally, extend the data access layer to use Service Gateways to communicate with other service providers. If application navigation logic is sufficiently complex, consider user interface process components as part of the presentation layer to encapsulate and reuse this logic.

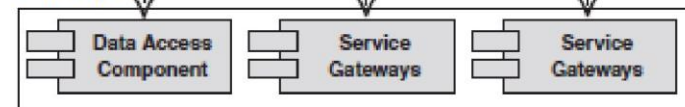
Presentation Layer



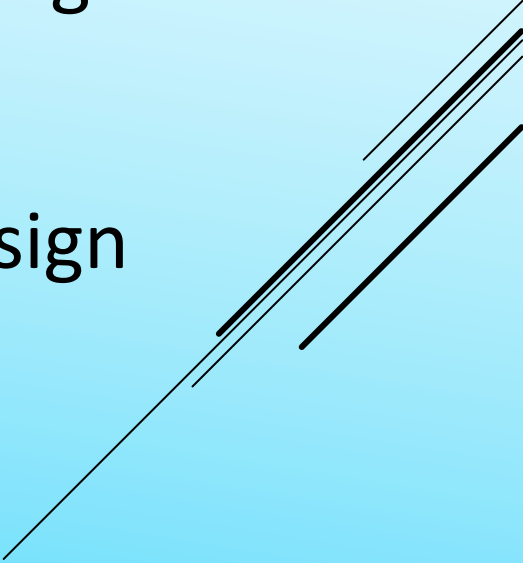
Business Layer



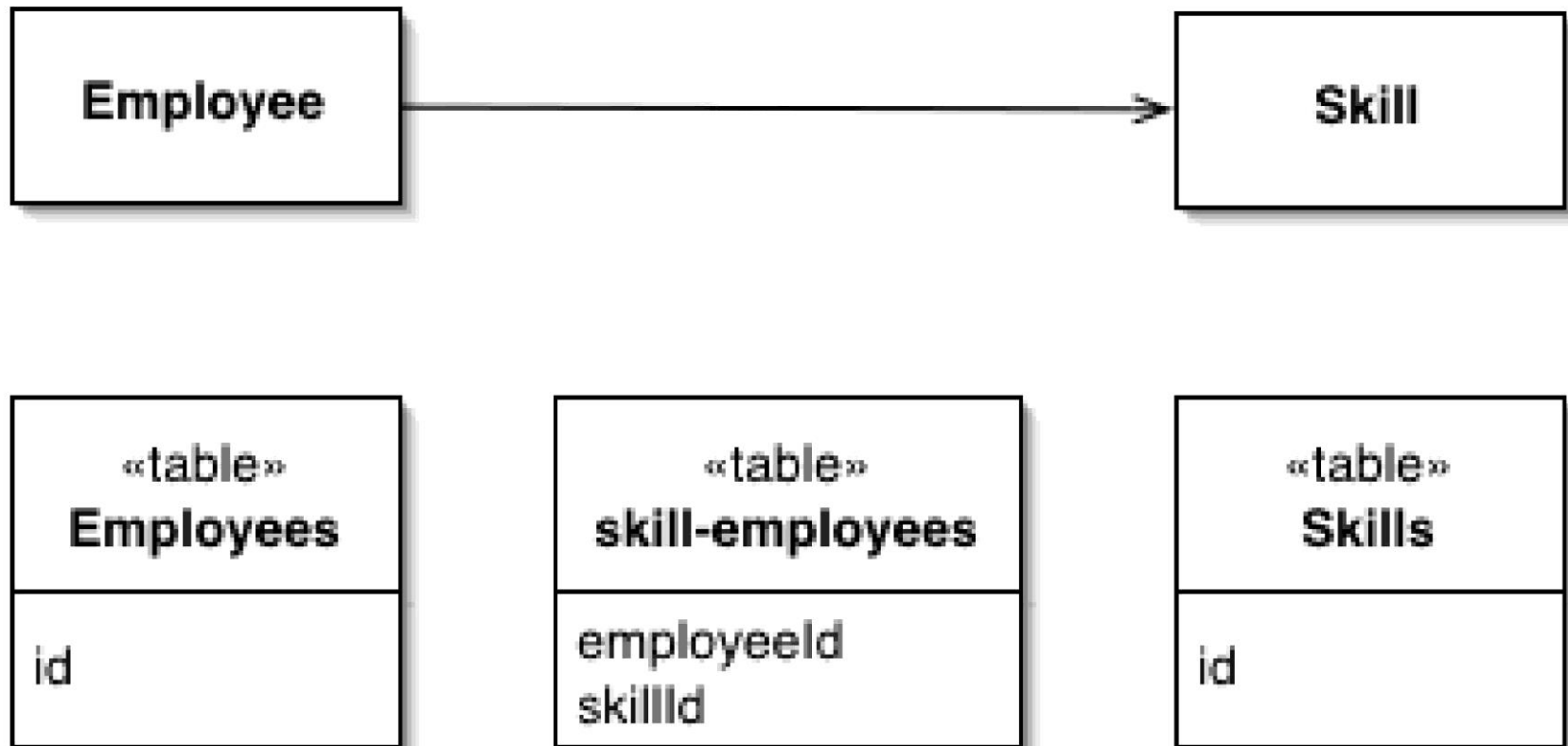
Data Layer



Patterns for "enterprise" architecture

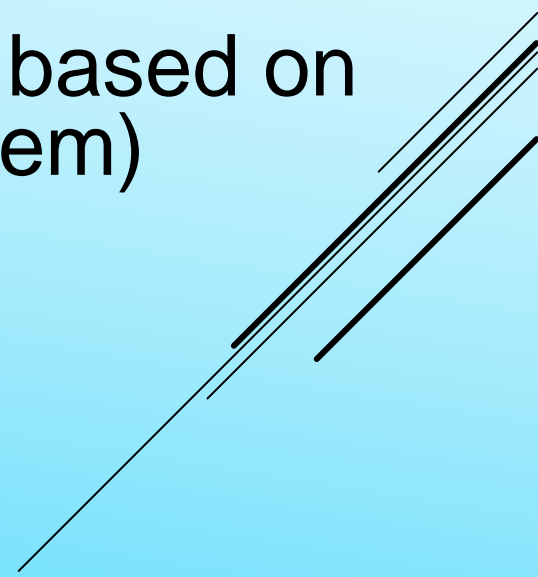
- Martin Fowler, Microsoft
 - Layout the application architecture into multiple layers.
 - Basic principles for linking domain logic and relational data.
 - Principles of Presentation Layer Design
- 

Association Table Mapping



Frameworks

- Java, .NET Framework,
- How do frameworks and patterns relate?
- By definition, the pattern is always re-implemented.
- Frameworks provide a solution based on patterns (if not, watch out for them)

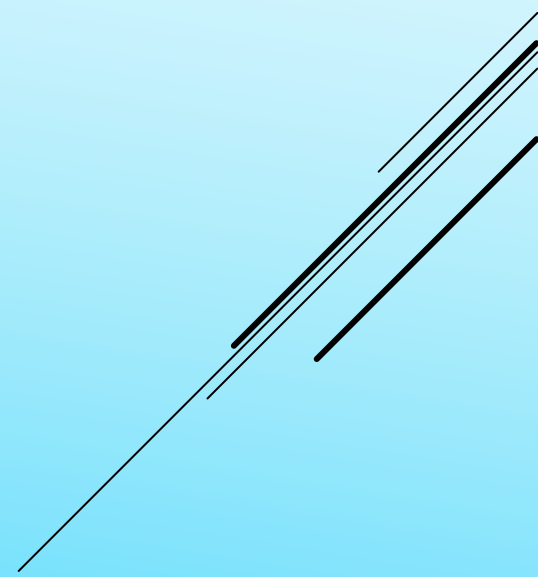


Domain Specific Language

Domain-specific language

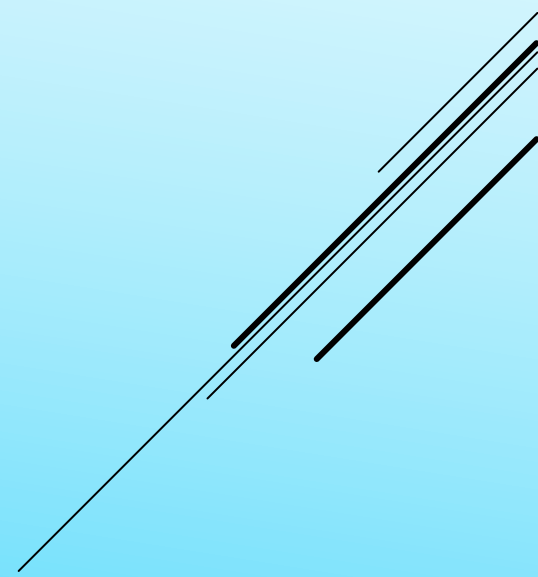
A computer programming language of limited expressiveness focused on a particular domain.

- Computer programming language
- Language nature
- Limited expressiveness
- Domain focus



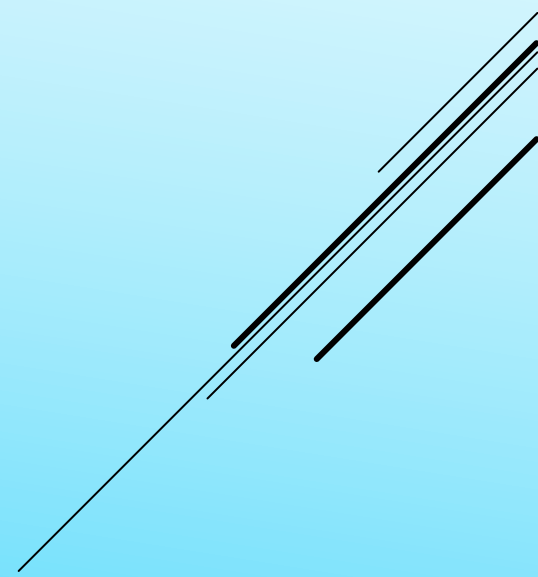
Life cycle of the information system

- Old versus new approaches
- Requirements, parameters
- Increments and iterations
- When it starts and when it ends
- How and why to document

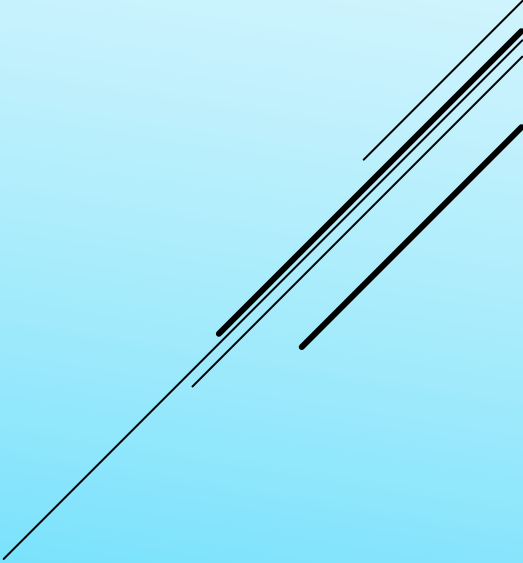


Methodology

- Unified process
- Robust x agile approach
- Focused on processes or people?
- RUP, SCRUM, TDD, EP

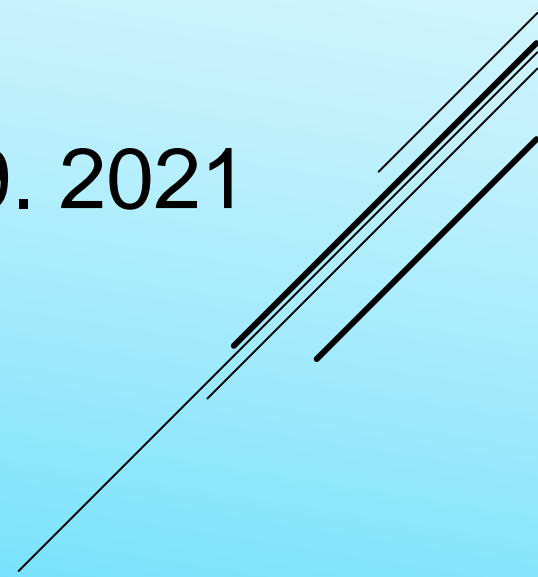


Information system - development

- Develop a new one?
 - Deploy existing ones?
 - Deploying an information system is rarely an isolated task.
 - How to convince the customer?
 - How not to upset the customer?
- 

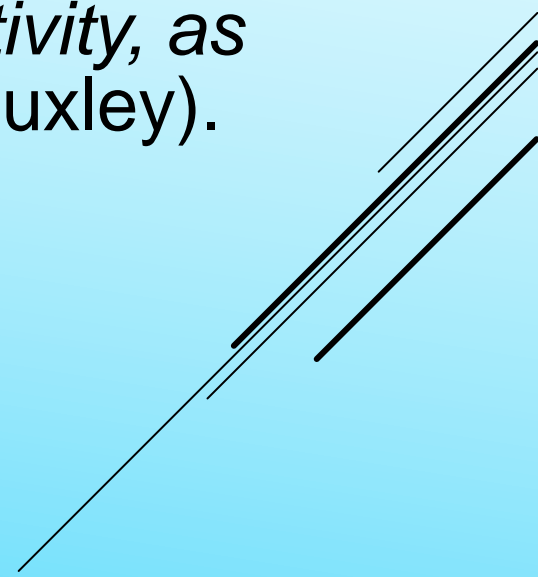
Exercises - design and implementation of IS

- Minimized range
- More complex architecture
- Limited use of technology
- Minimum documentation
- Revised vision of IS up to 29. 9. 2021

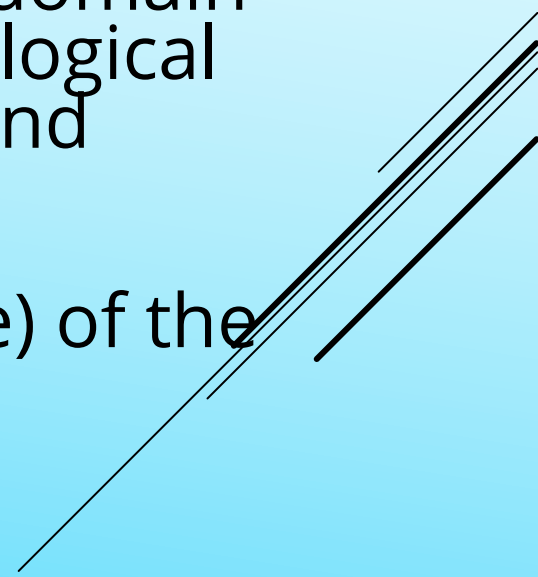


Requirements

- Seven artifacts (things) continuously consulted on exercises.
- The form and content of individual artefacts, will be continuously discussed.
- *An artifact refers to any object or process that has been created by human activity, as opposed to natural objects (Julian Huxley).*



Artifacts

- [2/1] Vision (document describing the system from the customer's point of view).
 - [7/4] Functional specification (use case model - description of individual cases, use case diagram, activity diagrams).
 - [3/2] Technical specification (first domain model, documentation for technological decisions, selected technologies and processes).
 - [2/1] Sketch (wireframe, prototype) of the user interface.
- 
- A series of three parallel diagonal lines in the bottom right corner of the slide, extending from the bottom right towards the center.

Artifacts II

- [7/4] Domain model design (classes, relationships, interactions - static class diagram, sequential diagram, patterns used).
- [3/2] Description of the system architecture (layout and interconnection of the logical and physical layers, diagram of components).
- [18/9] Consistent functional part of the selected information system with high emphasis on the architecture and design (layering, design in individual layers, patterns). It is assumed implementation of two simple user interfaces on different platforms, at least three non-trivial use cases in each of them and the use of two data storage methods (SQL database, XML etc.).

Exercise tasks

- Repetition of UML
- Project Assignment and Discussion
- Create code example for two classes with association relation between them.

