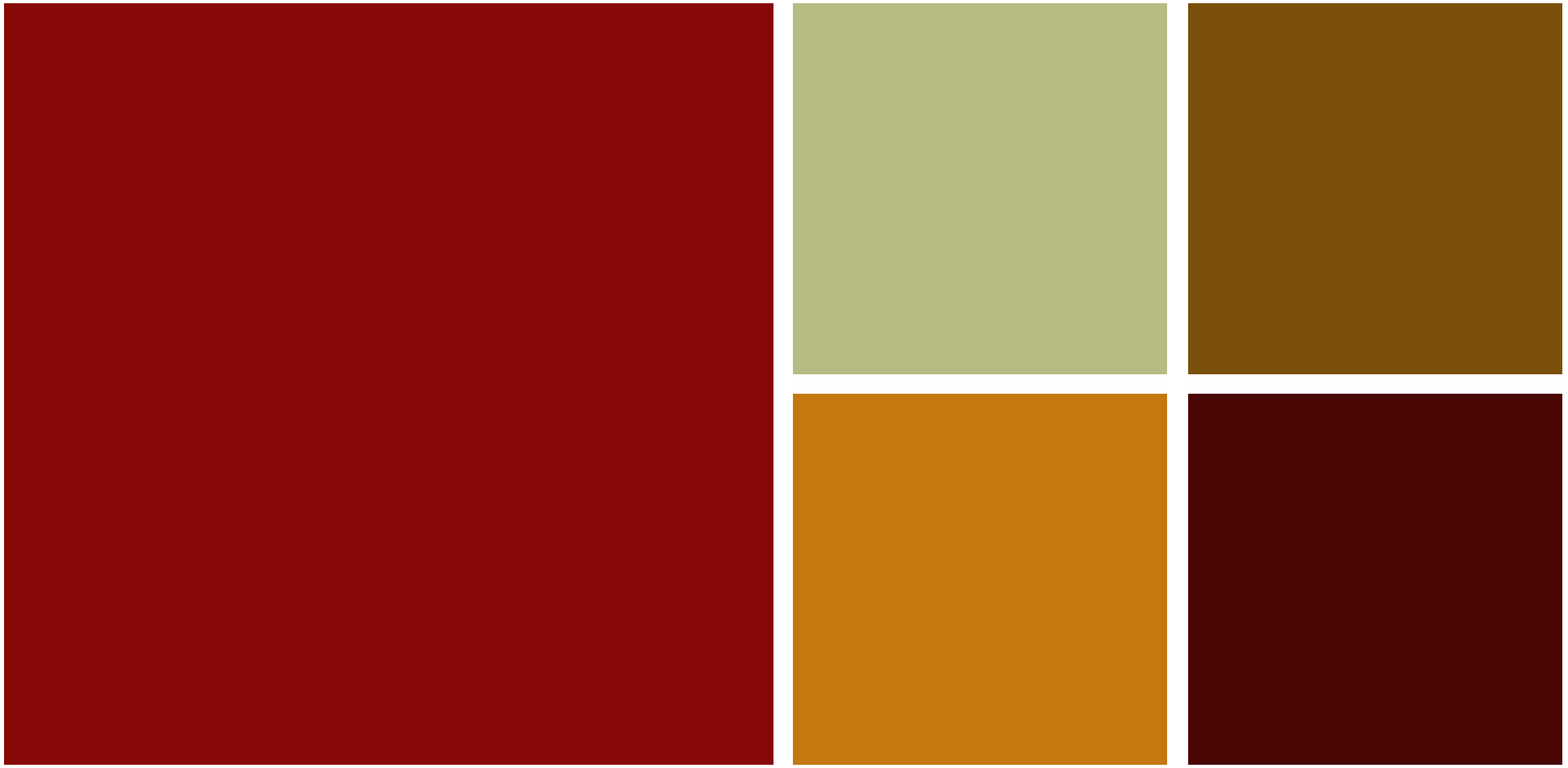


Software Architecture in Action

Flavio Oquendo, Jair C Leite, Thais Batista

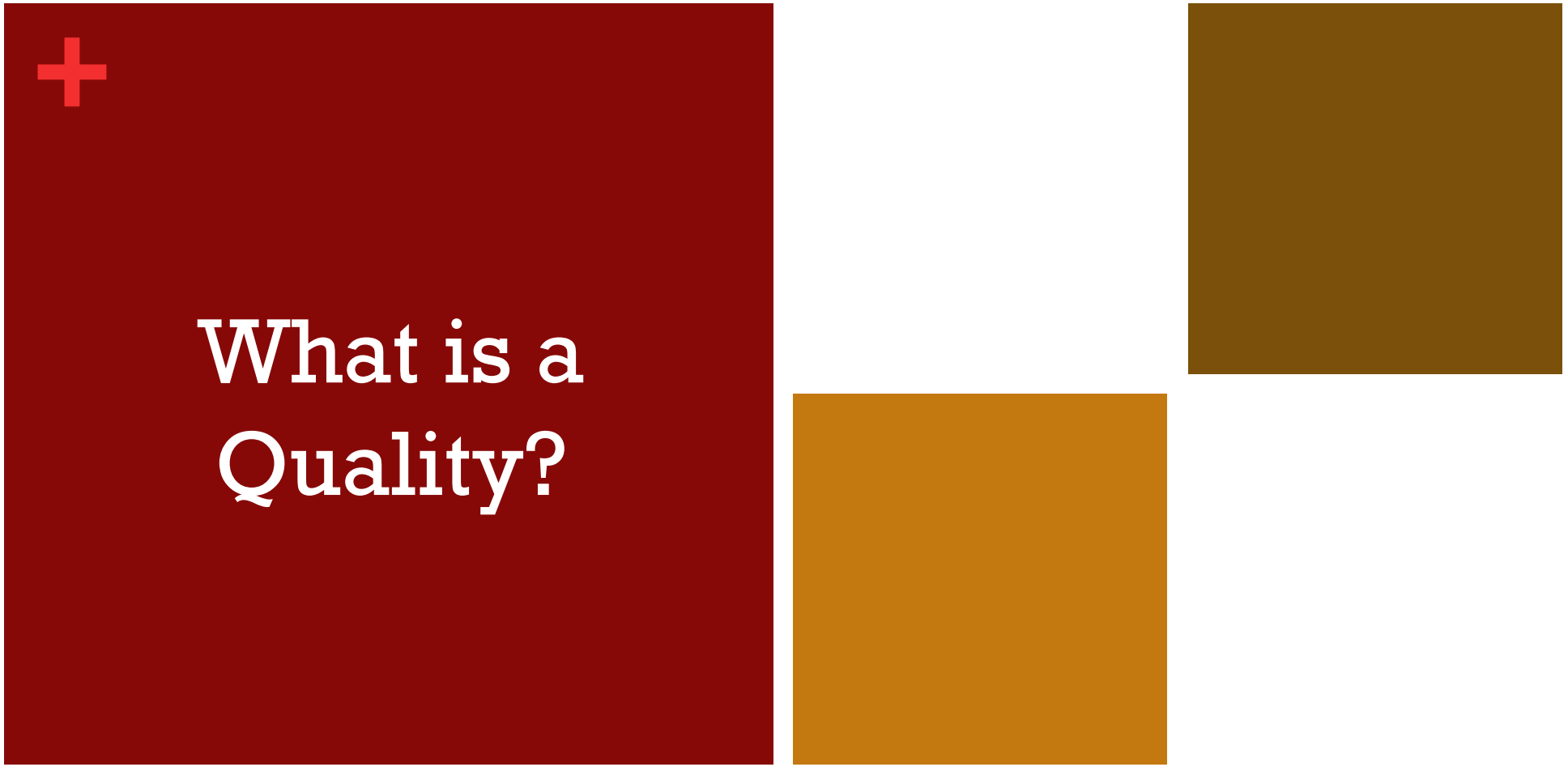
Part II
Quality-based architecture
Chapter 8
Introduction

Learning outcomes of this chapter

- You will learn:
  - what is quality
  - what is quality-based architectures
  - quality attributes that are significant to software architecture
  - an introduction to three quality attributes
- You will know how to:
  - associate design decisions to quality attributes
  - use SysADL to represent quality attributes
  - use SysADL to represent the relationship between design decisions and quality attributes

The structure of this chapter

- What is a Quality
- What is Quality-based Architectures
- Analysing Quality-based Architectures
- Quality Attributes: Modifiability, Scalability, Fault Tolerance
- Summary


What is a
Quality?

Quality

Conceptual overview

- A quality is a **non-functional requirement** that a system must satisfy.
  - More precisely, the quality of a system is **the degree to which the system satisfies the stated and implied needs** of its various stakeholders, and thus provides value [ISO 25010].
- Several architectures can be designed to satisfy functional requirements. However, **these architectures may vary in the degree in which the non-functional requirements are satisfied.**
  - For instance, **one** of these architectures can be **easily modified** and **another** could have a **high cost to allow modifications**.

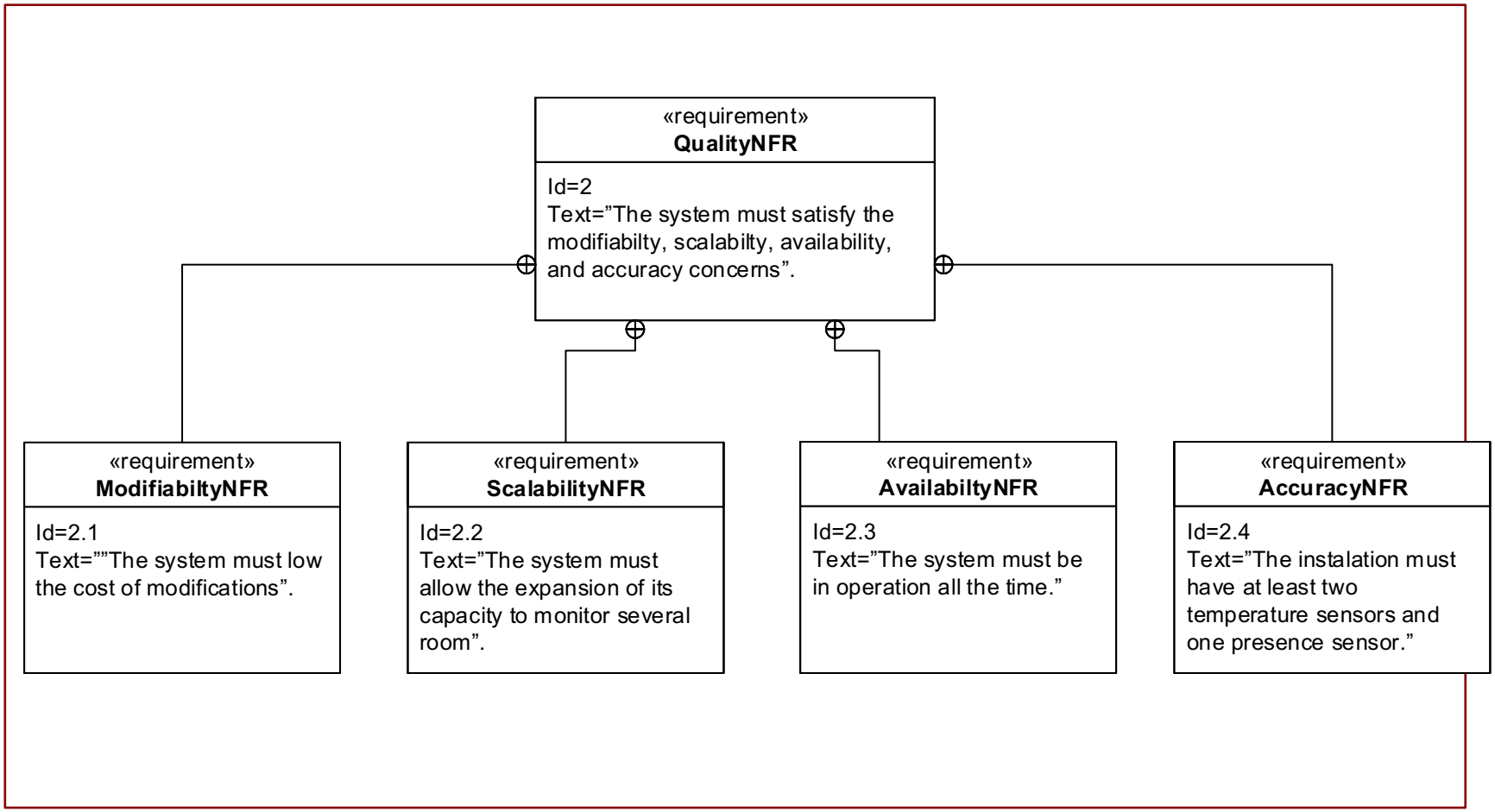
Quality

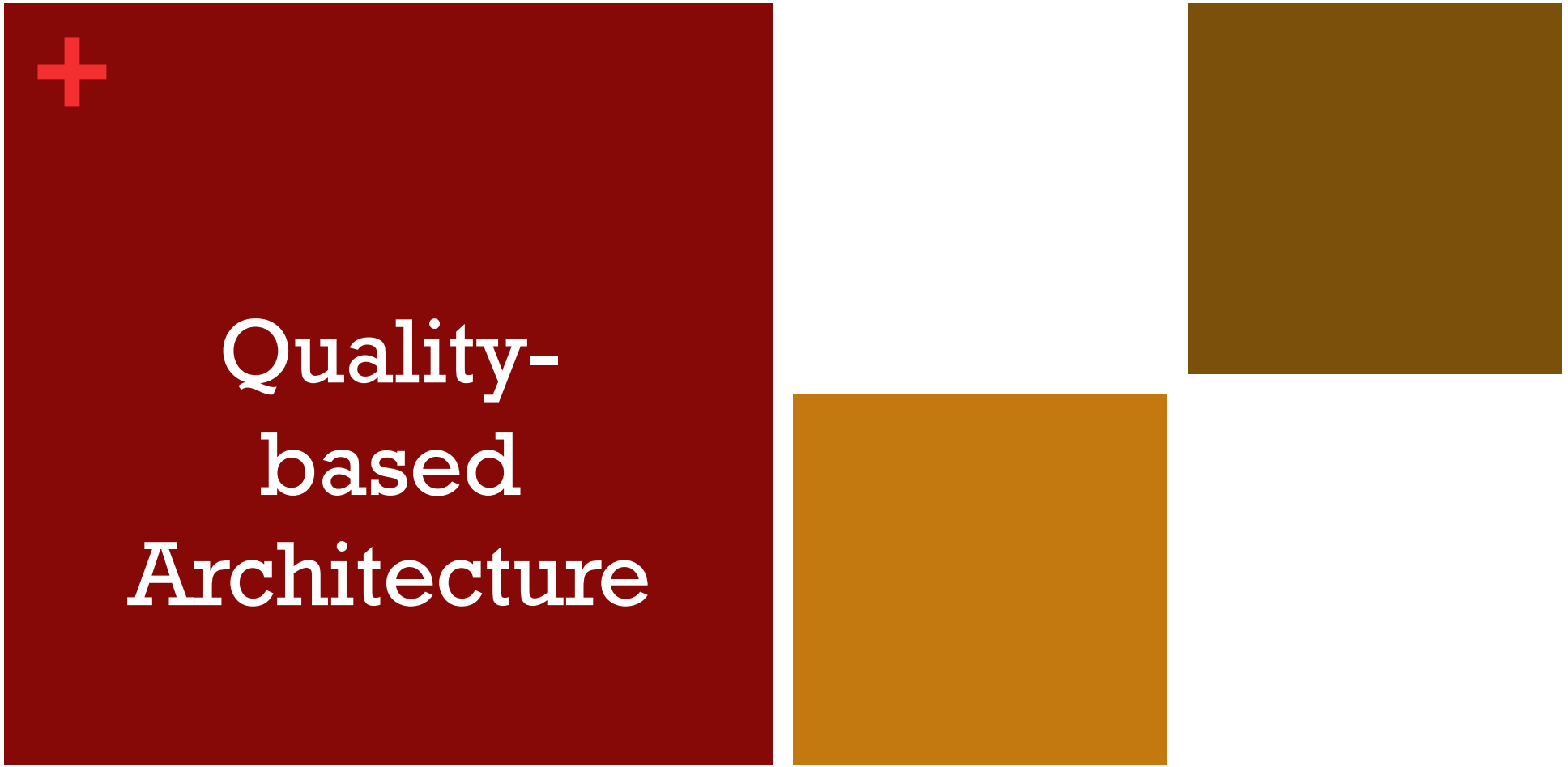
Example (1/2)

- In our running case study, the *RTC system* has the **following non-functional requirements**, which quality attributes:
  - the architecture must **minimize the cost of modifications**;
  - the architecture must **allow the expansion of its capacity** to monitor several rooms; and
  - the system must be **in operation all the time**.

Quality

Example (2/2)

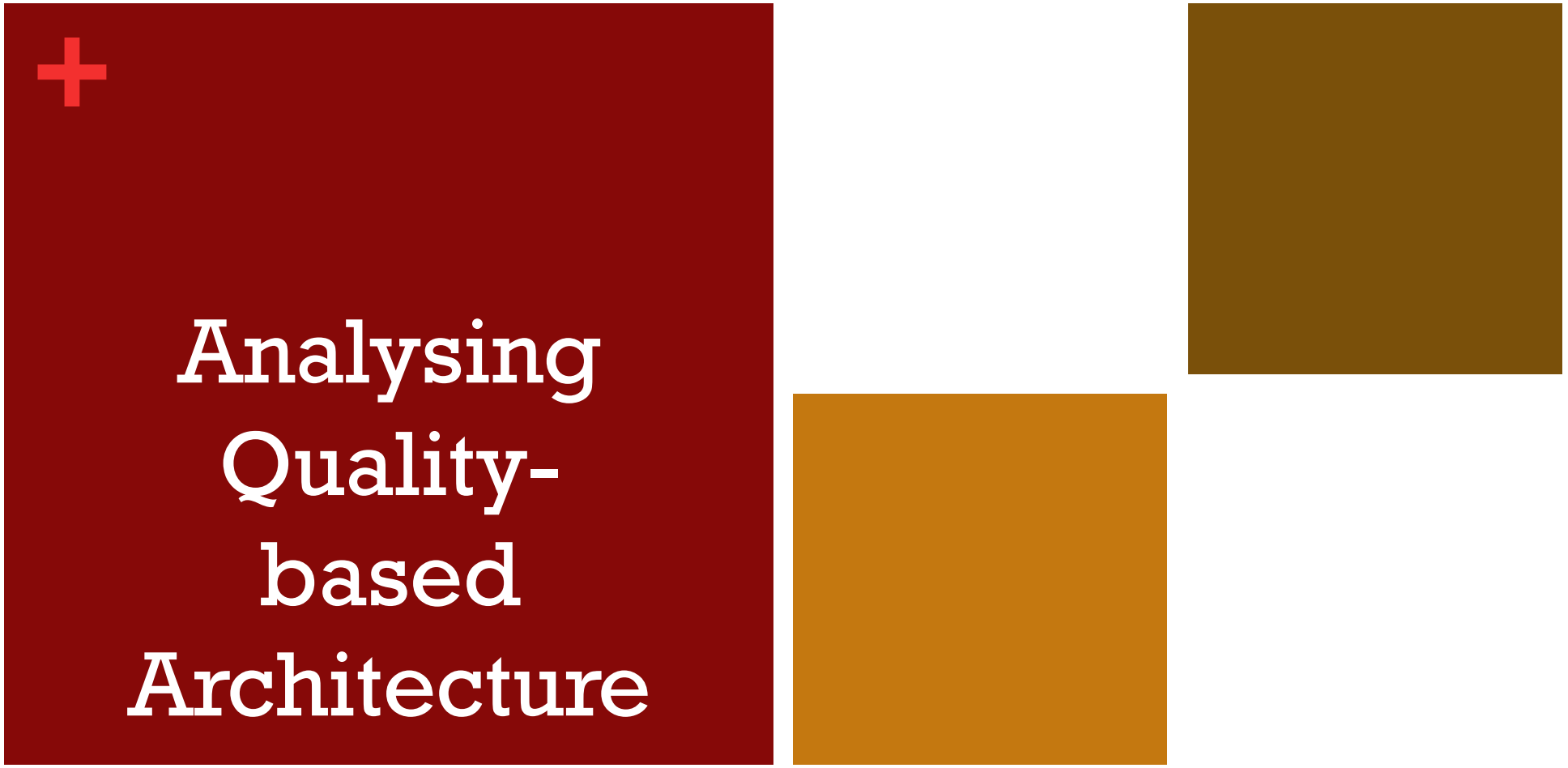



Quality-based Architecture

Quality-based Architecture

Conceptual Overview

- A quality-based architecture is one **designed to satisfy a quality attribute or a set of quality attributes**.
- In most cases, **it is impossible to maximize all of them** and the architect should consider a trade-off.
  - An architecture that must satisfy both security and performance requirements.
    - An architecture to increase security must add architectural elements to implement the security policies. For instance, adding an architectural element to encrypt information that is transmitted between components. That architecture decision implies in performance degradation.
    - For addressing the trade-off, the architect should design an architecture that both security and performance has acceptable values.


Analysing
Quality-
based
Architecture

Analysing Quality-based Architecture

Conceptual overview

- The analysis of quality-based architectures involves the following activities:
  - To derive an **architectural requirement** for each non-functional requirement related to quality attribute;
  - To define the **causes** for each architectural requirement;
  - To analyse the **ripple effects** in different architectures for the same cause.
- Quality attributes can be defined as a **cause-effect relationship**, so we need to express both the causes and the effects.

Analysing Quality-based Architecture

Example

- For instance, considering the **quality attribute of modifiability** applied to our *RTC System*
  - one possible architectural requirement is that **the system must allow the modification of the temperature sensors** to improve temperature measures.
- Two possible modification causes related to this requirement are:
  - (C1) **to add a new temperature sensor** of a defined type to improve accuracy, or
  - (C2) **to replace a temperature sensor** for another one of a new type, e.g. to support other temperature units.
  - A lot of other causes can also be included.
- In terms of **ripple effect** associated to the C1 cause includes:
  - **to replace the room temperature controller** by another that is a specialization of the previous one with a new temperature port.

Quality
Attributes

Quality Attributes

Conceptual Overview (1/3)

- This Part II focuses on discussing three quality attributes: Modifiability, Scalability, and Fault Tolerance.
- ***Modifiability*** refers to the degree to which a system can be effectively or efficiently modified without introducing defects or degrading existing system quality.
- In Chapter 9
  - we discuss **architectural requirements involved in the modifiability for the *RTC System*** with the causes and effects on other architectural elements (ripple effect).
  - we present **an example of modifiability** where a Fahrenheit temperature sensor is replaced by a Celsius temperature sensor.
  - we also analyse **the cost of modifying two architectures of the *RTC system*,** i.e. the number of additional architectural elements that need to be modified when a given modification is needed (ripple effects).

Quality Attributes

Conceptual Overview (2/3)

- ***Scalability*** refers to the degree to which a system can effectively and efficiently be scaled for increasing operational usage.

- In Chapter 10
  - we discuss **architectural requirements involved in scalability**, with the causes and effects on other architectural elements.
  - we explain **how an architecture of the *RTC system* can be scaled** to cope with an increasing number of component usage that may be of several orders of magnitude.

Quality Attributes

Conceptual Overview (3/3)

- ***Fault tolerance*** refers to the degree to which a system can maintain operational service even in the presence of faults.
- In Chapter 11
  - we discuss **architectural requirements involved in fault tolerance**, with the causes and effects on other architectural elements.
  - we will **analyse the fault tolerance on two different architectures of the *RTC system*.**

Summary

- In this chapter you learnt
  - the quality concept;
  - The quality-based architecture concept;
  - the activities to analyse quality-based architectures.