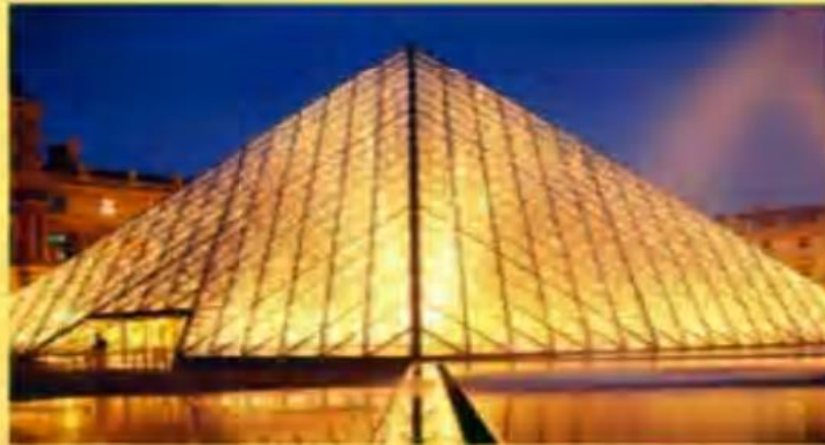


SEI SERIES IN SOFTWARE ENGINEERING

Software Architecture in Practice

Second Edition



Len Bass

Paul Clements

Rick Kazman

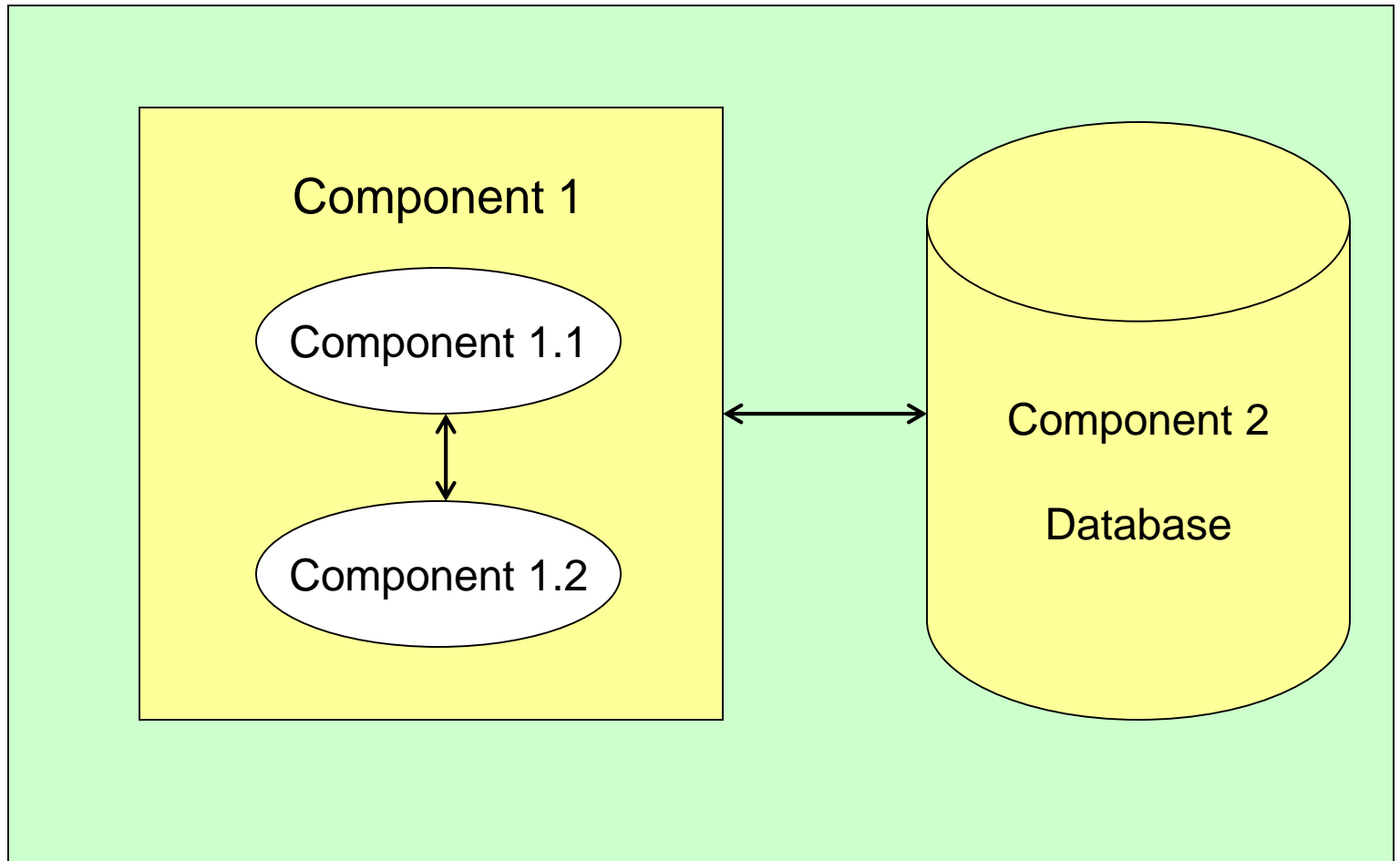
Introduction

Informal Definition of SA

- First step in developing the solution
- Overall (high level) structure of the software system
- Software architecture = Components + Connectors

What are components and connectors?

A simple example



Why Software Architecture?

- Complexity → Divide and Conquer
 - Process: Divide design process to phases
 1. Architectural design
 2. Detailed design
 - Product: Decompose system to components
- Assuring fulfillment of required quality attributes (performance, changeability, etc) from the beginning

Roots of Software Architecture

- Software architecture is similar to building architecture in many ways.
- The idea is not new. Concepts related to SA have been in the literature since 60's and 70's (e.g. modularity, info. hiding).
- However, the term is new.
- During the past 10 years SA have received considerable attention and have been subject to many research projects.

Architecture Business Cycle

Motivation

- We add a new role to software development team: Software Architect
- What does software architect do? Simply drawing the some diagrams
- What else is related to SA?
- Are 2 SAs developed in different environmental conditions for a single system the same?

This part covers two issues:

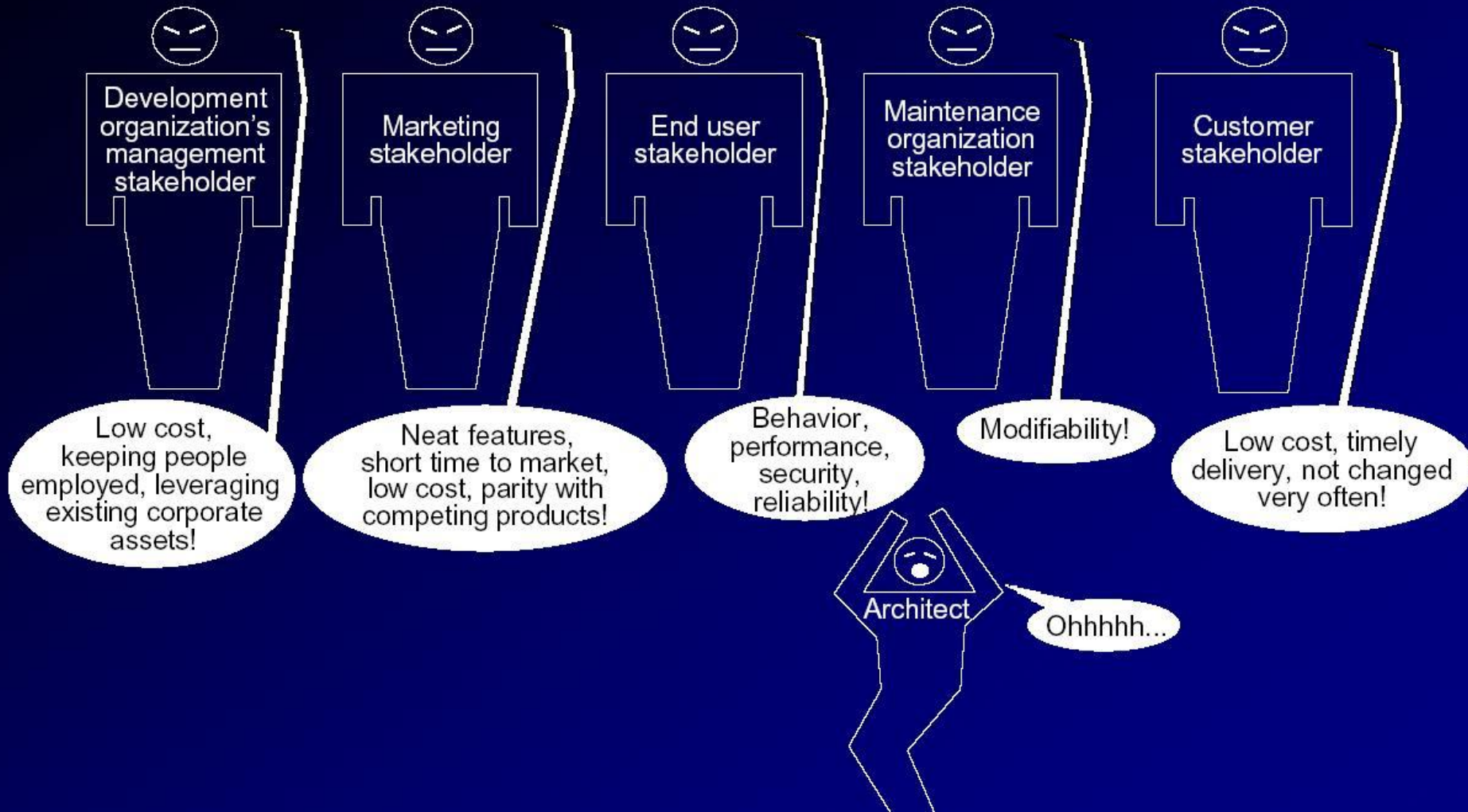
- What influences software architecture?
- What are influenced by software architecture?

Architectural Influences

- Stakeholders
 - each stakeholder has different concerns & goals, some contradictory
- Development Organization
 - immediate business, long-term business, and organizational (staff skills, schedule, & budget)
- Background & Experience of the Architects
 - repeat good results, avoid duplicating disasters
- The Technical Environment
 - standard industry practices or common SE techniques

Who influences SA?

Stakeholders of a System



Customers and End Users

- Requirements (including qualities such as performance, maintainability, etc)
- Budget Limitation
- Time Limitation
- Force to apply specific technology, methodology, or organizational discipline

Developing Organization Concerns

Business issues

- investing in, and then amortizing the infrastructure (domain analysis rather than application analysis)
- keeping cost low
- simplicity of implementation

Organizational issues

- using the current organizational structure
- utilizing personnel

Technical Environment

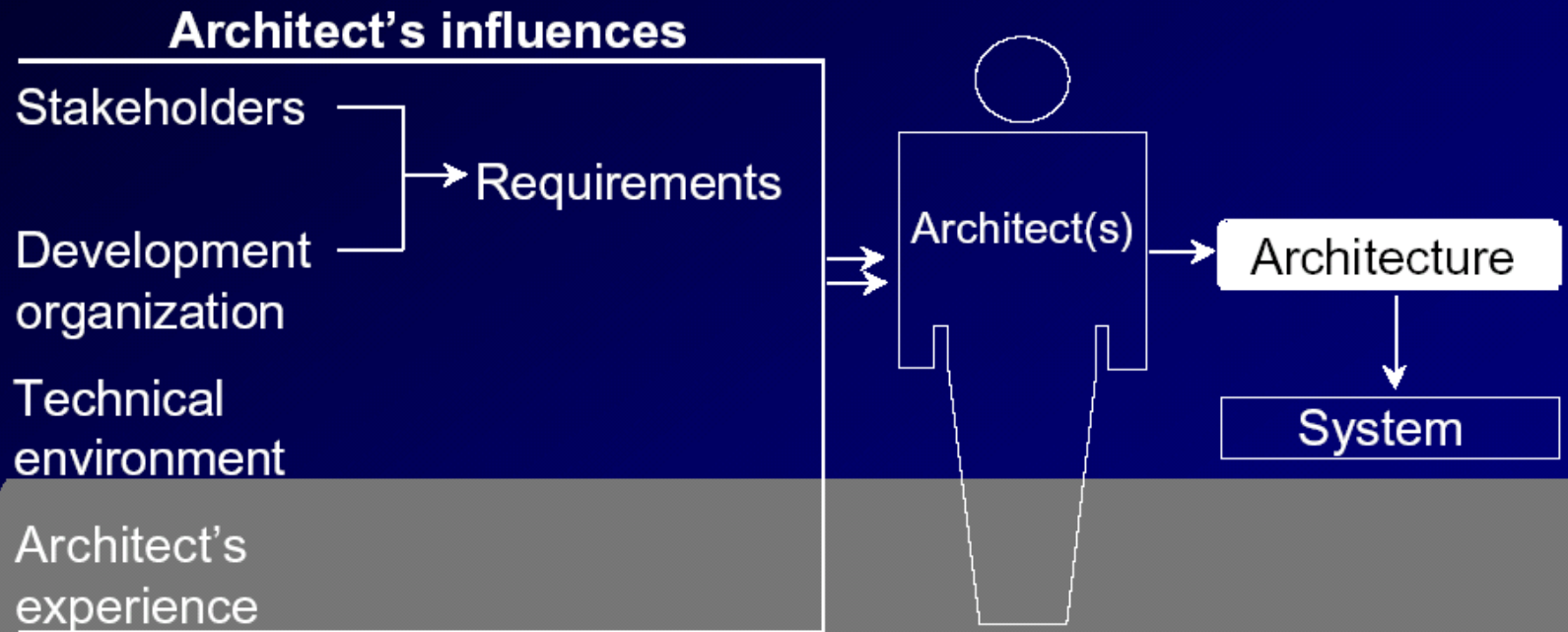
- Current trends: today's information system are web-based and use middleware systems (e.g. J2EE, .Net)
- Available technology: decisions on using a centralized or decentralized system depend on processor cost and communication speed; both are changing quantities.

Architect's Background

Architects develop their mindset from their past experiences.

- Prior good experiences will lead to replication of prior designs.
- Prior bad experiences will be avoided in the new design.

Summary: Influences on the Architect



Architecture Influences the Development Organization

- Organizational Structure and Recourses
 - Work units are organized around architectural units
 - Schedule
 - Budget
- Enterprise Goals
 - Expertise in building a kind of systems
 - Success in a market
 - Evaluating a market

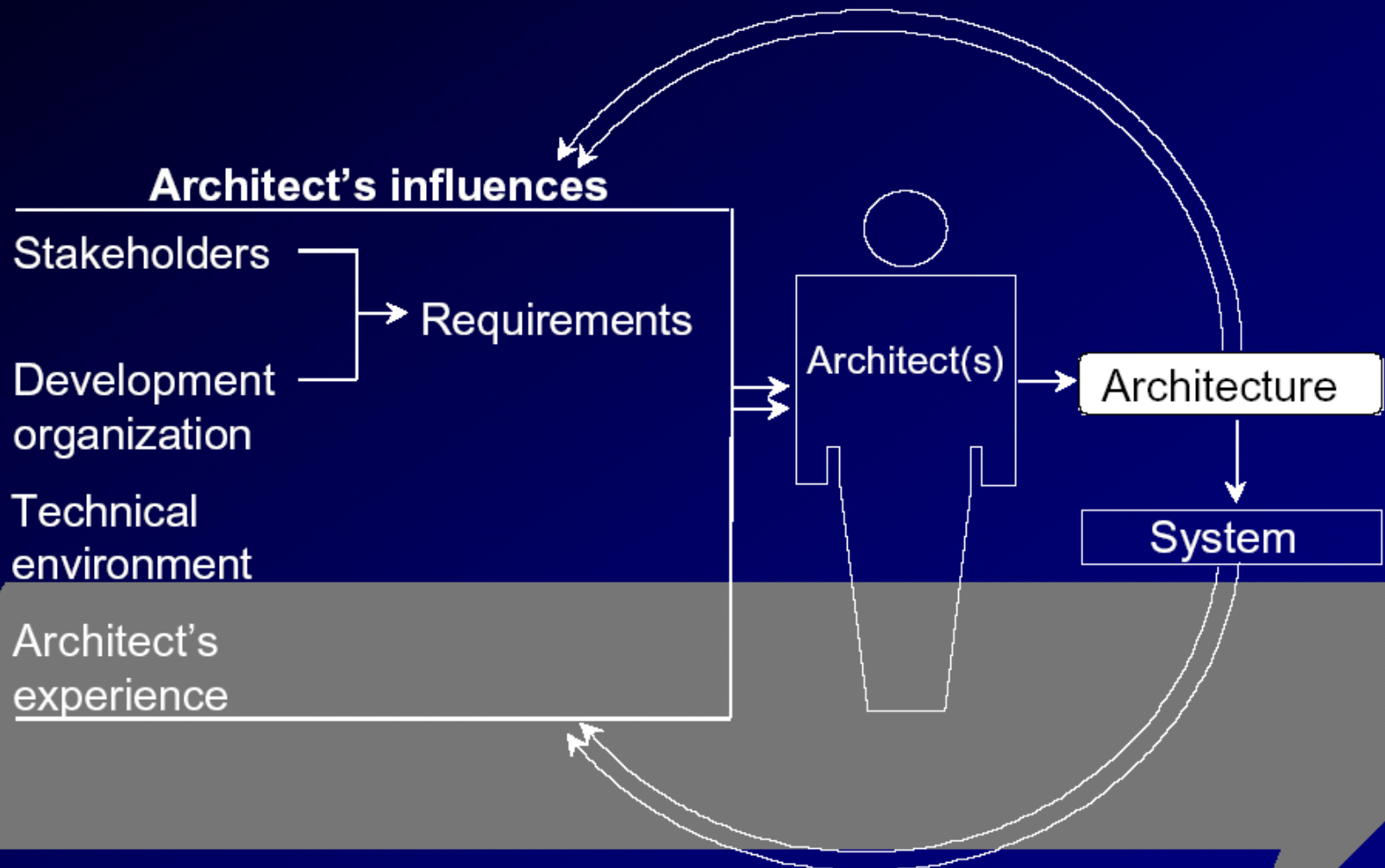
Architecture Influences Customer Requirements

- Knowledge of customers to ask for particular features in next systems.
- Support of upgrade, adaptation, etc.

Architecture Influences the Architect's Experience and Technical Environment

- Creation of a system affects the architect's background.
- Occasionally, a system or an architecture will affect the technical environment.

Architecture Business Cycle



Process Steps in Architecture-Based Development

- Creating the business case for the system
- Understanding the requirements
- Creating, customizing, or selecting the architecture
- Representing and communicating the architecture
- Analyzing or evaluating the architecture
- Implementing based on architecture
- Ensuring conformance

What makes a Good Architecture?

- No such thing as an inherently good or bad architecture.
- Architectures are more or less fit for some stated purpose.
- Architectures can be evaluated - one of the great benefits of paying attention to them - but *only in the context of specific goals*.
- Rules of Thumb: *process & product (structural) recommendations*

Rules of Thumb

- Process Recommendations:
 - include functional requirements and a prioritized list of quality attributes the system must satisfy
 - analyze & formally evaluate *before it is too late* to change
- Product Recommendations:
 - well-defined modules using principles of information hiding & separation of concerns
 - separate modules that produce data from those that consume data to increase modifiability & staged upgrades
 - write tasks or processes to allow easy reallocation, perhaps at runtime