Software Design and Architecture

Uzair Rasheed

Why SDA?

- ❑ Did they have a good design?
- ❑ Could the design be done better?
- ❑ Was there even a design at all?
- ❑ How do you know if the software was well-designed? Think of how easy it was to make changes to your code.

Why SDA?

- ❑ Did a small code change produce a ripple-effect for changes elsewhere in the code?
- ❑ Was your code hard to reuse?
- ❑ Was the software difficult to maintain after a release?

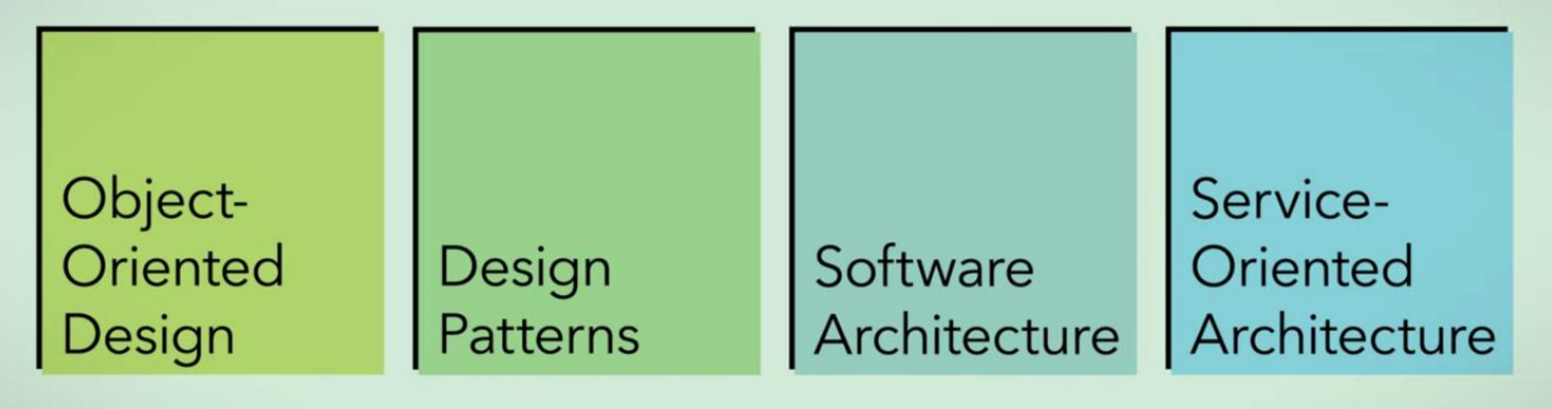
What is software Design and Architecture?

- ❑ How does it improve your software products?
- ❑ Consider this scenario.
  - ❑ You join a project that's been in development for a while. You look at the code and become instantly overwhelmed. You can't tell what the purpose of the pieces are, things are unorganized, and design documentation is non-existent. You don't even know where to begin. These are all signs that the project was not well-designed from the outset.

Course

- In this course, you will learn how to apply design principles, patterns, and architectures to create reusable and flexible software applications and systems.
- This course primarily covers software design and not user-interface design. Although the two aspects of design are closely related, we will mostly focus on software design.

Pillars of Software Design & Architecture



Object-Oriented Design

Design Patterns

Software Architecture

Service-Oriented Architecture

Object Oriented Modeling

- You will learn about object-oriented design, built upon the basics of Java and take you to the next level by covering object-oriented analysis and design.
- You will discover how to create flexible, reusable, and maintainable software by applying object oriented design principles.
- You will learn how to communicate these designs by expressing them in a visual notation known as Unified Modeling Language or UML.

Design Patterns

- Design issues and applications can be resolved through design patterns commonly applied by experts.
- This extends your knowledge of object-oriented analysis and design by covering design patterns used in interactive applications.
- Through a survey of established design patterns, you will gain a foundation for more complex software applications.

Software Architecture

- You will be equipped with software design patterns and principles and will be ready to learn how architecture can be used as a basis for organizing the software systems found in industry today.
- you will explore the structure and behavior of large scale software systems. Specific UML diagrams will express important architectural perspectives.
- You will learn how to analyze and evaluate a given architecture by examining the trade-offs between competing quality attributes such as modifiabilty and performance.

Service-
Oriented
Architecture

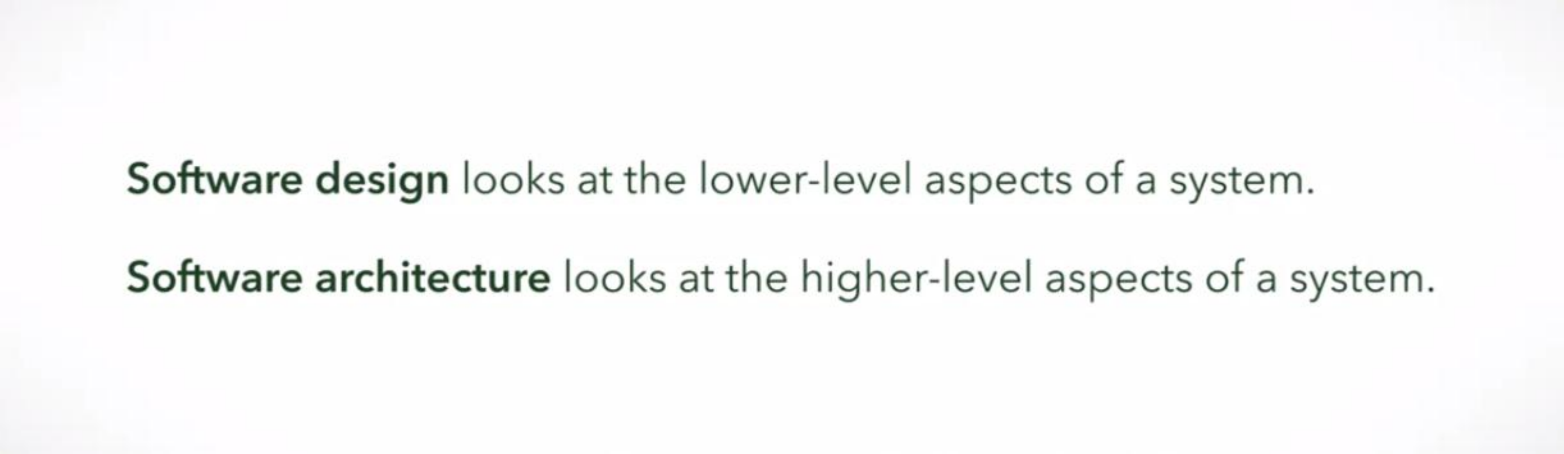
Service-oriented Architecture

- Based on the understanding of architectural styles that you learned; you will review architectures for web applications.
- You will then explore the basics of service-oriented architecture or SOA in two approaches, web services and representational state transfer or REST architecture.
- By the end of this course, you'll know how to design it right.

Software Design Vs Software Architecture

- ❑ What is the difference between software design and software architecture?
- ❑ Like many roles in the software industry, the software designer or a software architect role can look very different from company to company. Characteristics like company size, the scope of the project, the experience of the development team, the organizational structure and the age of the company can all impact what these roles look like.

Software Design Vs Software Architecture



Software design looks at the lower-level aspects of a system.

Software architecture looks at the higher-level aspects of a system.

Software Architect and Design Roles in Industry

- ❑ In some companies, there may be a distinct role for a software designer or architect.
- ❑ In other companies, the design may be completed by a member or members of the development team.
- ❑ Typically, the software designer role would be responsible for outlining a software solution to a specific problem by designing the details of individual components and their responsibilities.

Software Architect and Design Roles in Industry

- ❑ A software architect role would be responsible for looking at the entire system and choosing appropriate frameworks, data storage, solutions and determining how components interact with each other.
- ❑ That brings us to the primary difference between software design and software architecture.

Software Architect and Design Roles in Industry

- Think of this like designing a building. An architect focuses on the major structures and services, while an interior designer focuses on the smaller spaces within.
- Great software designers and architects are detail-oriented, forward thinkers. They need to be able to see the product at both the low and high levels.
- They need to be creative problem solvers in order to come up with a quality solution for the problem at hand and they need to be able to express these ideas effectively with the product manager and the development team.