SOFTWARE DESIGN AND ARCHITECTURE

INSTRUCTOR : MS. QUDSIA YOUSAF

CLASS : BSSE – SEMESTER V-A

QUDSIA.YOUSAF@CS.UOL.EDU.PK

2/11/2024

LECTURE NO. 2

Software Design and Architecture

KEY OBJECTIVES OF TODAY'S LECTURE

- The Design Process
- Software Architecture Design
- Role of Architect.
- Quality Attributes and Scenarios.

THE DESIGN PROCESS

- Study and understand the problem from different viewpoints
- Identify potential solutions and evaluate the tradeoffs
  - Design experience, reusable artifacts, simplicity of solutions
  - Sub-optimal, but familiar solutions often preferred advantages / disadvantages well known
  - Design is about making tradeoffs!
- Develop different models of system at different levels of abstraction and for different perspectives

THE DESIGN PROCESS

- Any design in any discipline can be viewed as a set of decisions : Example (Painting)
- Early design decisions embodied by software architecture are:
  - Will the system run on one processor or distributed across multiple processors?
  - Will components communicate synchronously or asynchronously? Will they interact by transferring control or data or both?
  - Will the system depend on specific features of the operating system or hardware?
  - Will the information flows through the system be encrypted or not?
  - What operating system will we use?
  - What communication protocol will we choose?

TOP-DOWN VS. BOTTOM-UP DESIGN

- Top-Down
  - Recursively partition problem into smaller sub-problems
  - Continue until tractable solutions found
  - Note: Not practical for large system in its pure form
- Bottom-Up
  - Assemble, adapt, and extend existing solutions to fit the problem
- In practice: A combination of both
  - Decompose large problems into smaller, but using previous design knowledge
  - Use existing components and solutions
  - Perhaps tackle problematic portions first

LIST OF DESIGN GOALS

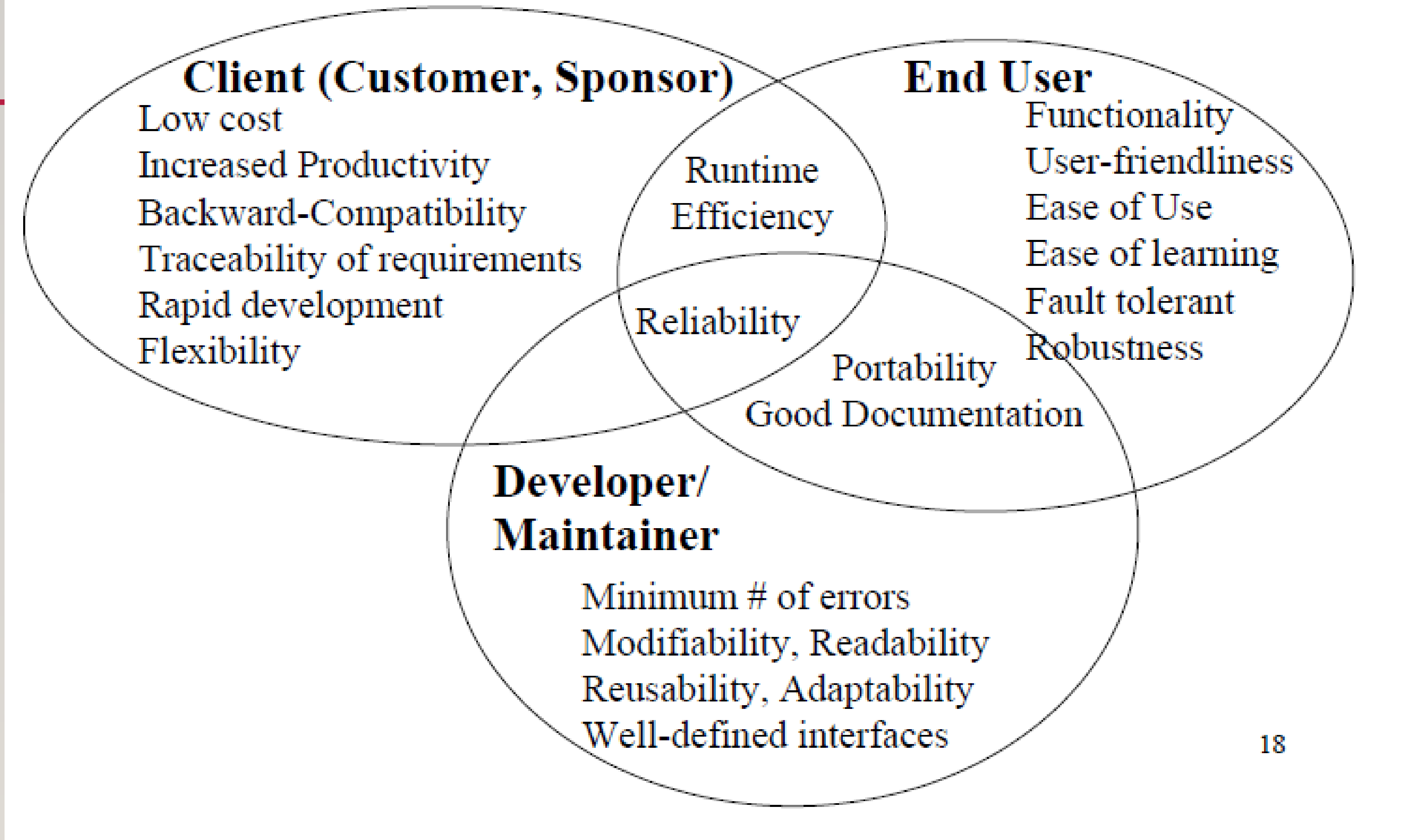
Reliability

- Modifiability
- Maintainability
- Understandability
- Adaptability
- Reusability
- Efficiency
- Portability
- Traceability of requirements
- Fault tolerance
- Backward-compatibility
- Cost-effectiveness
- Robustness
- High-performance

Good documentation

- Well-defined interfaces
- User-friendliness
- Reuse of components
- Rapid development
- Minimum # of errors
- Readability
- Ease of learning
- Ease of remembering
- Ease of use
- Increased productivity
- Low-cost
- Flexibility

RELATIONSHIP BETWEEN DESIGN GOALS



Client (Customer, Sponsor)

Low cost
Increased Productivity
Backward-Compatibility
Traceability of requirements
Rapid development
Flexibility

End User

Functionality
User-friendliness
Ease of Use
Ease of learning
Fault tolerant
Robustness

Runtime
Efficiency

Reliability

Portability
Good Documentation

**Developer/
Maintainer**

Minimum # of errors
Modifiability, Readability
Reusability, Adaptability
Well-defined interfaces

18

SOFTWARE ARCHITECTURE DESIGN

- Definition:

"The architecture design defines the relationship between major structural elements of the software, the styles and design patterns that can be used to achieve the requirements defined for the system, and the constraints that affect the way in which architecture can be implemented" (Garlan and Shaw, 1996)

- The architecture design representation is derived from the system requirement specification and the analysis model.

SOFTWARE ARCHITECTURE DESIGN

- Architecture serves as a blueprint for a system.
- It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components.
- It defines a structured solution to meet all the technical and operational requirements.
- Optimize the common quality attributes like performance and security.
- Involves a set of significant decisions about the organization related to software development

EVERY SYSTEM HAS A SOFTWARE ARCHITECTURE

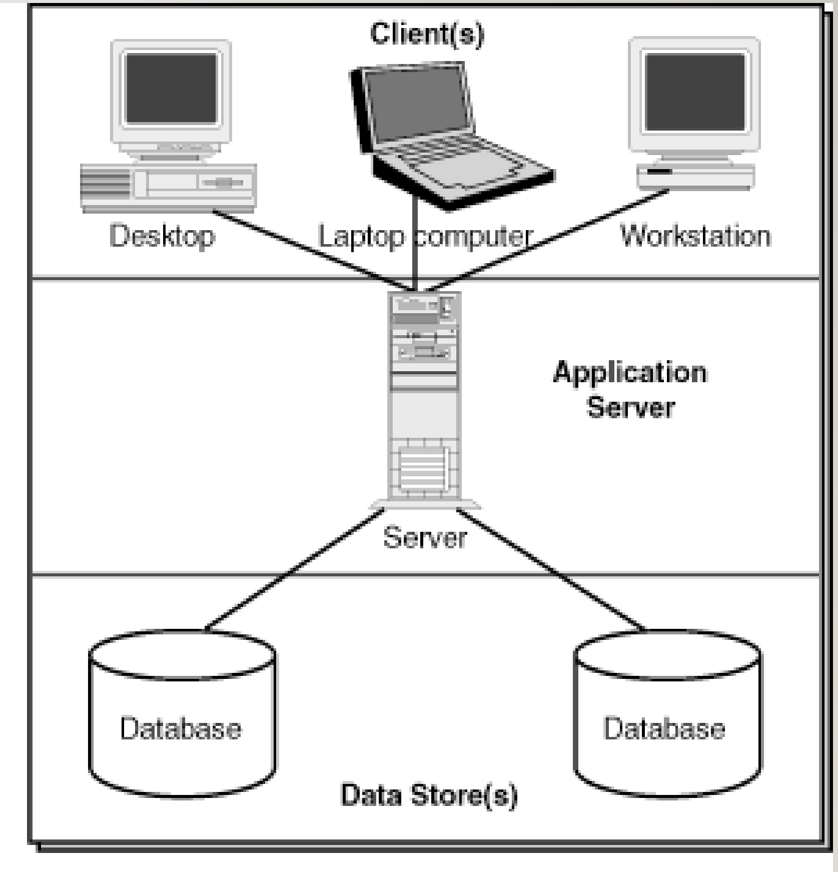
- Every system comprises elements and relations among them to support some type of reasoning.
- But the architecture may not be known to anyone.
  - Perhaps all of the people who designed the system are long gone
  - Perhaps the documentation has vanished (or was never produced)
  - Perhaps the source code has been lost (or was never delivered)
- An architecture can exist independently of its description or specification.
- Documentation is critical.

WHICH STRUCTURES ARE ARCHITECTURAL?

- A structure is architectural if it supports reasoning about the system and the system's properties.
- The reasoning should be about an attribute of the system that is important to some stakeholder.
- These include
  - functionality achieved by the system
  - the availability of the system in the face of faults
  - the difficulty of making specific changes to the system
  - the responsiveness of the system to user requests,
  - many others.

ARCHITECTURE EXAMPLE

- **Components** are such things as clients and servers, databases, layers, etc.
- **Interactions** among components can be procedure calls, shared variable access, etc.
- At the architectural level, we also consider system-level issues such as capacity, consistency, performance, etc.



IMPORTANCE OF SOFTWARE ARCHITECTURAL DESIGN

- A poor software design:
  - May result in a deficient product that does not meet system requirements.
  - Is not adaptive to future requirement changes, is not reusable, exhibits unpredictable behavior, or performs badly.
  - Without proper planning in the architecture design stage, software production may be very inefficient in terms of time and cost.

IMPORTANCE OF SOFTWARE ARCHITECTURAL DESIGN

- A good software design:
  - Reduces the risks associated with software production.
  - Helps development teams work together in an orderly fashion.
  - Makes the system traceable for implementation and testing.
  - Leads to software products that have higher quality attributes.

ROLE OF SOFTWARE ARCHITECT

- Software Architect provides a solution that the technical team can create and design for the entire application.

- Software architect should have expertise in the following areas:
  - Design Expertise
  - Domain Expertise
  - Technology Expertise
  - Methodological Expertise

ROLE OF SOFTWARE ARCHITECT

Design Expertise:

- Expert in software design, including diverse methods and approaches such as object-oriented design, event-driven design, etc.

- Lead the development team and coordinate the development efforts for the integrity of the design.

- Should be able to review design proposals and tradeoff among themselves.

ROLE OF SOFTWARE ARCHITECT

Domain Expertise :

- Expert on the system being developed or planned for software evolution.
- Assist in the requirement investigation process, assuring completeness and consistency.
- Coordinate the definition of domain model for the system being developed.

ROLE OF SOFTWARE ARCHITECT

Technology Expertise :

- Expert on available technologies that help in the implementation of the system.
- Coordinate the selection of programming language, framework, platforms, databases, etc.

ROLE OF SOFTWARE ARCHITECT

Methodological Expertise:

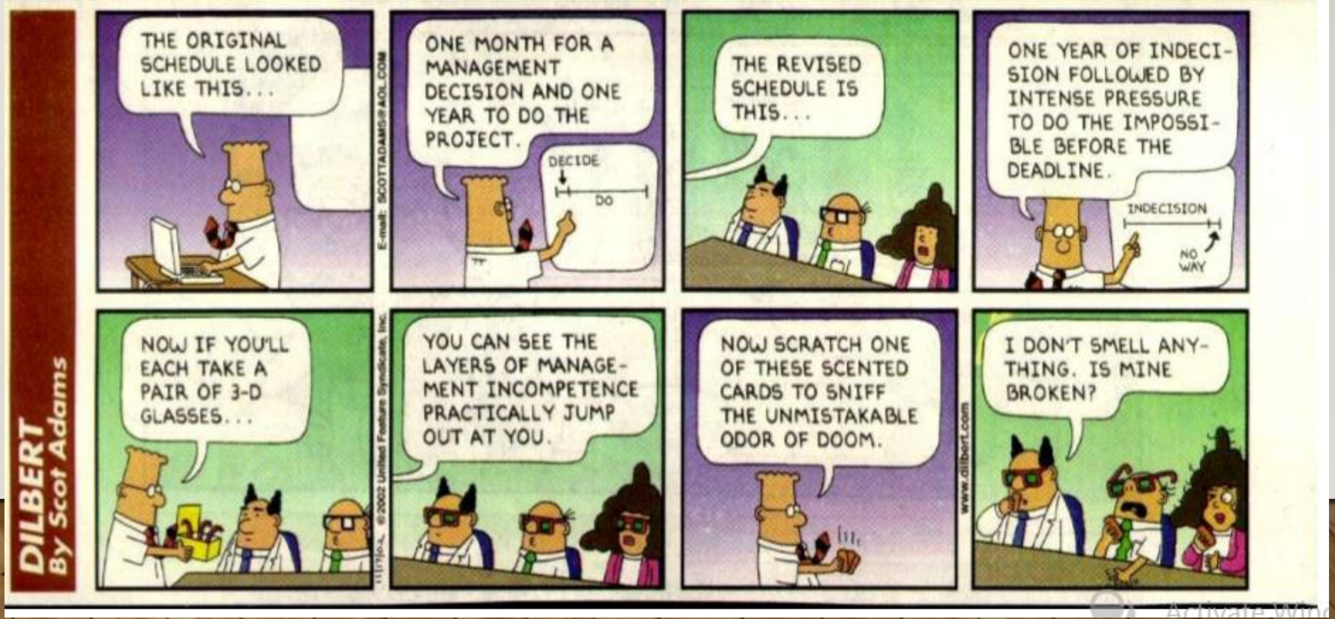
- Expert on software development methodologies that may be adopted during SDLC (Software Development Life Cycle).
- Choose the appropriate approaches for development that help the entire team.

DELIVERABLES OF THE ARCHITECT

- Architect is expected to deliver clear, complete, consistent, and achievable set of functional goals to the organization.
- Also provides:
  - A simplified concept of the system
  - A design in the form of the system, with at least two layers of decomposition.
  - A functional description of the system, with at least two layers of decomposition.
  - A notion of the timing, operator attributes, and the implementation and operation plans.
  - A document or process which ensures functional decomposition is followed, and the form of interfaces is controlled.

GETTING THE ARCHITECTURE RIGHT IS HARD!

"..The life of a software architect is a long (and sometimes painful) succession of sub-optimal decisions made partly in the dark…" (Philippe Kruchten)


DILBERT By Scot Adams
THE ORIGINAL SCHEDULE LOOKED LIKE THIS...
ONE MONTH FOR A MANAGEMENT DECISION AND ONE YEAR TO DO THE PROJECT.
DECIDE
DO
E-mail: SCOTTADAMS@AOL.COM
THE REVISED SCHEDULE IS THIS...
ONE YEAR OF INDECISION FOLLOWED BY INTENSE PRESSURE TO DO THE IMPOSSIBLE BEFORE THE DEADLINE.
INDECISION
NO WAY
NOW IF YOU'LL EACH TAKE A PAIR OF 3-D GLASSES...
©2002 United Feature Syndicate, Inc.
YOU CAN SEE THE LAYERS OF MANAGEMENT INCOMPETENCE PRACTICALLY JUMP OUT AT YOU.
NOW SCRATCH ONE OF THESE SCENTED CARDS TO SNIFF THE UNMISTAKABLE ODOR OF DOOM.
www.dilbert.com
I DON'T SMELL ANYTHING. IS MINE BROKEN?

EXAMPLE

- The user is concerned that the system is fast, reliable, and available when needed.
- The customer is concerned that the architecture can be implemented on schedule and budget.
- The manage is worried (in addition to schedule and budget) that the architecture will allow teams to work largely independently, interacting in disciplined and controlled way.
- The architect is worried about strategies to achieve all of those goals.

DUTIES OF A SOFTWARE ARCHITECT

- Reinforce the trust relationship among team members.
- Protect team members from the external forces that could distract them and bring less value to the project.
- Perform tradeoff analysis on quality attributes and other nonfunctional requirements during the selection of architecture styles.

THANK YOU !