

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

Dice modularidad en Prin

- > · Preface
- 1. Introduction
 - Defining Software Architecture
 - Laws of Software Architecture
 - Expectations of an Architect
 - Make Architecture Decisions
 - Continually Analyze the Architecture
 - Keep Current with Latest Trends
 - Ensure Compliance with Decisions
 - Understand Diverse Technologies
 - Know the Business Domain
 - Possess Interpersonal Skills
 - Understand and Navigate Politics
 - Roadmap
 - I. Foundations
- 2. Architectural Thinking
 - Architecture Versus Design
 - Strategic Versus Tactical Decisions
 - Level of Effort
 - The Significance of Trade-Offs
 - Technical Breadth
 - The 20-Minute Rule
 - Developing a Personal Radar
 - Analyzing Trade-Offs
 - Understanding Business Drivers
 - Balancing Architecture and Hands-On Coding
 - There's More to Architectural Thinking
- 3. Modularity
 - Modularity Versus Granularity
 - Defining Modularity
 - Measuring Modularity
 - From Modules to Components
- 4. Architectural Characteristics Defined
 - Architectural Characteristics and System Design
 - Architectural Characteristics (Partially) Listed
 - Trade-Offs and Least Worst Architecture

Chapter 3. Modularity

Architects and developers have struggled with the concept of modularity for quite some time, as is evident in this quote from *Computer Architecture: Art and Science* (Van Nostrand Reinhold, 1978):

95% of the words [written about software architecture] are extolling the benefits of “modularity” and little, if anything, about how to achieve it.

—Glenford J.

Arch Characteristics/Quality attributes

ISO 25010:2023, SAiP

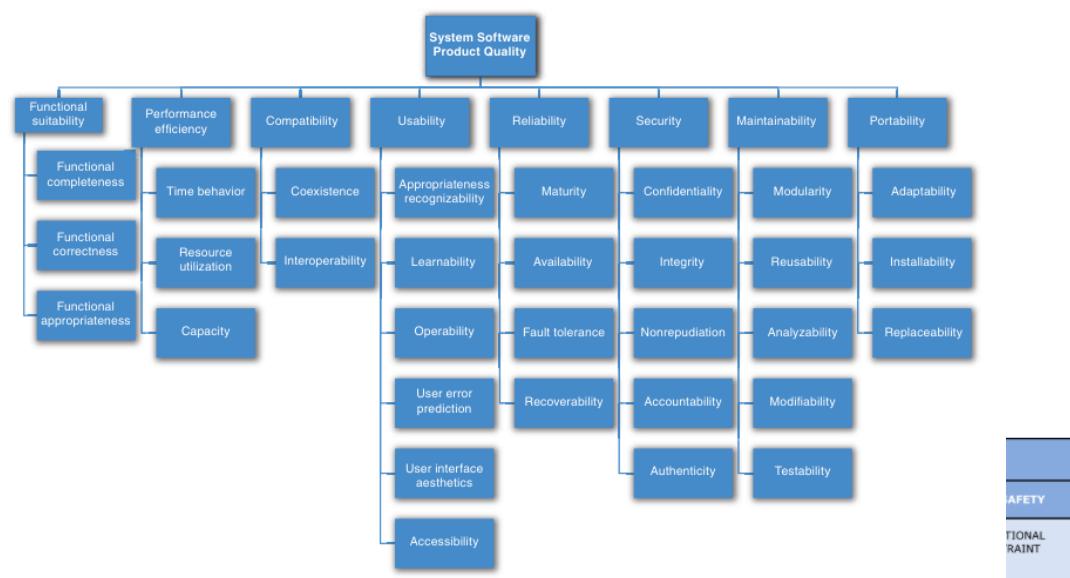


FIGURE 14.1 ISO/IEC FCD 25010 Product Quality Standard



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

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	Maintainability	Legal
		Portability	Privacy
		Robustness	Security
		Scalability	Supportability
			Usability/achievability

HLD

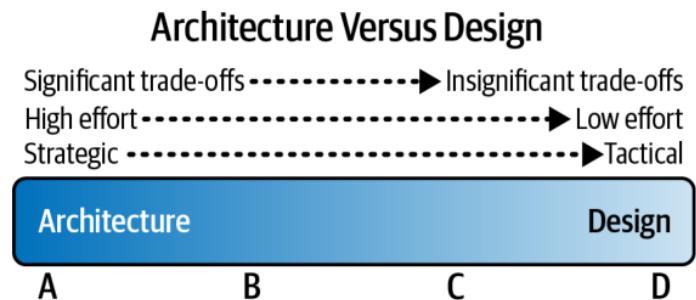


Figure 2-1. The spectrum between architecture and design

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

Attribute	Score	Availability	Integrity	Performance	Reliability	Robustness	Security	Usability	Verifiability
Availability	2		^	^	<	^	^	<	
Integrity	6		<	<	<	^	<		
Performance	4			<	<	^	^	<	
Reliability	2				<	^	^	^	
Robustness	1					^	^	<	
Security	7						^		
Usability	5							<	
Verifiability	1								<

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

SWR, Ch14, - Quality attribute trade-offs

	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability																
Efficiency	+			-	-	+	-			-	+		-			
Installability	+							+					+			
Integrity			-		-				-		+		+	-	-	
Interoperability	+	-	-	-			-	+	+		+	-		-		
Modifiability	+	-					-		+	+						
Performance	+			-	-			-								
Portability		-			+	-	-						+			
Reliability	+	-		+		+	-							+		
Reusability		-		-	+	+	-		+							
Robustness	+	-	+	+	+	+		-								
Safety		-		+	+			-					+			
Scalability	+	+		+				+	+	+			+			
Security	+			+	+			-	-	+			+			
Usability		-	+					-	-	+			+			
Verifiability	+		+	+		+			+	+	+		+			

Figure 14-2. Positive and negative relationships among selected quality attributes.

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

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 requirements

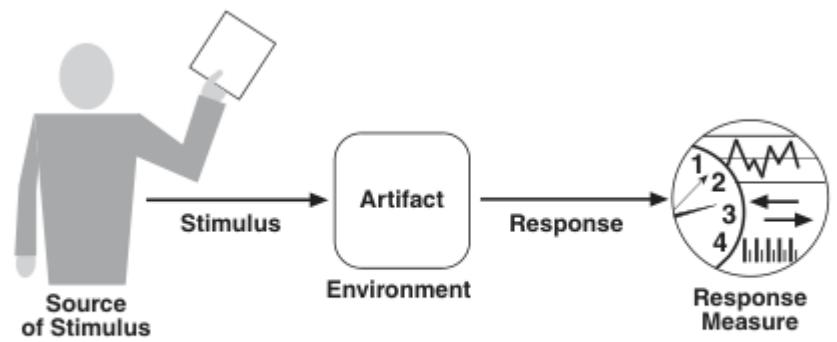


FIGURE 4.1 The parts of a quality attribute scenario

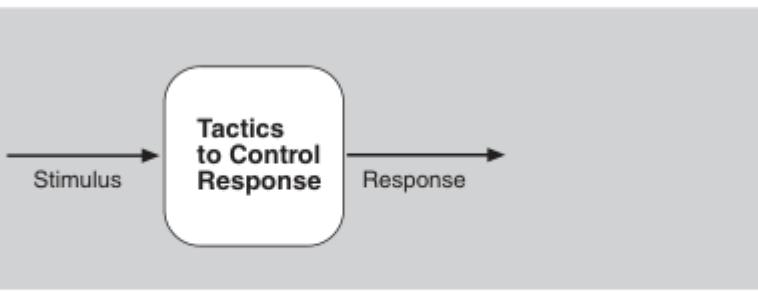


FIGURE 4.3 Tactics are intended to control responses to stimuli.

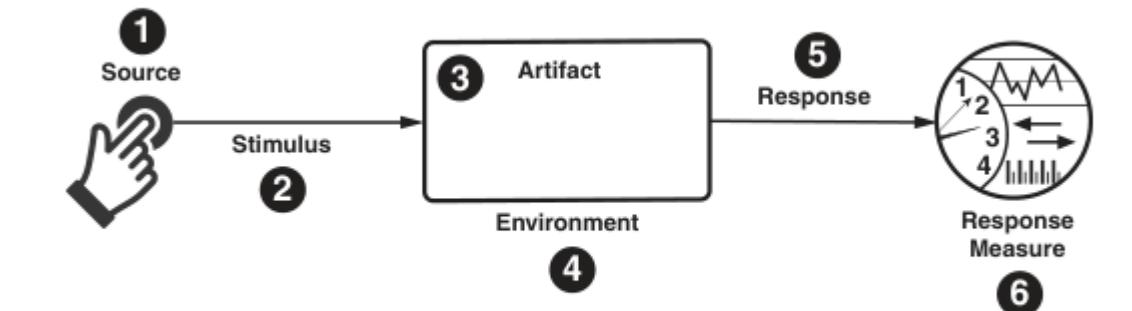


FIGURE 3.1 The parts of a quality attribute scenario

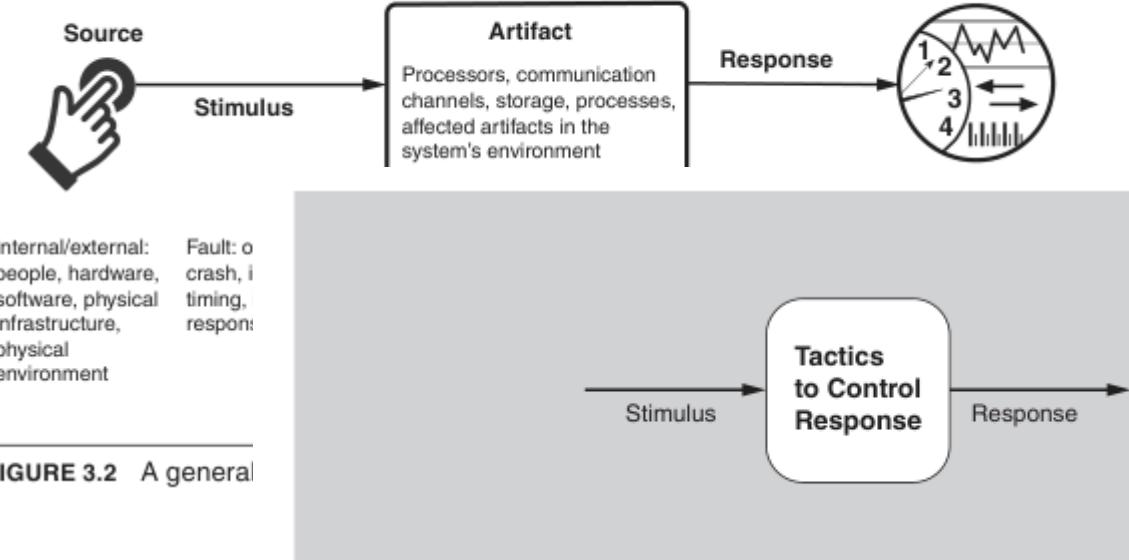


FIGURE 3.2 A general

FIGURE 3.3 Tactics are intended to control responses to stimuli.

3rd Edicion

- PART TWO: QUALITY ATTRIBUTES
 - > · 4 Understanding Quality Attributes
 - > · 5 Availability
 - > · 6 Interoperability
 - > · 7 Modifiability
 - > · 8 Performance
 - > · 9 Security
 - > · 10 Testability
 - > · 11 Usability
 - > · 12 Other Quality attributes
 - > · 13 Architectural Tactics and Patterns
 - > · 14 Quality Attribute Modeling and Analysis

4th Edition

- PART II: QUALITY ATTRIBUTES
 - > · CHAPTER 3 Understanding Quality Attributes
 - > · CHAPTER 4 Availability
 - > · CHAPTER 5 Deployability
 - > · CHAPTER 6 Energy Efficiency
 - > · CHAPTER 7 Integrability
 - > · CHAPTER 8 Modifiability
 - > · CHAPTER 9 Performance
 - > · CHAPTER 10 Safety
 - > · CHAPTER 11 Security
 - > · CHAPTER 12 Testability
 - > · CHAPTER 13 Usability
 - > · CHAPTER 14 Working with Other Quality Attributes

- Availability
- Modifiability
- Performance
- Security
- Testability
- Usability

3rd Edition

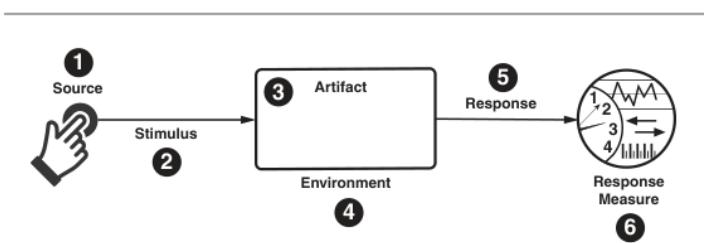


FIGURE 3.1 The parts of a quality attribute scenario

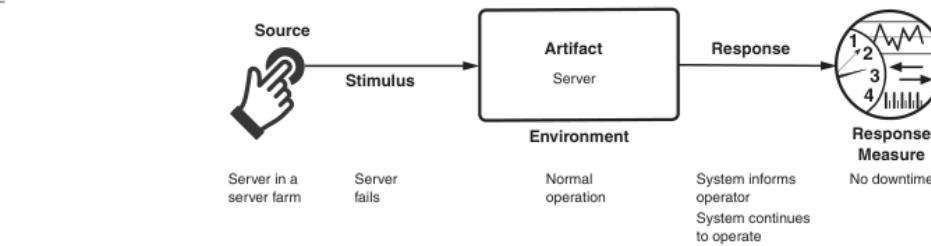


FIGURE 4.1 Sample concrete availability scenario

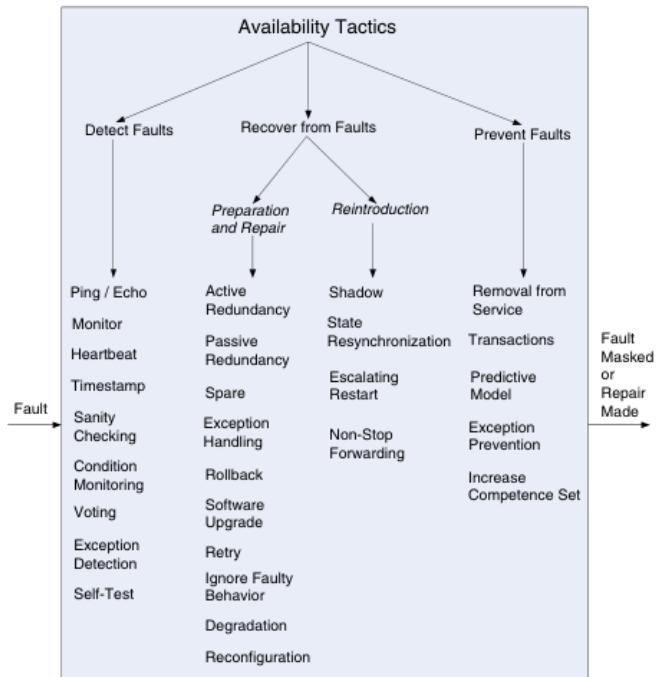


FIGURE 5.5 Availability tactics

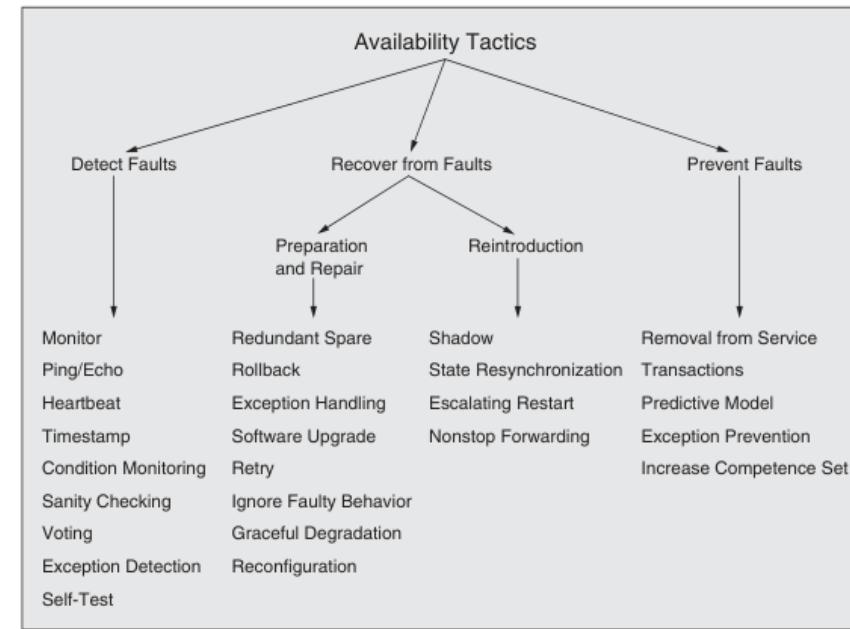


FIGURE 4.3 Availability tactics

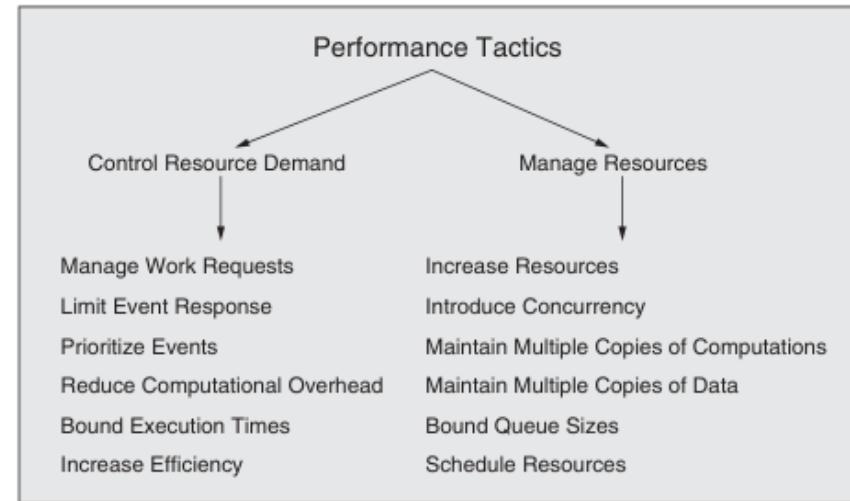
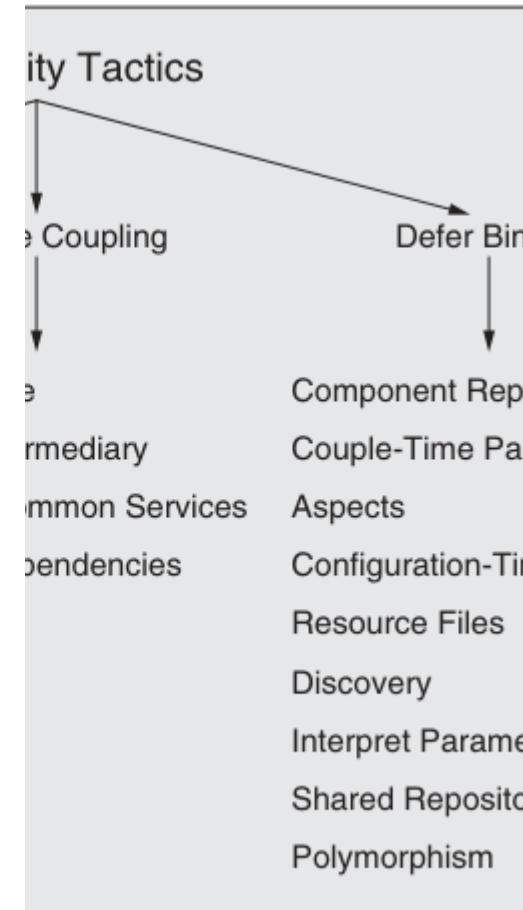
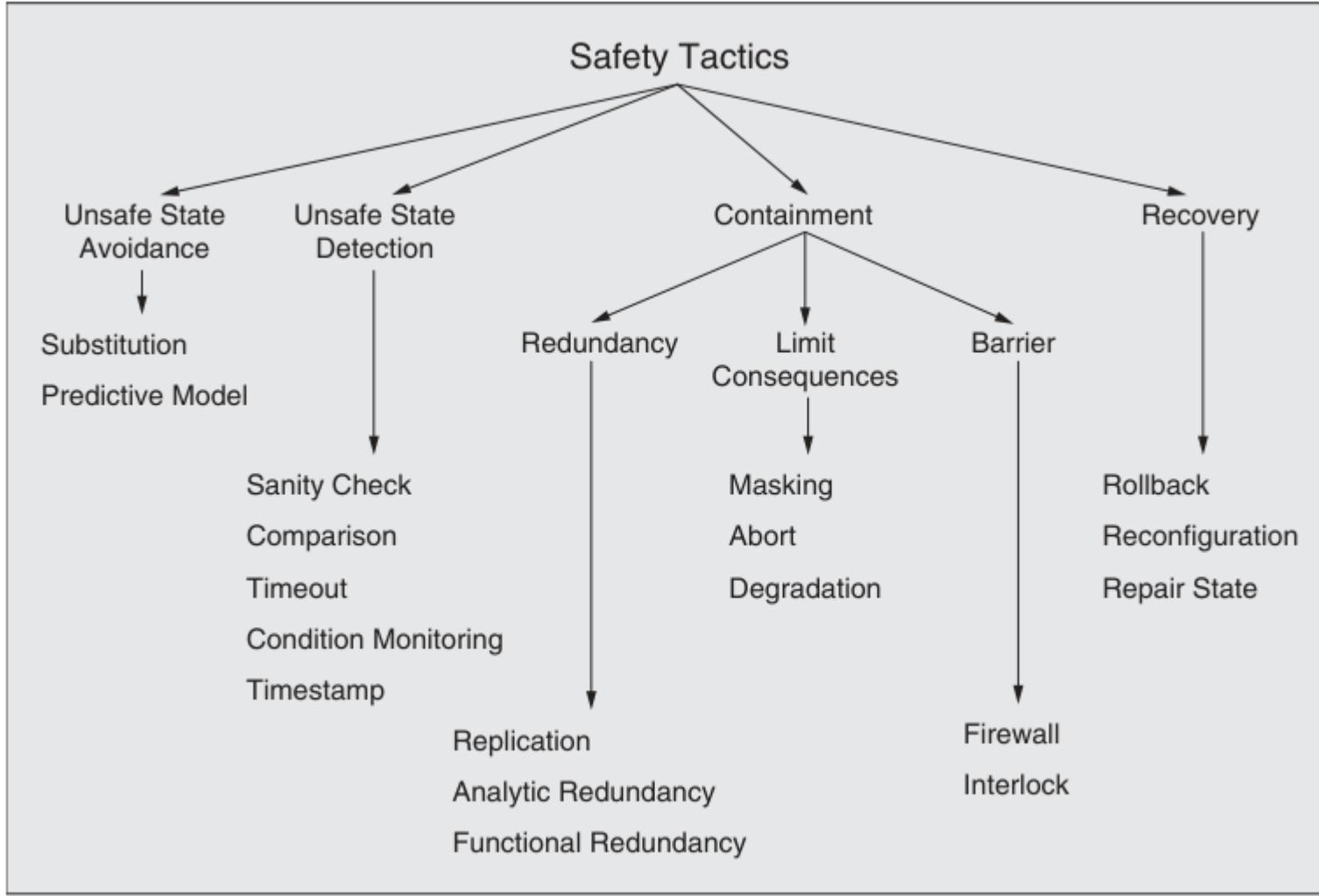
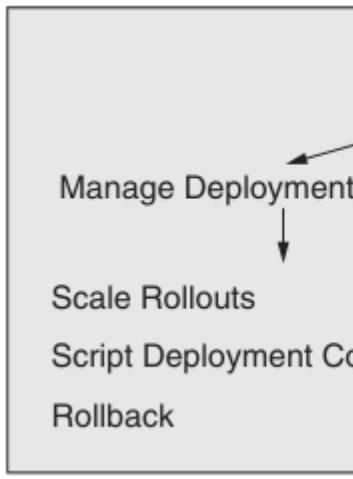
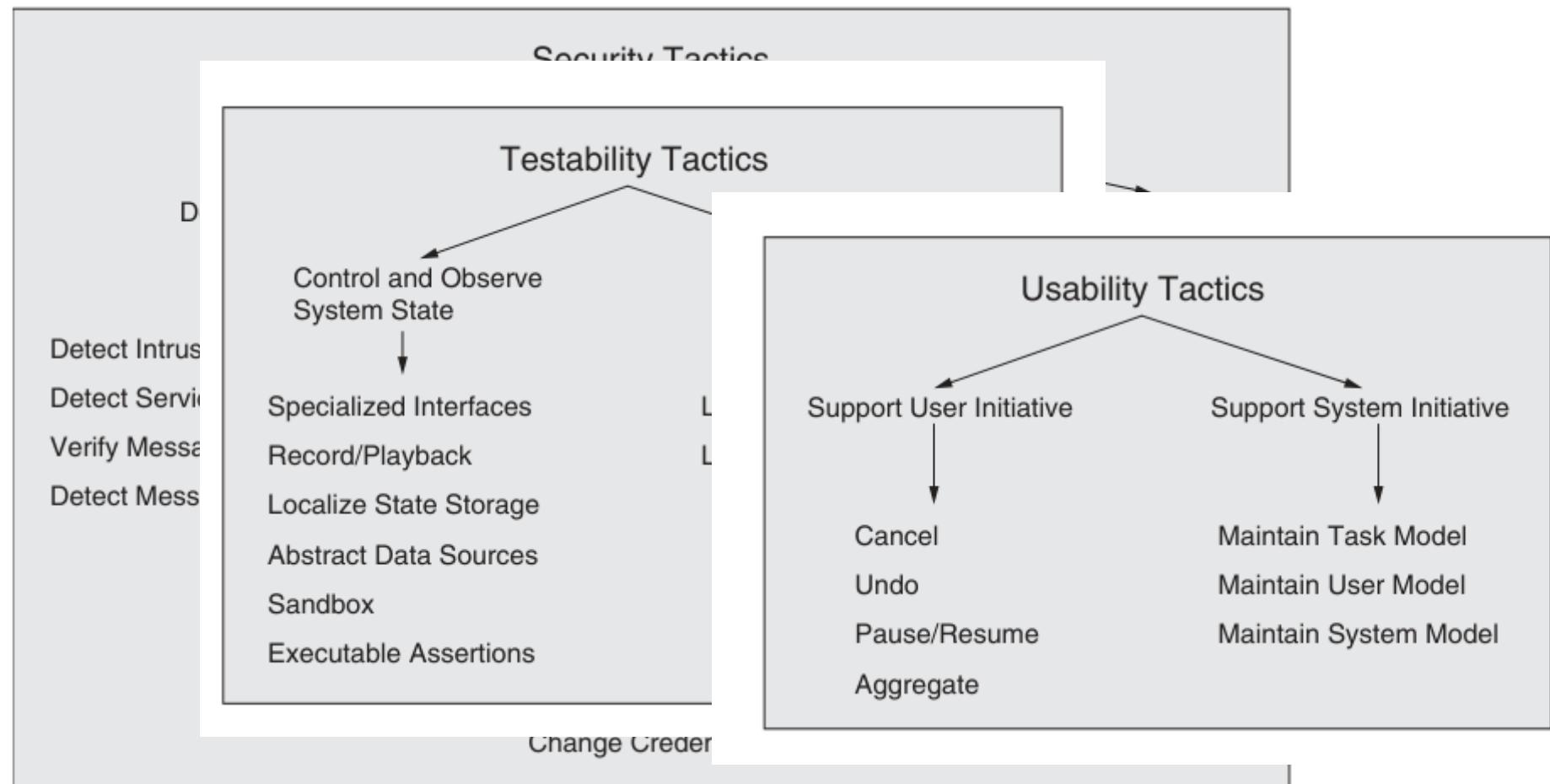


FIGURE 9.3 Performance tactics





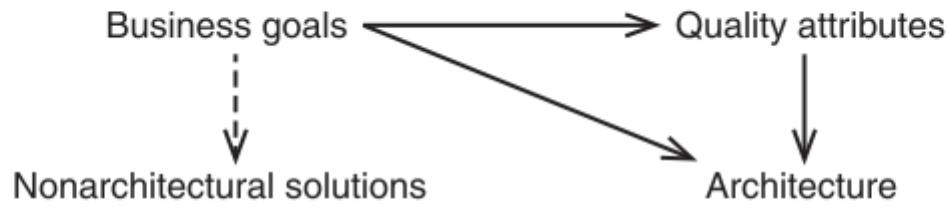


FIGURE 19.1 Some business goals may lead to quality attribute requirements, or lead directly to architectural decisions, or lead to non-architectural solutions.

TABLE 19.1 Tabular Form of the Utility Tree for a System in the Healthcare Space

Quality Attribute	Attribute Refinement	ASR Scenario
Performance	Transaction response time	A user updates a patient's account in response to a change-of-address notification while the system is under peak load, and the transaction completes in less than 0.75 seconds. (H, H)
	Throughput	At peak load, the system is able to complete 150 normalized transactions per second. (M, M)
Usability	Proficiency training	A new hire with two or more years' experience in the business can learn, with 1 week of training, to execute any of the system's core functions in less than 5 seconds. (M, L)
	Efficiency of operations	A hospital payment officer initiates a payment plan for a patient while interacting with that patient and completes the process with no input errors. (M, M)
Configurability	Data configurability	A hospital increases the fee for a particular service. The configuration team makes and tests the change in 1 working day; no source code needs to change. (H, L)
Maintainability	Routine changes	A maintainer encounters response-time deficiencies, fixes the bug, and distributes the bug fix with no more than 3 person-days of effort. (H, M)
		A reporting requirement requires a change to the report-generating metadata. Change is made and tested in 4 person-hours of effort (M, L)
	Upgrades to commercial components	The database vendor releases a new major version that is successfully tested and installed in less than 3 person-weeks. (H, M)
Security	Adding new feature	A feature that tracks blood bank donors is created and successfully integrated within 2 person-months. (M, M)
	Confidentiality	A physical therapist is allowed to see that part of a patient's record dealing with orthopedic treatment, but not other parts or any financial information. (H, M)
	Resisting attacks	The system repels an unauthorized intrusion attempt and reports the attempt to authorities within 90 seconds. (H, M)
Availability	No down time	The database vendor releases new software, which is hot-swapped into place, with no downtime. (H, L)
		The system supports 24/7/365 web-based account access by patients. (M, M)

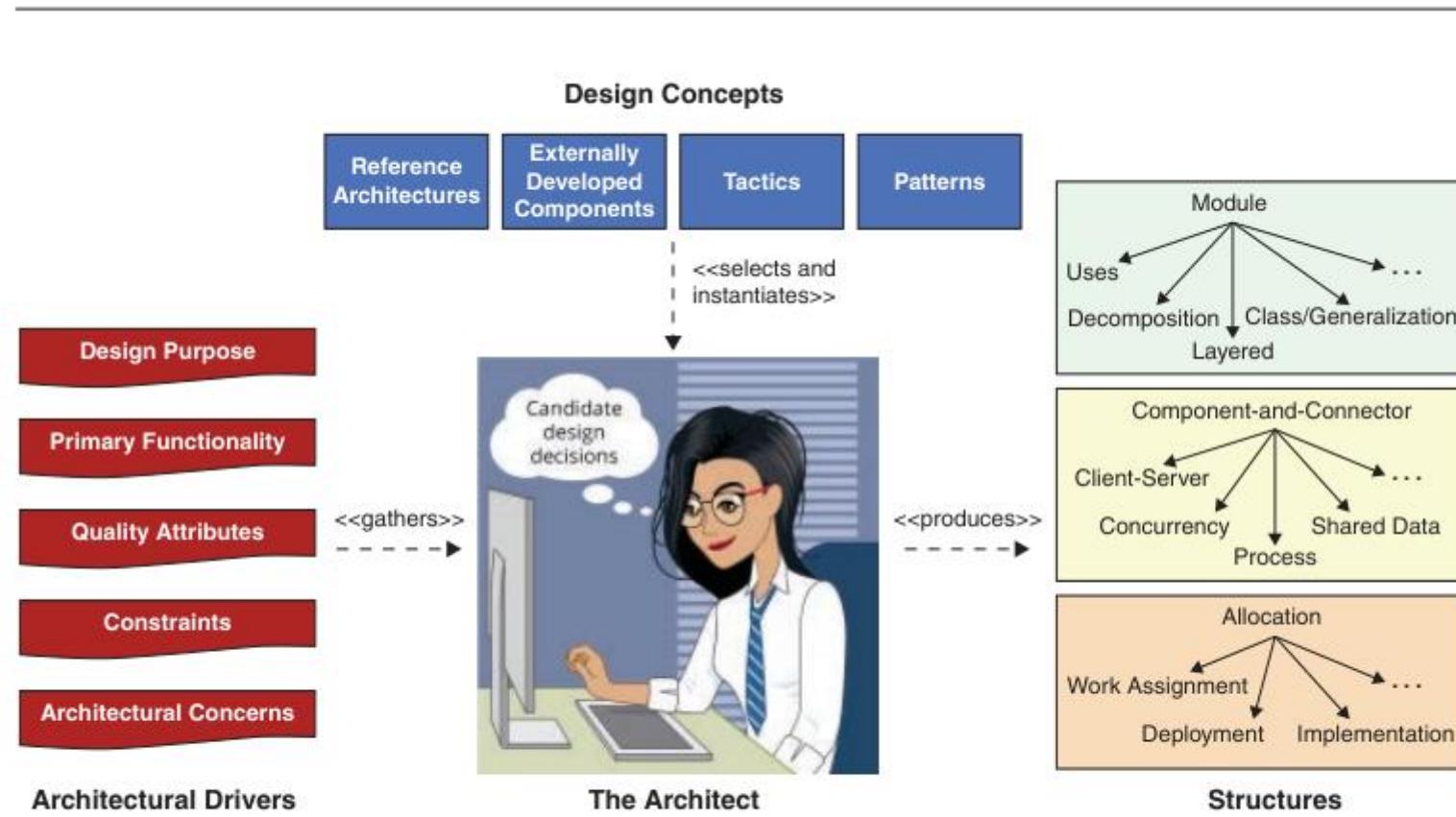


FIGURE 20.1 Overview of the architecture design activity

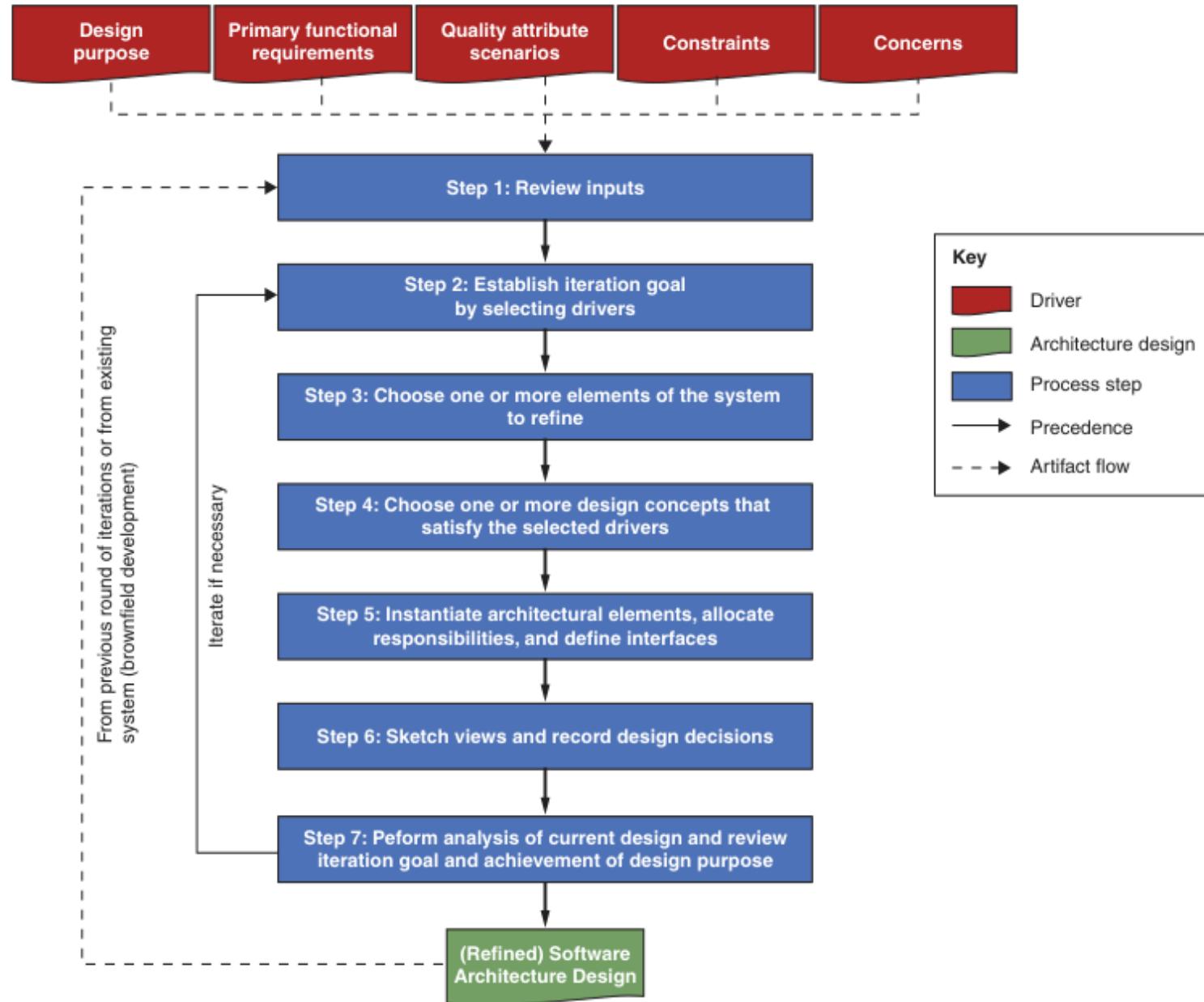


FIGURE 20.3 Steps and artifacts of ADD

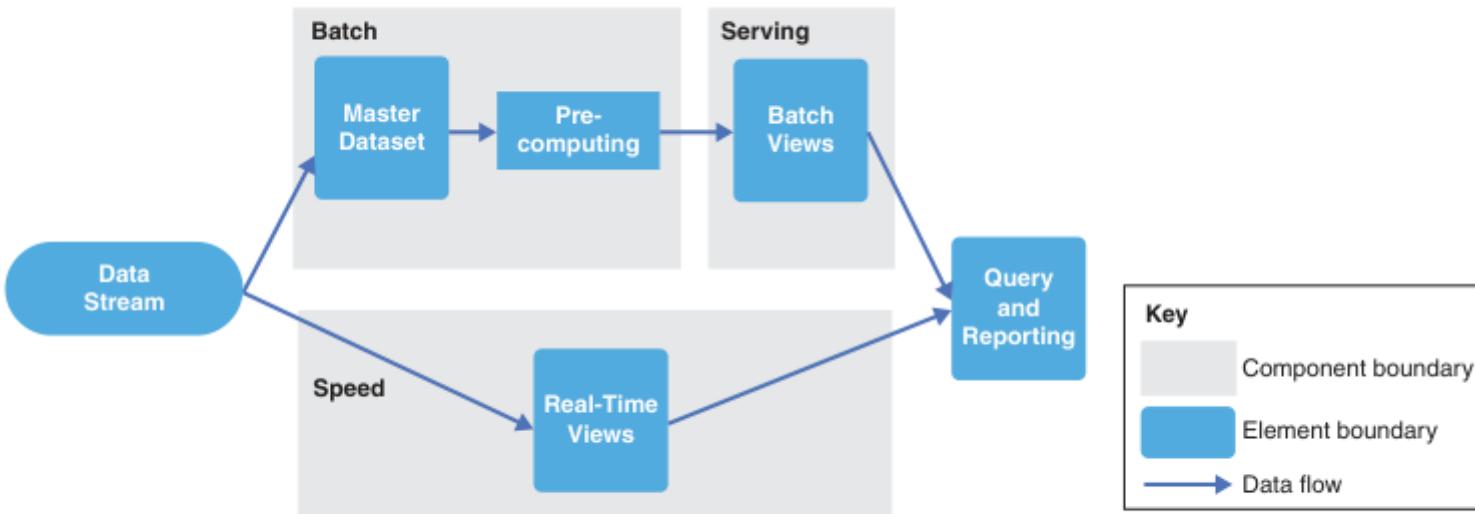


FIGURE 20.4 Example preliminary documentation

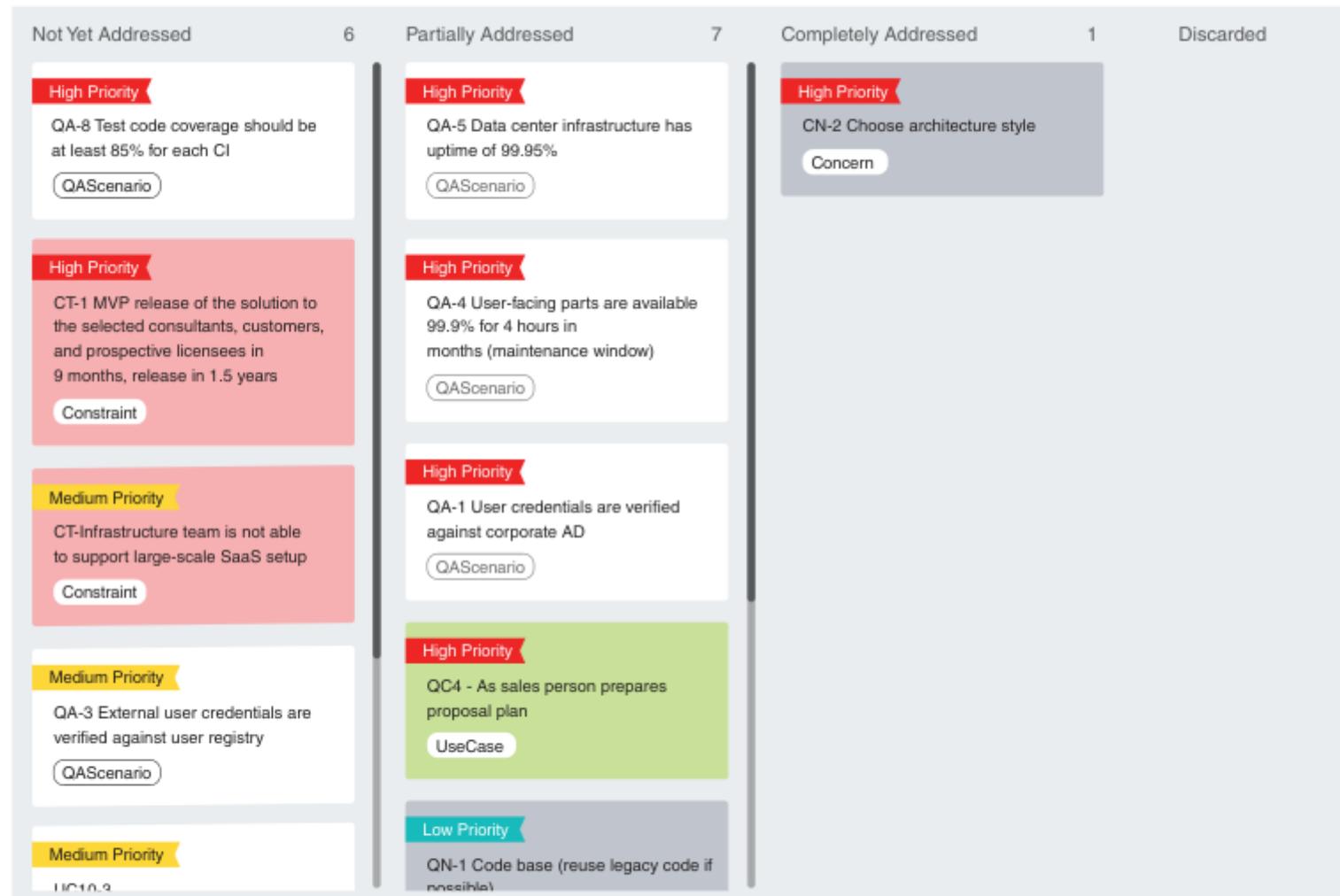


FIGURE 20.5 A Kanban board used to track design progress

