CptS 487
Software Design and Architecture

Lesson 18 (pt1)

Quality Attributes 1

Introduction

WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Instructor: Bolong Zeng

Before you proceed

- Read [BASS] Chapter 4

Outline

- Intro to the concept of Quality Attributes (QA)
  - Part II of the [BASS] book
- [BASS] presents 7 major QAs with details
  - Availability
  - Interoperability
  - Modifiability
  - Performance
  - Security
  - Testability
  - Usability
  - and how to extend upon other QAs.

What is a Quality Attribute (QA)

- QA is a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders.
  - Stakeholders: “people who have interests in the success of the system”
    - User, Manager, Developer, Client, HR, Marketing, Graphic, Story, Tester, Maintenance, etc.…
    - Keep in mind, different stakeholders will have different priorities, focus, etc…

Addressing requirements w/ Architecture

- *Functional Requirements?*
  - Assigning proper sequence of responsibilities throughout the design.
- *QA Requirements?*
  - Structures of the architecture, and behaviors and interactions of the elements that populate the structures.
- *Constraints (non-functional requirements)?*
  - Accepting the design decisions, and reconciling and compromising.

Challenges w/ Discussing QA

- 1. Definition alone is not testable.
  - Saying “the system will be *modifiable*” is pointless.
- 2. Pin-point a concern to one specific QA is hard and unreasonable.
  - A system failure under DDOS: availability/security/performance/…
- 3. Competing vocabulary from different attribute community.
- [BASS]’s solution:
  - Quality Attribute Scenarios
  - Discuss fundamental concepts for each attribute community.

QA Context

- Runtime and Development QA
  - Availability, performance, usability
  - Modifiability, testability.
- QA should not be isolated.
  - Sometimes they compete with each other too.
- We’ll start with learning the context for these QAs.
  - How to specify a QA
  - What architectural decisions enable a QA
  - Checklist of questions to make the correct decisions.

Specifying QA Requirements

- Six part scenarios:
  - Source of stimulus
  - Stimulus
  - Environment
  - Artifact
  - Response
  - Response measure

- General vs Specific scenarios
  - System independent vs. System specific

Tactics

- Tactics to achieve QAs
  - Focus on a single QA, holding off considerations of tradeoffs.

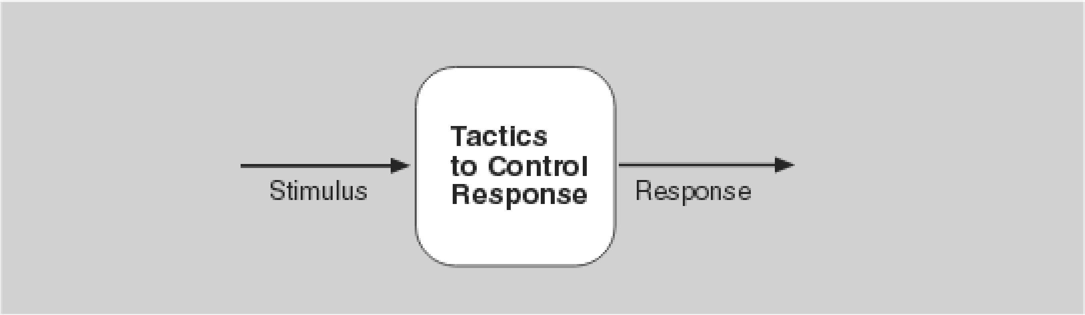

FIGURE 4.3 Tactics are intended to control responses to stimuli.

  - The end of the book features a collection of tactics to address each QA specifically

Checklists for Design Decisions

- Allocation of Responsibilities
- Coordination model
- Data model
- Management of resources
- Mapping among architectural elements
- Binding time decisions
- Choice of technology

In this lesson…

- We’ll cover 4 QAs specified in [BASS]
  - Availability
  - Performance
  - Security
  - Testability