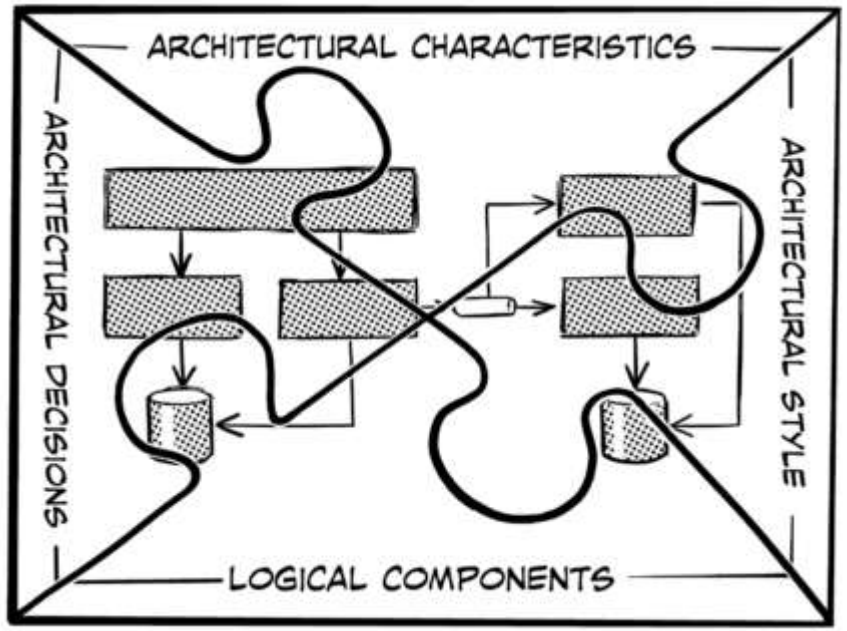
1. Funciones del Arquitecto, FOSA
2. Atributos de Calidad, FOSA y SEI.
3. Tacticas para Disponibilidad (Availability), Rendimiento (Performance) y Despliegue (Deployability). SEI.
4. SLA, SLI, SLO o KPI. SEI y otros.
5. Estilo vs Diseño, FOSA
6. Clasificacion de Estilos: por Despliegue: Monolitos vs Sistemas Distribuidos. SAP Report.
7. Listado de Estilos. FoSA, Azure
8. Listado de Patrones seleccionados. SAP Report, Azure



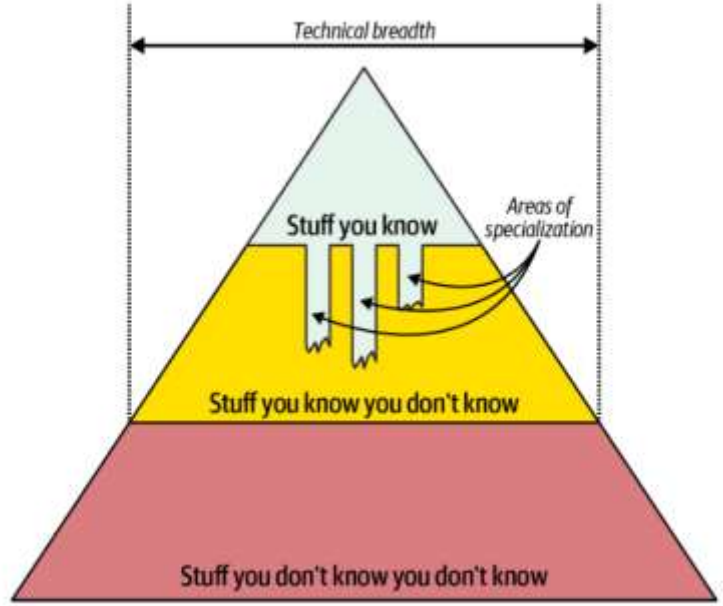


Figure 2-5. Enhanced breadth and shrinking depth for the architect role

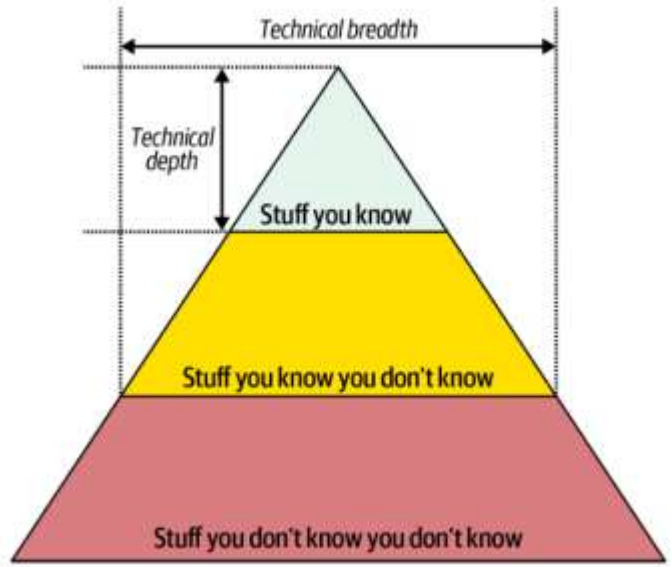


Figure 2-4. How much someone knows about a topic is technical depth, and how many topics someone knows is technical breadth

Everything in software architecture is a trade-off.

—First Law of Software Architecture

Why is more important than how.

—Second Law of Software Architecture

Most architecture decisions aren't binary but rather exist on a spectrum between extremes.

—Third Law of Software Architecture

- Make architecture decisions
- Continually analyze the architecture
- Keep current with latest trends
- Ensure compliance with decisions
- Understand diverse technologies, frameworks, platforms, and environments
- Know the business domain
- Lead a team and possess interpersonal skills
- Understand and navigate organizational politics

Dice modularidad en Prin

Chapter 3. Modularity

Architects and developers have struggled with the c
quite some time, as is evident in this quote from *Co*
Design (Van Nostrand Reinhold, 1978):

> *95% of the words [written about software archite*
> *extolling the benefits of "modularity" and little,*
> *about how to achieve it.*
>
> —Glenford J

Arch Characteristics/Quality attributes

ISO 25010:2023, SAiP

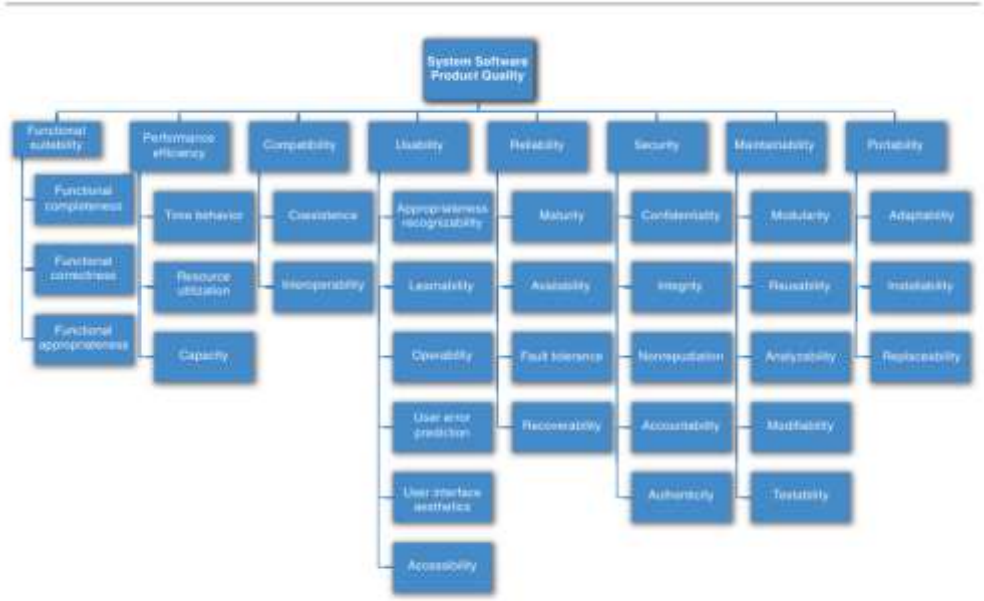


FIGURE 14.1 ISO/IEC FCD 25010 Product Quality Standard

FoSA. Ch 4

Architectural Characteristics Defined

Operational ACh	Structural ACh	Cloud Ch	Cross-Cutting ACh
Term	Term	Term	Term
Availability	Configurability	On-demand scalability	Accessibility
Continuity	Extensibility	On-demand elasticity	Archivability
Performance	Installability	Zone-based availability	Authentication
Recoverability	Leverageability/reuse	Region-based privacy and security	Authorization
Reliability/safety	Localization		Legal
Robustness	Maintainability		Privacy
Scalability	Portability		Security
	Upgradeability		Supportability
			Usability/achievability

Even though to enterprise interprability, but regulations/industries standard/market velocity/etc.

HLD

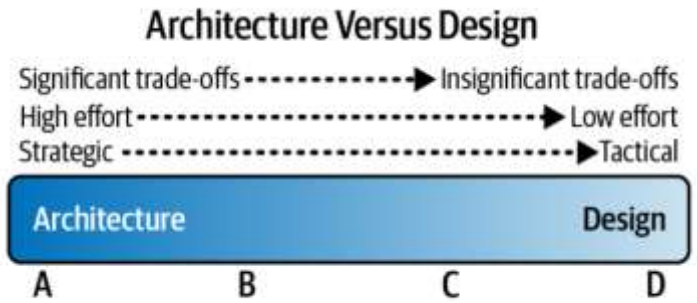


Figure 2-1. The spectrum between architecture and design

Who needs an architect – Fowler, who is architect vs Design: Link

Effort

Architecture is the stuff you can't Google or ask an LLM about.

—Mark Richards

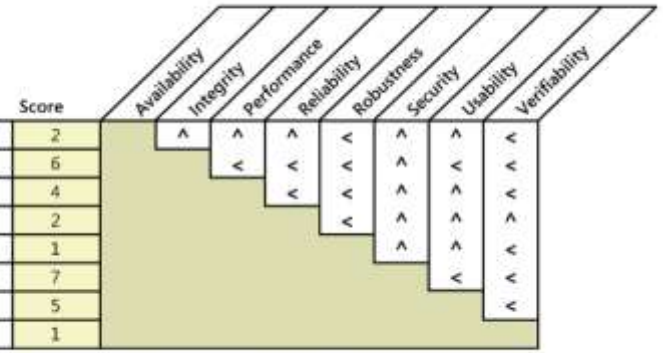
There are no right or wrong answers in architecture—only trade-offs.

—Neal Ford

choice. However, to quote Rich Hickey, the creator of the Clojure programming language:

> *Programmers know the benefits of everything and the trade-offs of nothing. Architects need to understand both.*
>
> —Rich Hickey

SWR, Ch14, - Quality attribute trade-offs



e 14-1. Sample quality attribute prioritization for an airport check-in kiosk.

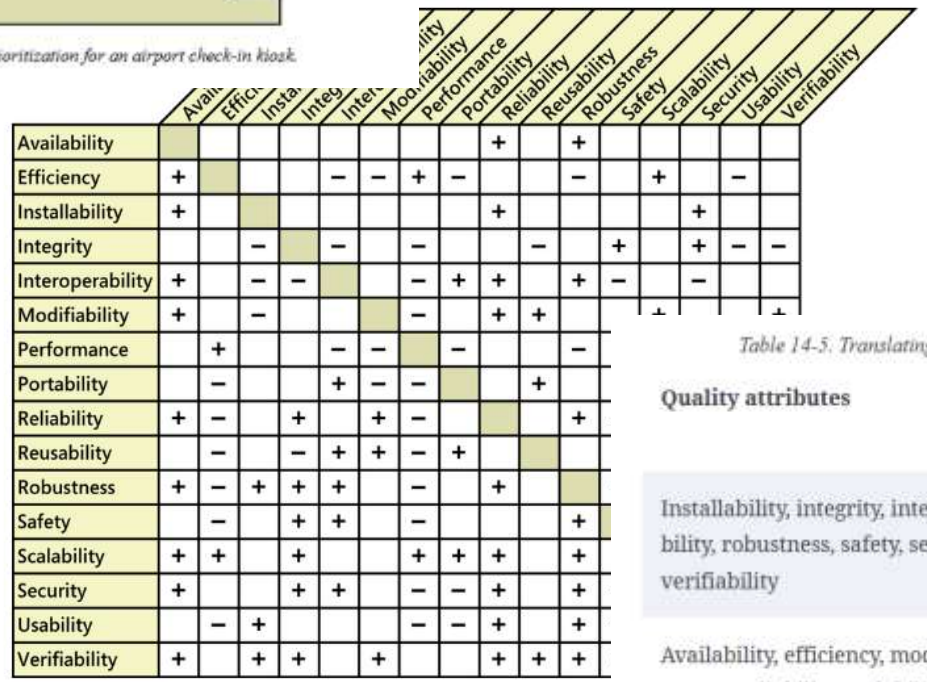
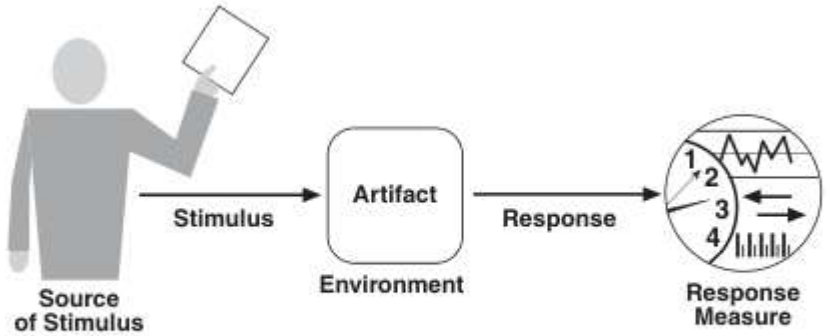


Figure 14-2. Positive and negative relationships among sel

Step 1: Start with a broad taxonomy
Step 2: Reduce the list
Step 3: Prioritize the attributes
Step 4: Elicit specific expectations for each
Step 5: Specify well-structured quality requ

Table 14-5. Translating quality attributes into technical specifications

Quality attributes	Likely technical information category
Installability, integrity, interoperability, reliability, robustness, safety, security, usability, verifiability	Functional requirement
Availability, efficiency, modifiability, performance, reliability, scalability	System architecture
Interoperability, security, usability	Design constraint
Efficiency, modifiability, portability, reliability, reusability, scalability, verifiability,	Design guideline



IGURE 4.1 The parts of a quality attribute scenario

IGURE 4.3 Tactics are intended to control responses to stimuli.

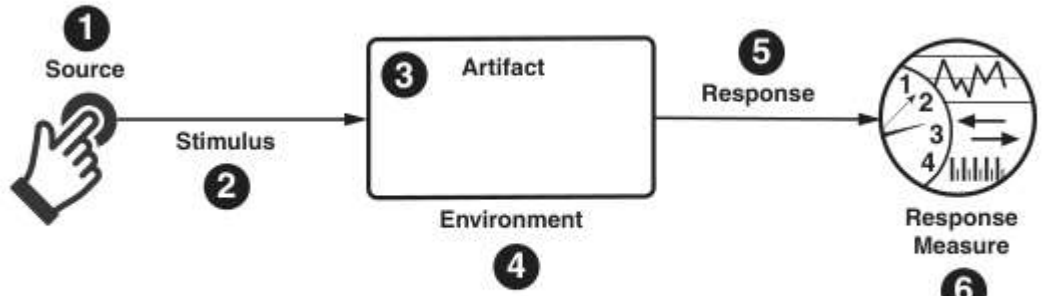
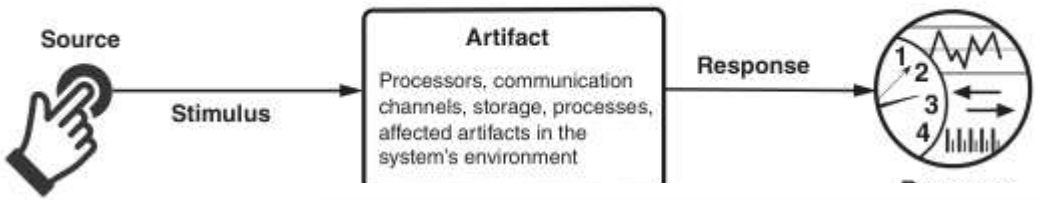


FIGURE 3.1 The parts of a quality attribute scenario



Internal/external: people, hardware, software, physical infrastructure, physical environment

Fault: omiss crash, incor timing, inco response

FIGURE 3.2 A general sc

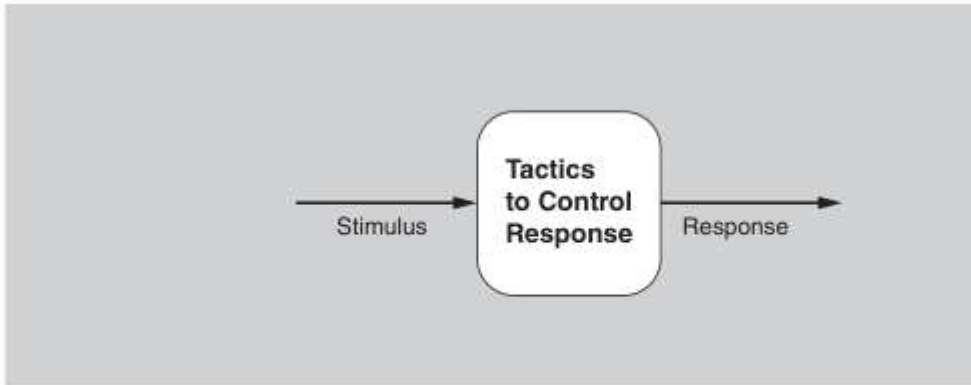
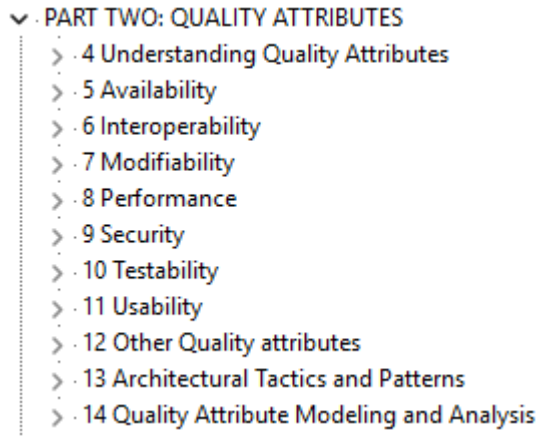
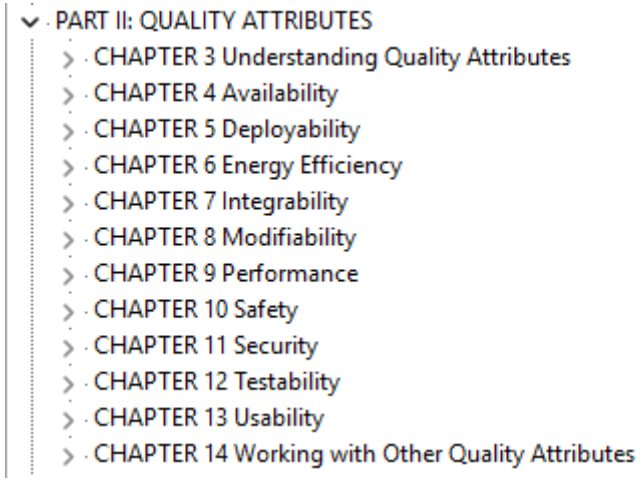


FIGURE 3.3 Tactics are intended to control responses to stimuli.

3rd Edicion



4th Edition



- Availability
- Modifibility
- Performance
- Security
- Testability
- Usability

3rd Edition

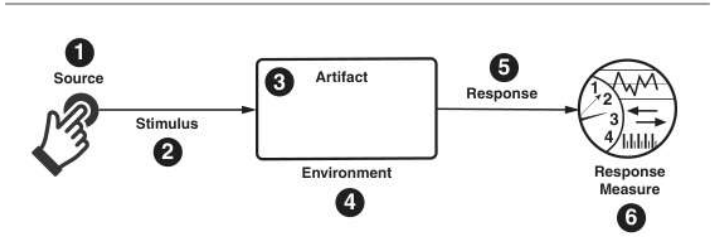


FIGURE 3.1 The parts of a quality attribute scenario

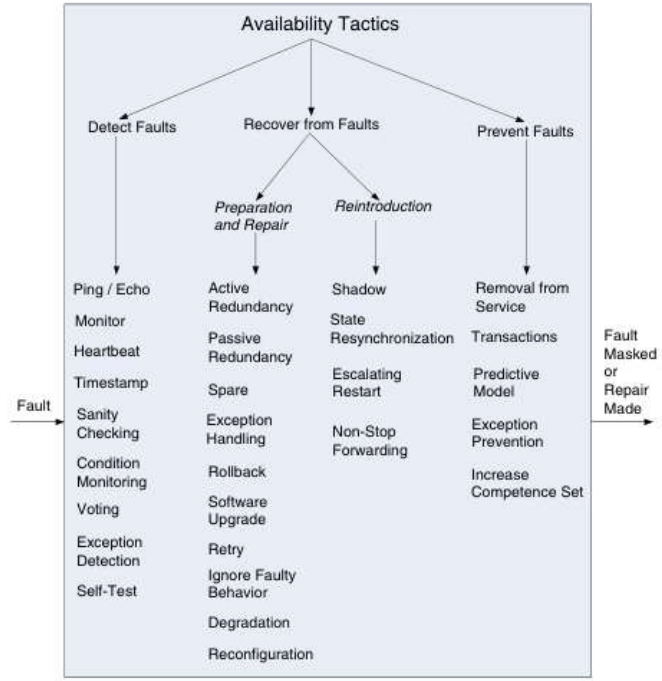


FIGURE 5.5 Availability tactics

4th Edition

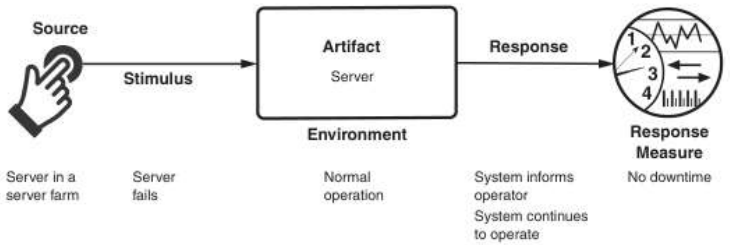


FIGURE 4.1 Sample concrete availability scenario

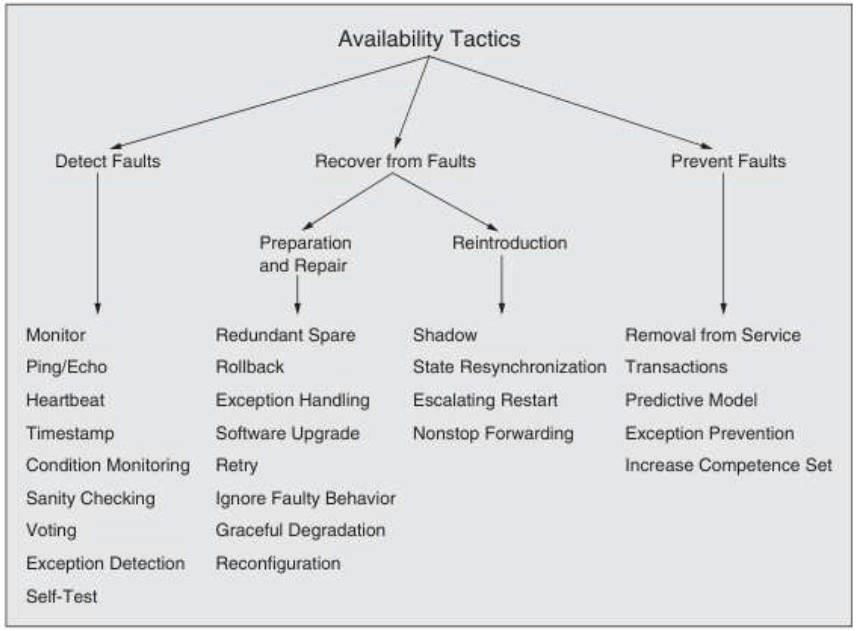


FIGURE 4.3 Availability tactics

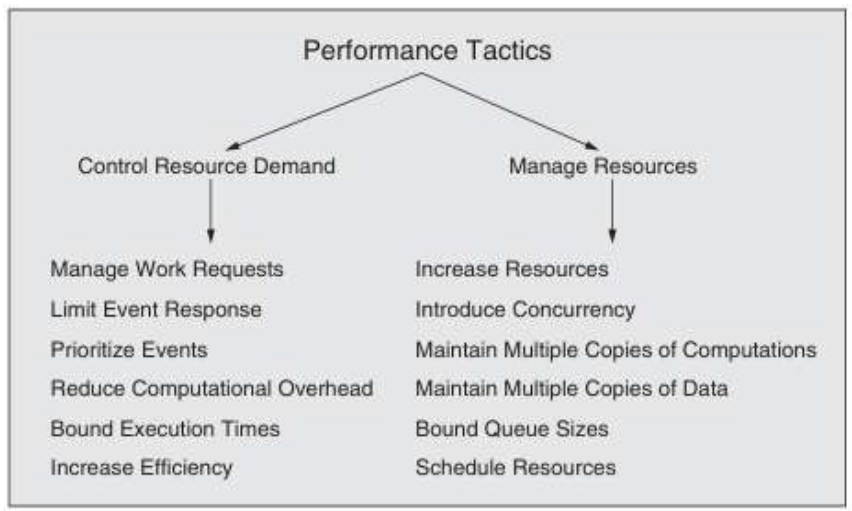


FIGURE 9.3 Performance tactics