Quality attributes generic scenarios

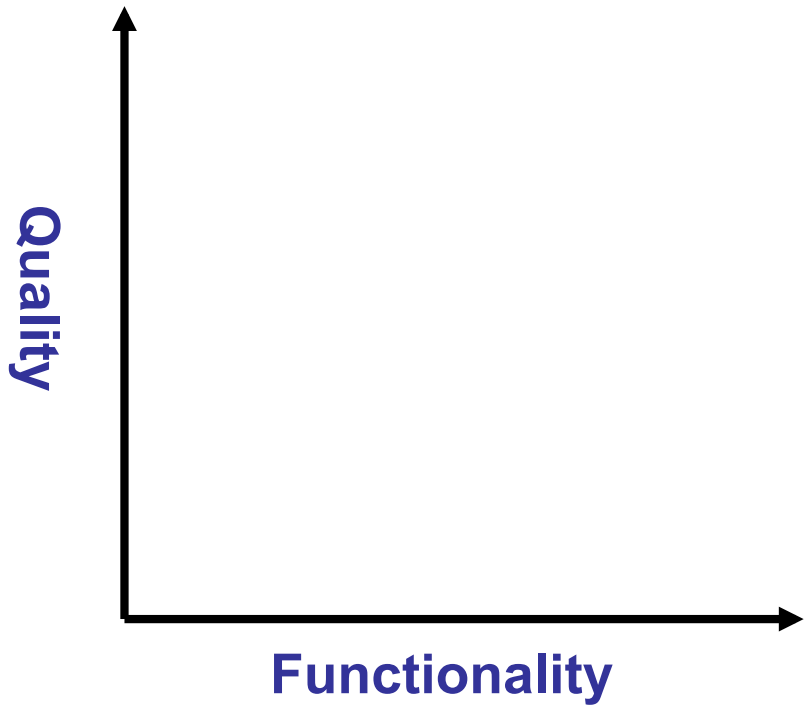
Prof. Paolo Ciancarini
Corso di Architettura del Software
CdL M Informatica
Università di Bologna

Agenda

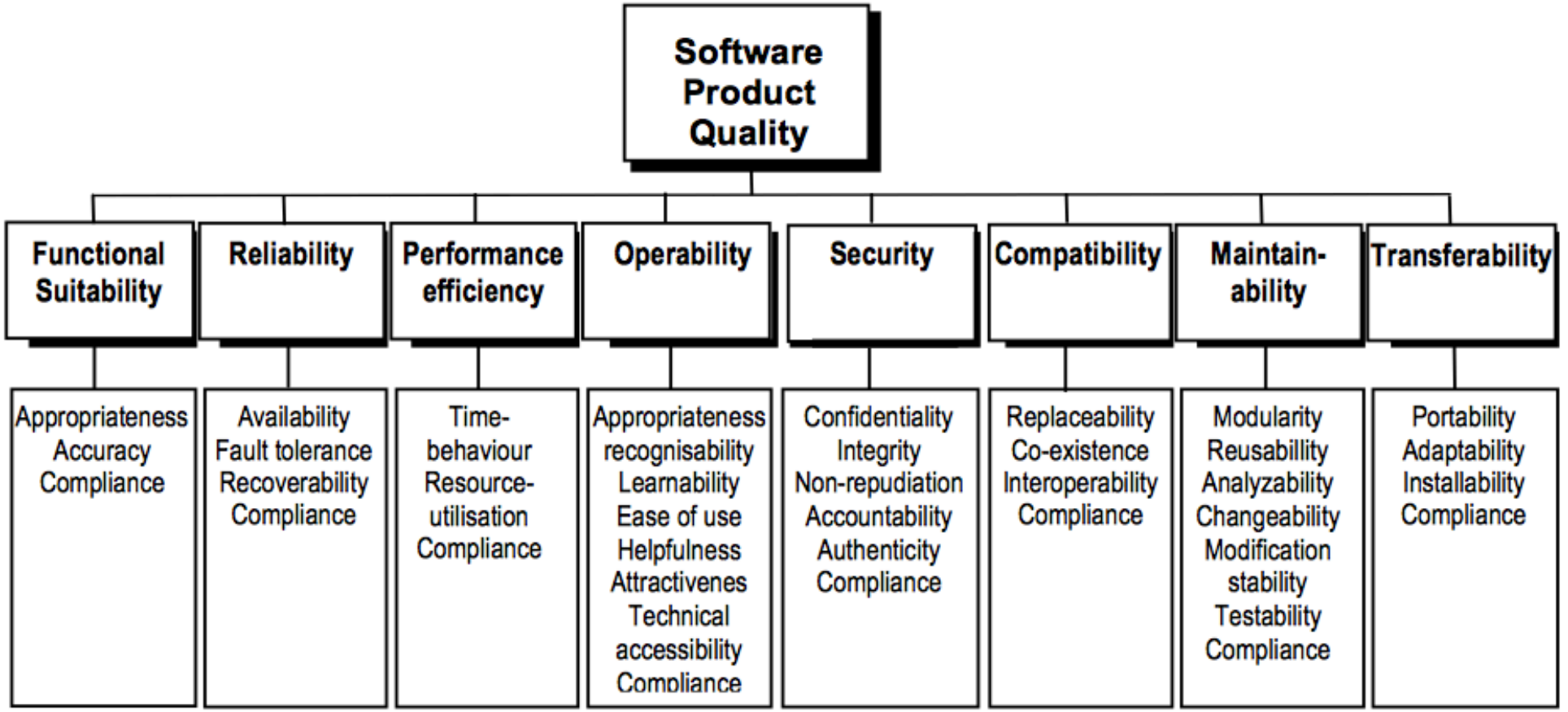
- The software architecture of any system is mainly influenced by extra-functional, or *non functional*, qualities
- How to express the qualities we want our architecture to provide?
- What does it mean to say that a system is modifiable or reliable or secure?

Functionality *vs* Quality Attributes

- Functionality and architectural qualities are orthogonal



Main software qualities (according to ISO 25010)



Some important qualities

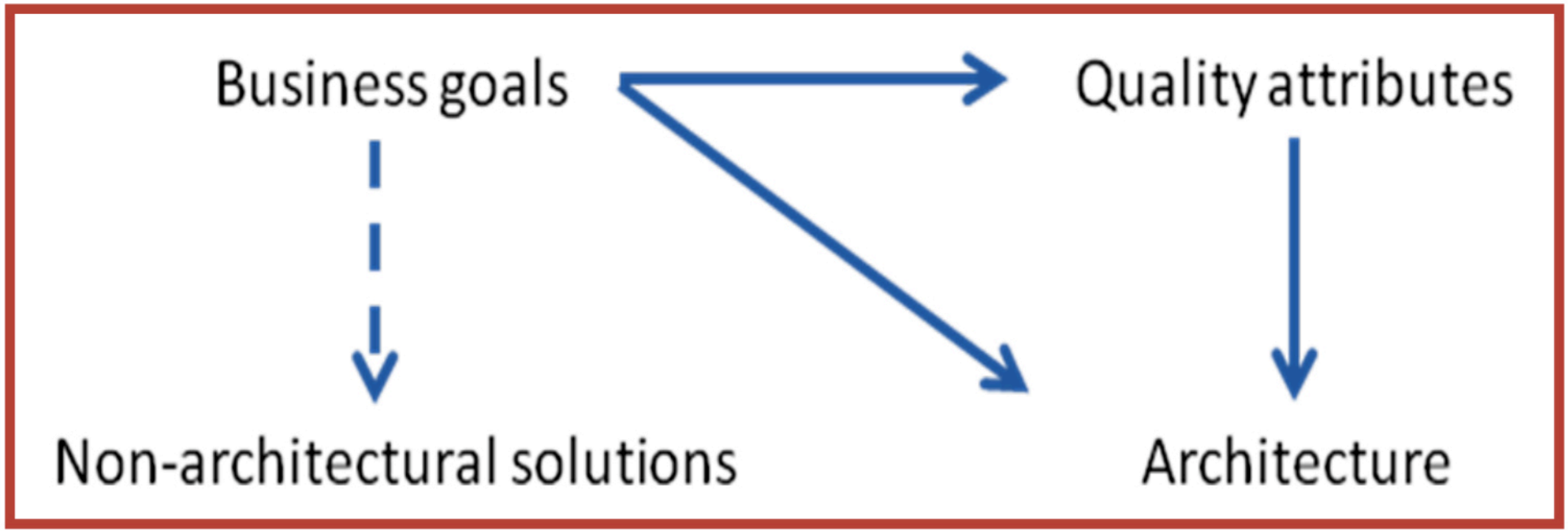
- Usability
- Modifiability
- Performance
- Security
- Testability
- Availability
- Time to market
- Cost and benefit
- Projected System lifetime
- Targeted Market
- Rollout Schedule
- Integration / Legacy

Some qualities are “architectural”

- Conceptual Integrity the design is coherent
- Correctness the design is correct wrt to the requirements
- Completeness the design covers all the requirements
- Flexibility the design supports future changes to its requirements
- Reusability the design uses existing assets
- Buildability the design is realistic and suitable for its context

Qualities and trade-offs

- The qualities are all good
- The value of a quality is project specific
- The qualities are not independent

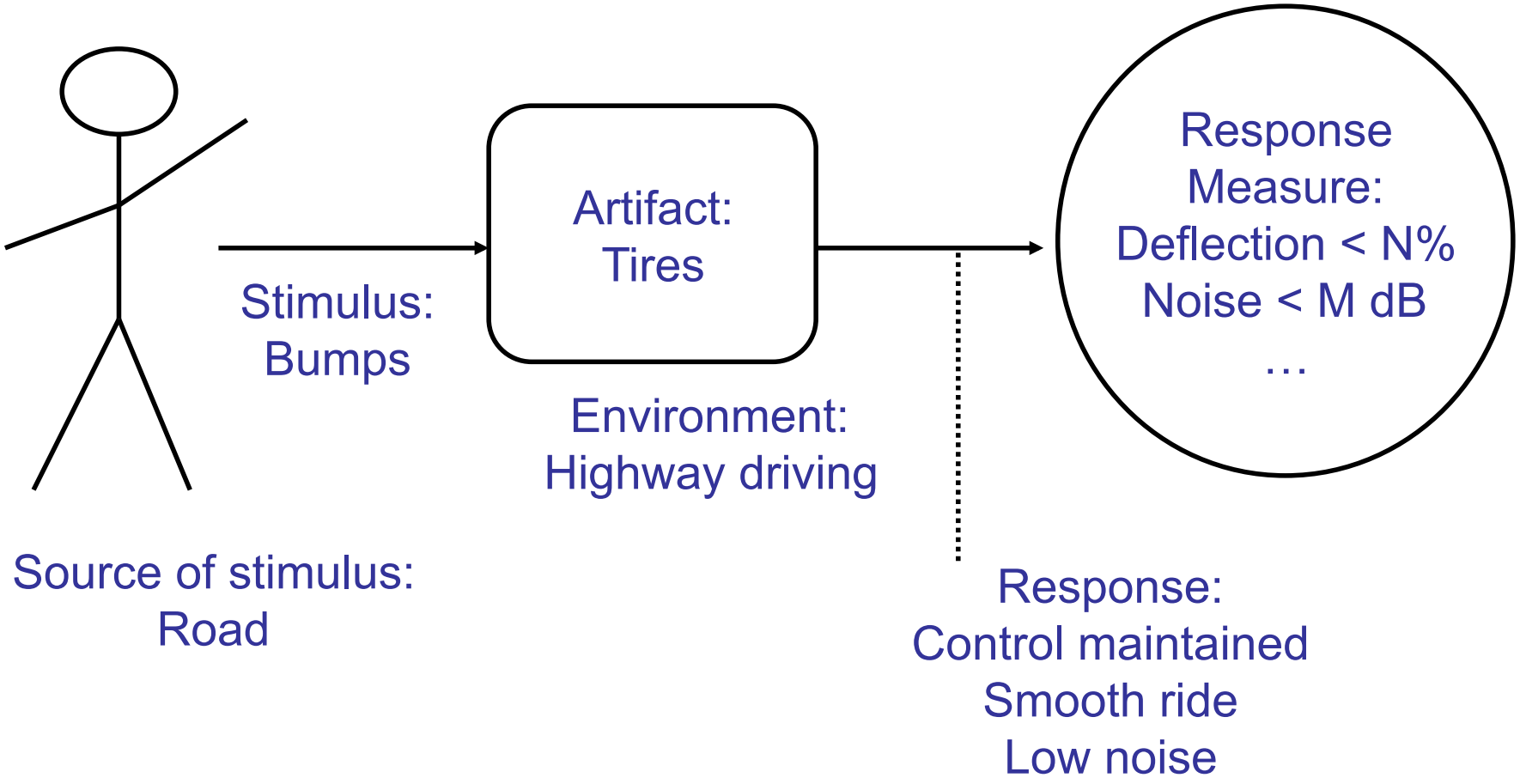


Quality Attribute Scenario

i. Source of stimulus
ii. Stimulus
iii. Environment
iv. Artifact
v. Response
vi. Response measure

In the *environment*, the *source* throws the *stimulus* and hits the system in the *artifact*

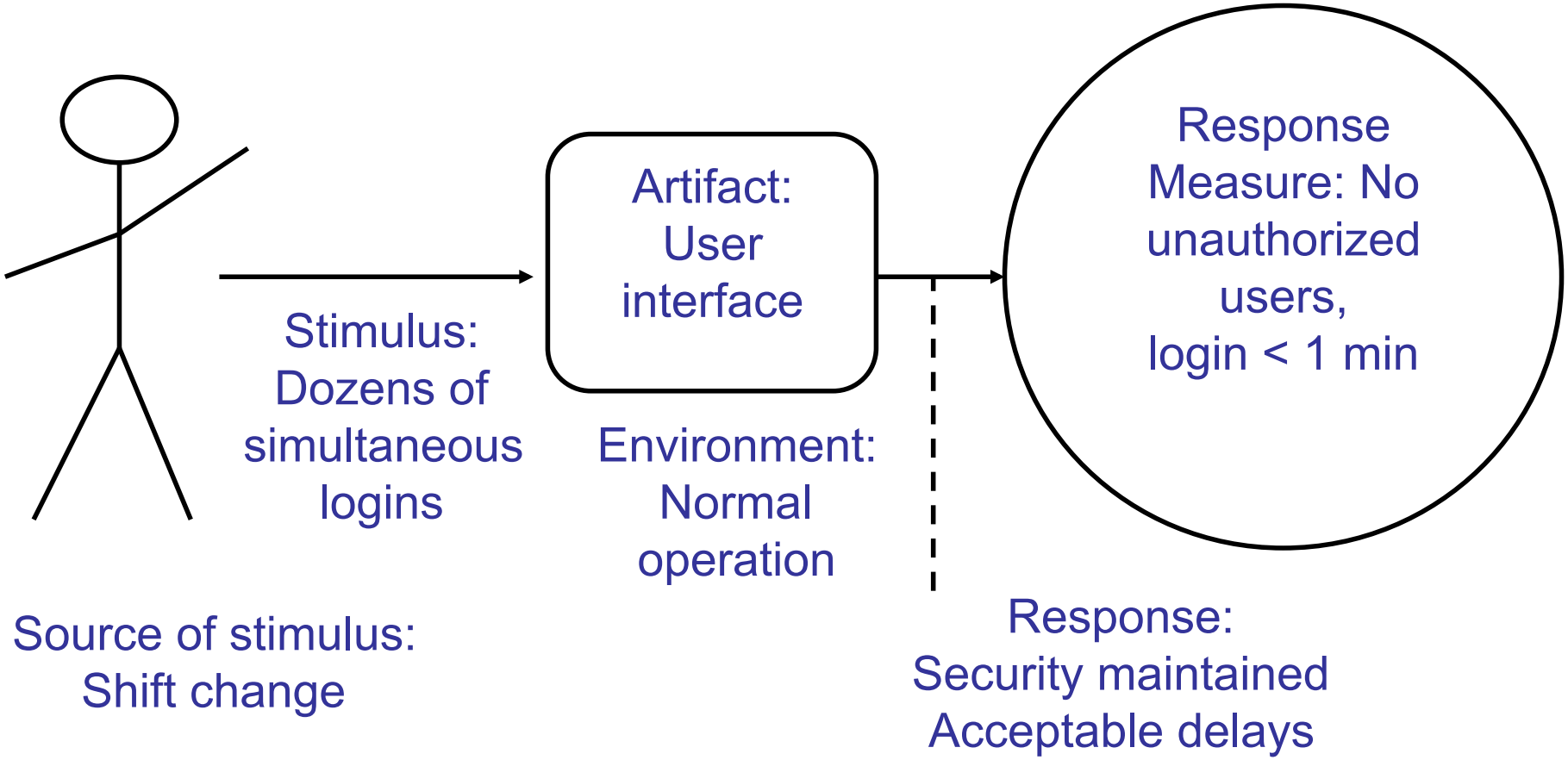
Example from cars



Remember

- One stimulus per scenario
- One environment per scenario
- One artifact per scenario
- Multiple response measures are OK

Example from software



Qualities must be testable

- To be effective, quality attribute scenarios must be *testable* (just like any other requirement)
- Therefore, the
  - Stimulus
  - Artifact
  - Environment
  - Response measure(s)

  must be clear and specific

Different quality attributes

- **Modifiability** is about the cost of change, both in time and money. Performance is about timeliness. Events occur and the system must respond in a timely fashion.
- **Testability** refers to the ease with which the software can be made to demonstrate its faults or lack thereof. To be testable the system must control inputs and be able to observe outputs
- **Maintainability** is the ease with which a product can be maintained in order to isolate and correct defects, prevent unexpected breakdowns, meet new requirements
- **Usability** is how easy it is for the user to accomplish tasks and what support the system provides for the user to accomplish this. Dimensions:
  - Learning system features
  - Using the system efficiently
  - Minimizing the impact of errors
  - Adapting the system to the user's needs
  - Increasing confidence and satisfaction

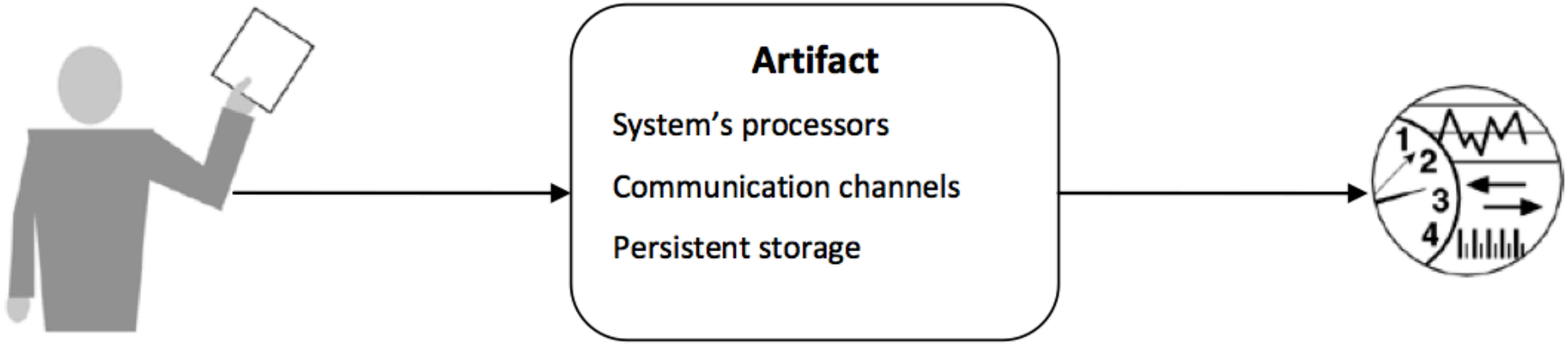
Quality attributes

- **Availability** is concerned with system failure and duration of system failures. System failure means unreadiness for correct service, when the system does not provide the service for which it was intended
- **Reliability**: the ability of a system or component to function under stated conditions for a specified period of time (=continuity of correct service)
- **Dependability**: availability + reliability + maintainability
- **Safety** absence of catastrophic consequences on the users or the environment
- **Security** is the ability of the system to prevent or resist unauthorized access while providing access to legitimate users. An attack is an attempt to breach security

Quality attribute scenarios

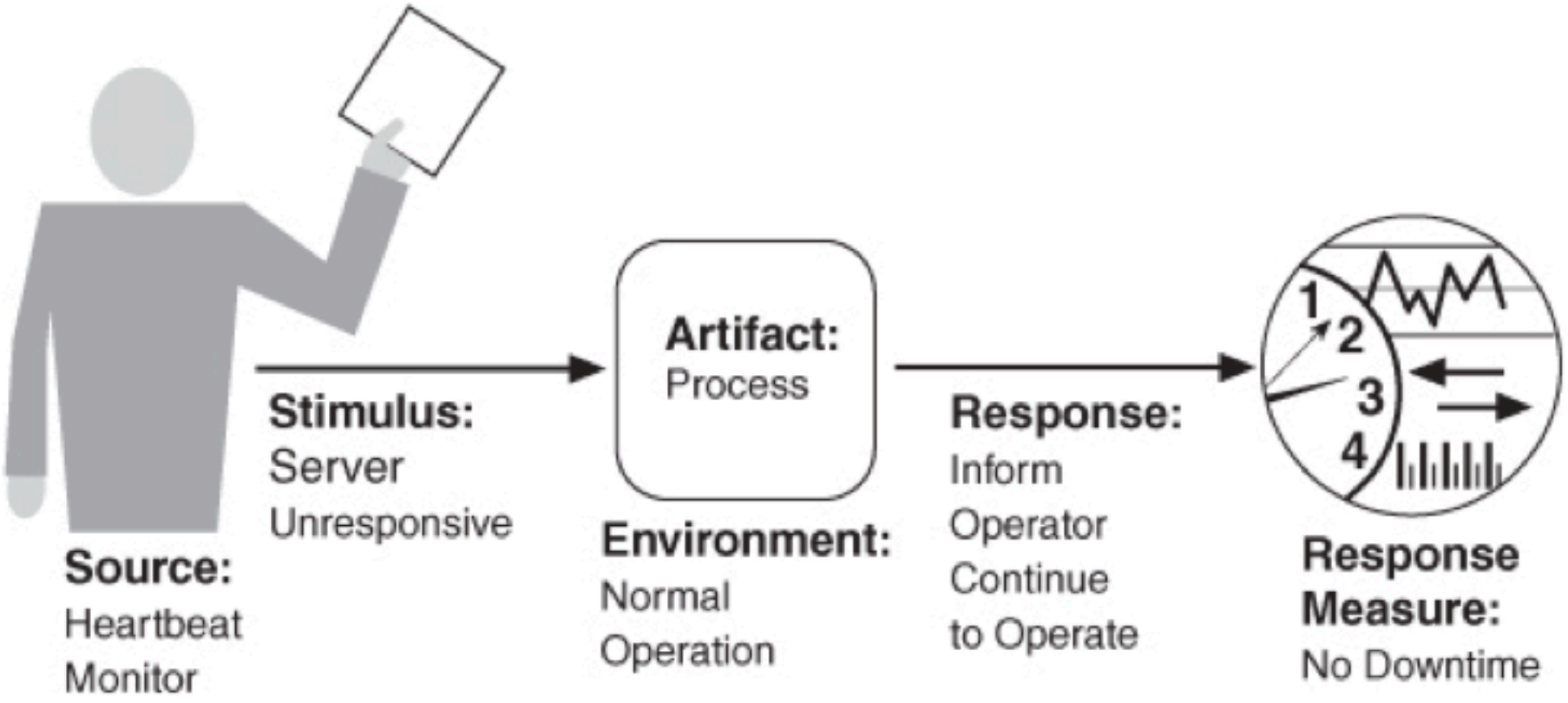
- A Quality Attribute Scenario is a specific requirement. There are 6 parts:
  1. Source of stimulus (e.g., human, computer system, etc.)
  2. Stimulus – a condition that needs to be considered
  3. Environment - what are the conditions when the stimulus occurs?
  4. Artifact – what elements of the system are stimulated.
  5. Response – the activity undertaken after arrival of the stimulus
  6. Response measure – when the response occurs it should be measurable so that the requirement can be tested.

General availability scenario

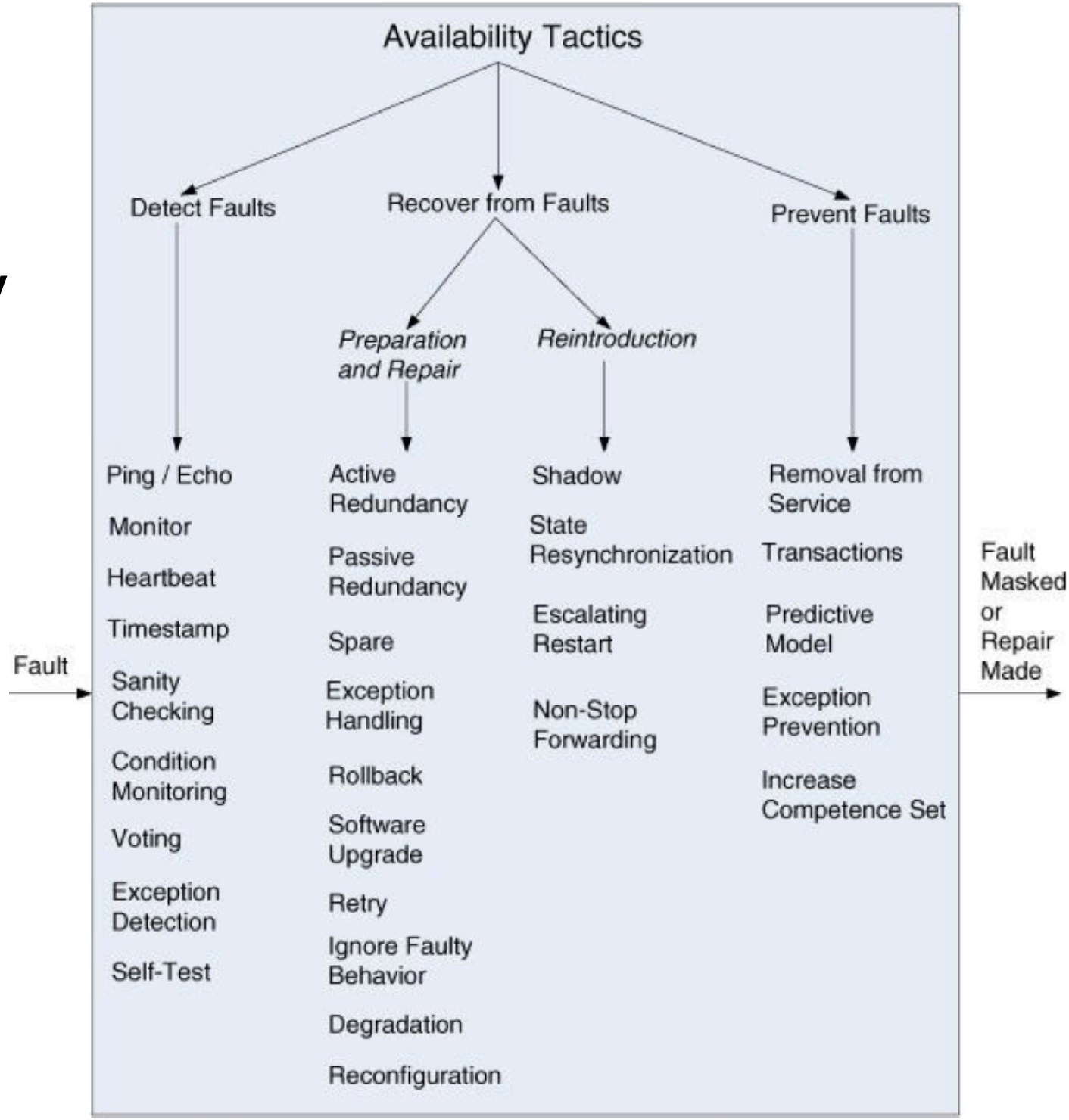


Source	Stimulus	Environment	Response	Measure
Internal to system	Crash	Normal operation	Prevent the failure	Time interval available
External to system	Omission	Startup	Log the failure	Availability %
	Timing	Shutdown	Notify users / operators	Detection time
	No response	Repair mode	Disable source of failure	Repair time
	Incorrect response	Degraded (failsafe) mode	Temporarily unavailable	Degraded mode time interval
		Overloaded operation	Continue (normal / degraded)	Unavailability time interval

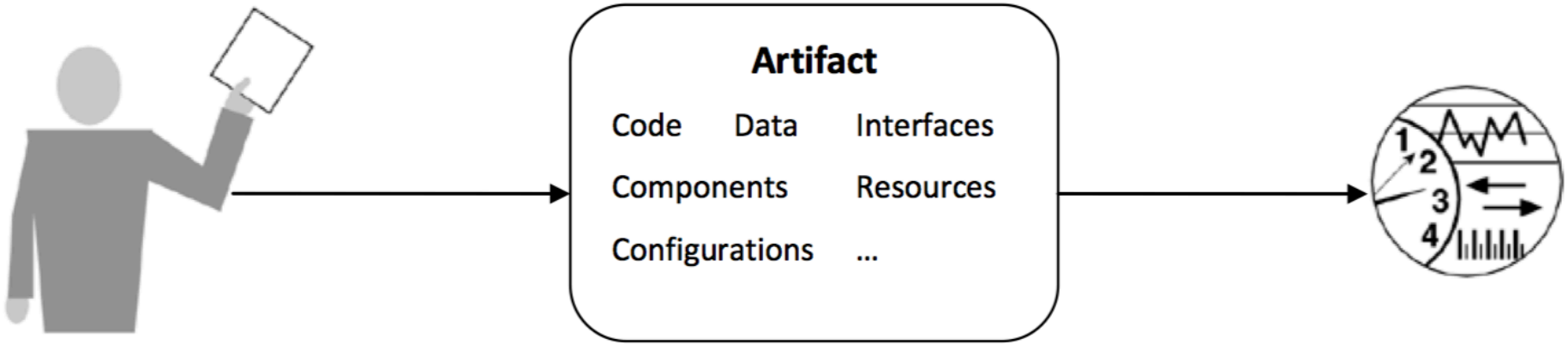
Sample availability scenario



Availability tactics

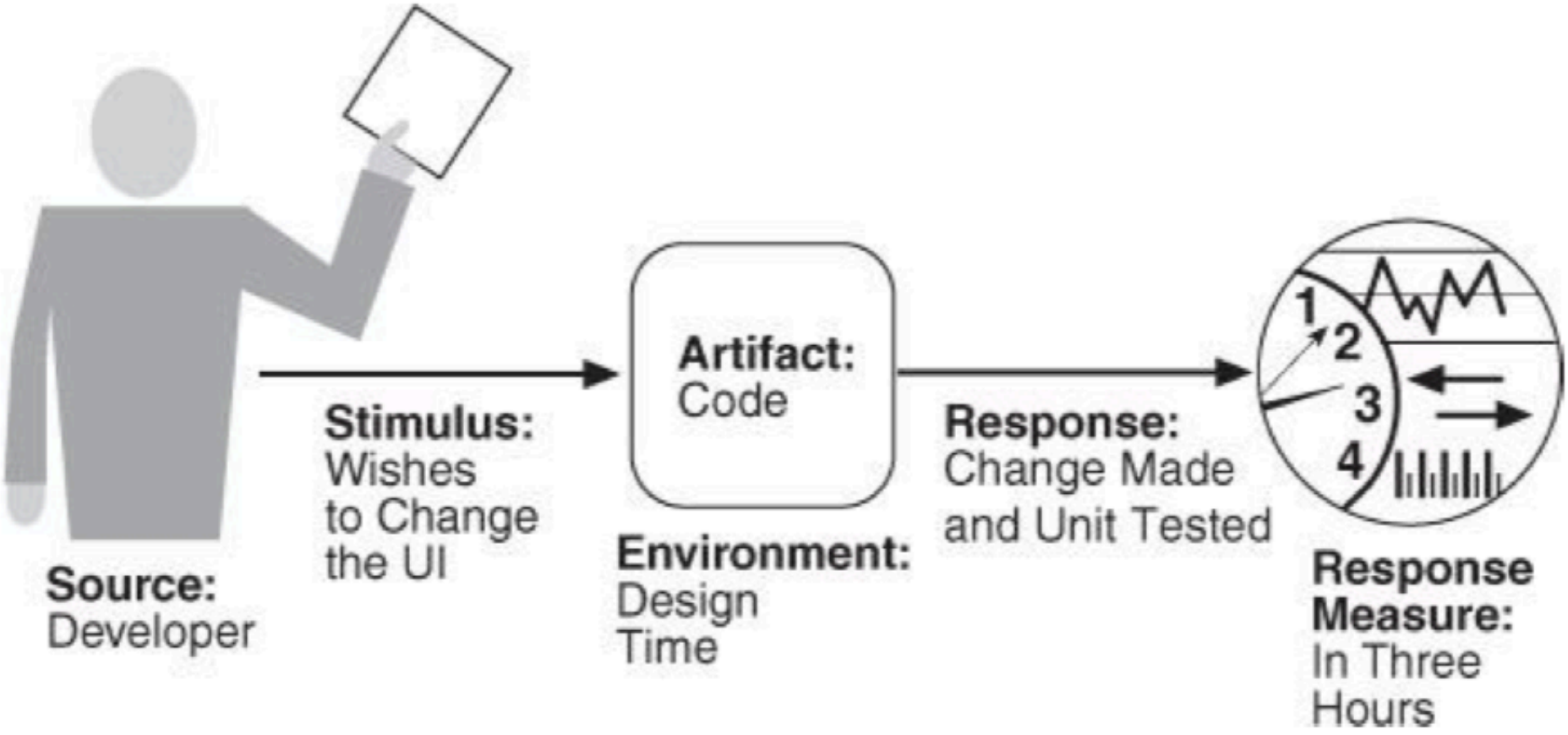


General modifiability scenario

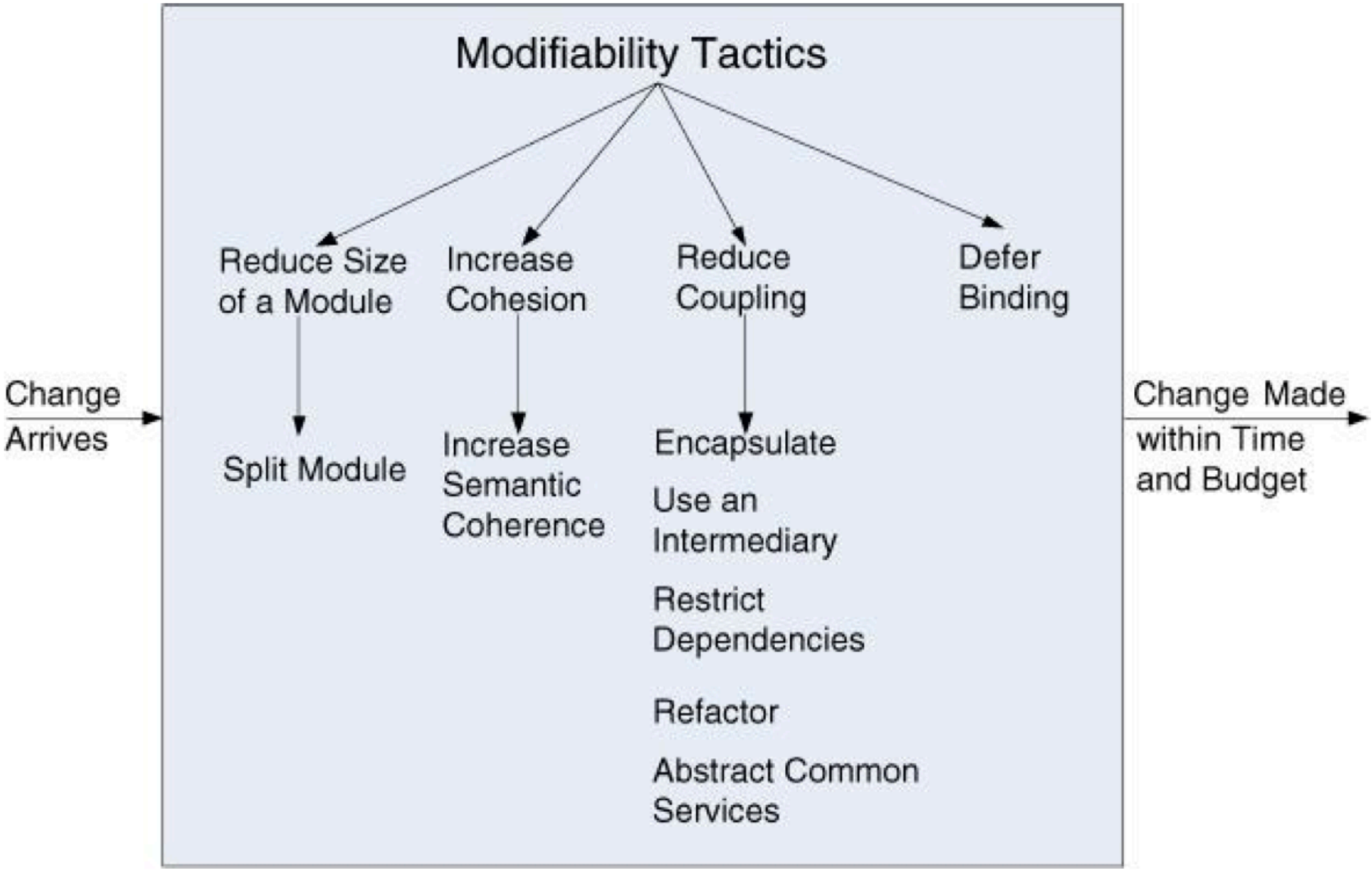


Source	Stimulus	Environment	Response	Measure
End-user	Add / delete / modify functionality, quality attribute, capacity or technology	Runtime	Make modification	Cost in effort
Developer		Compile time	Test modification	Cost in money
System-administrator		Build time	Deploy modification	Cost in time
		Initiation time		Cost in number, size, complexity of affected artifacts
		Design time		Extent affects other system functions or qualities
				New defects introduced

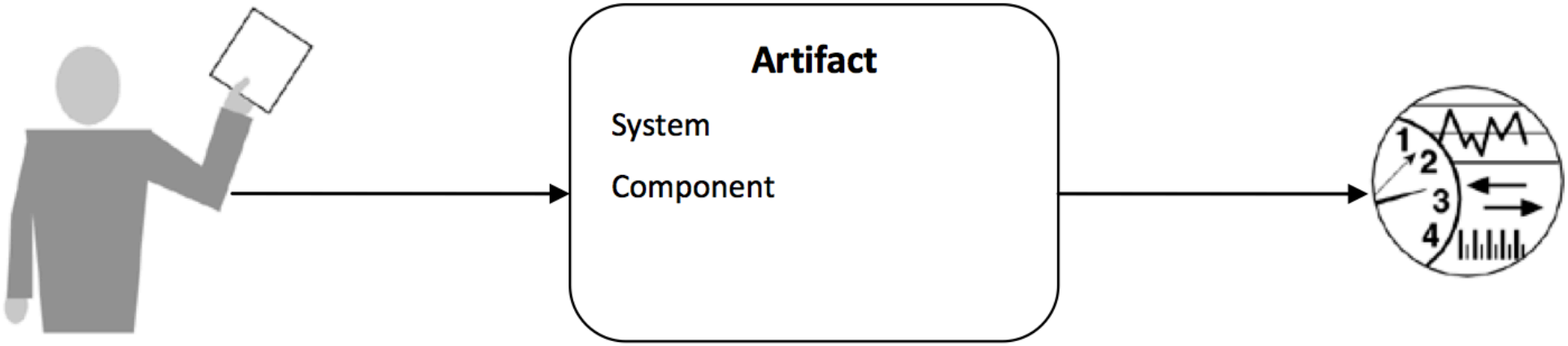
Sample modifiability scenario



Modifiability tactics

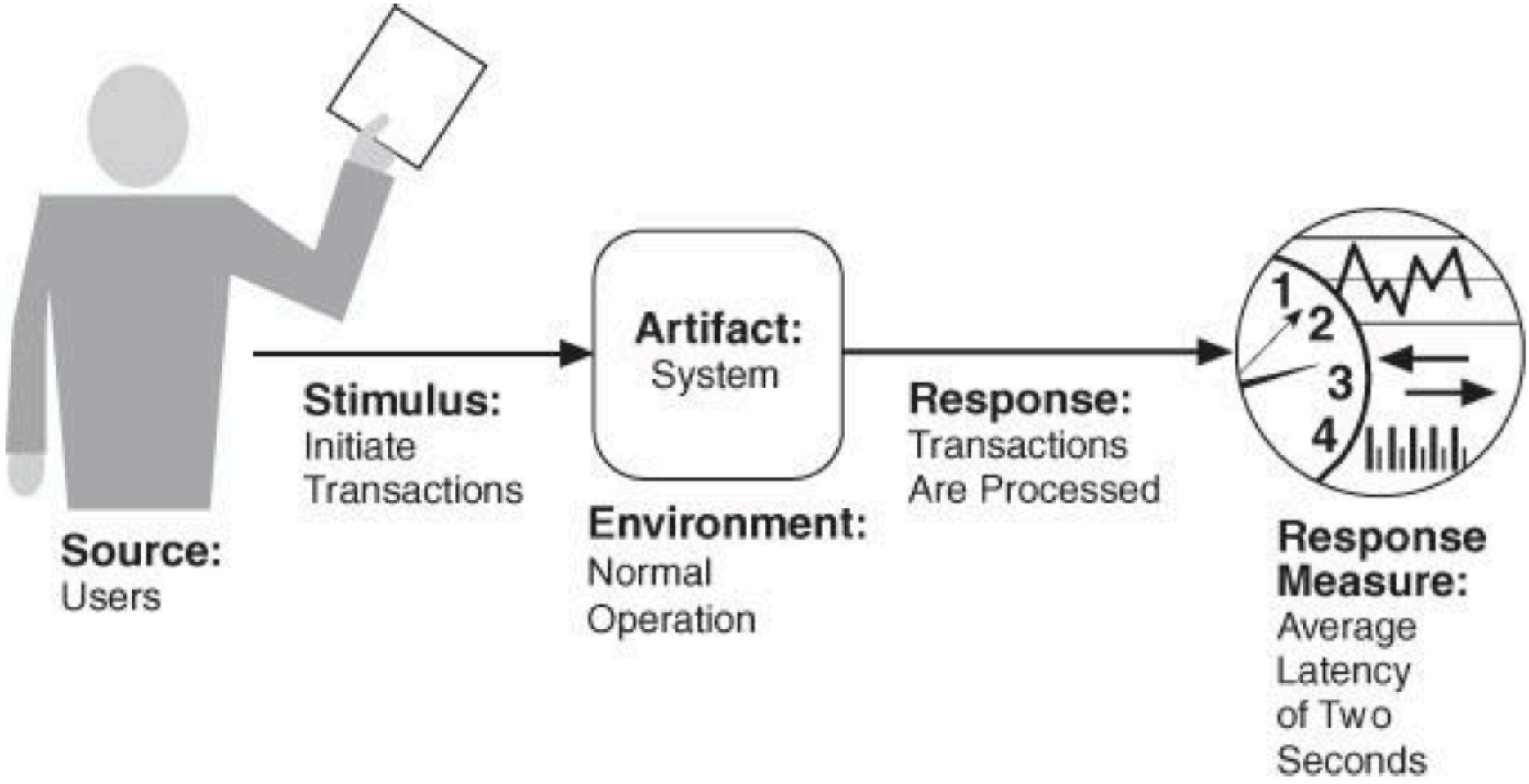


General performance scenario

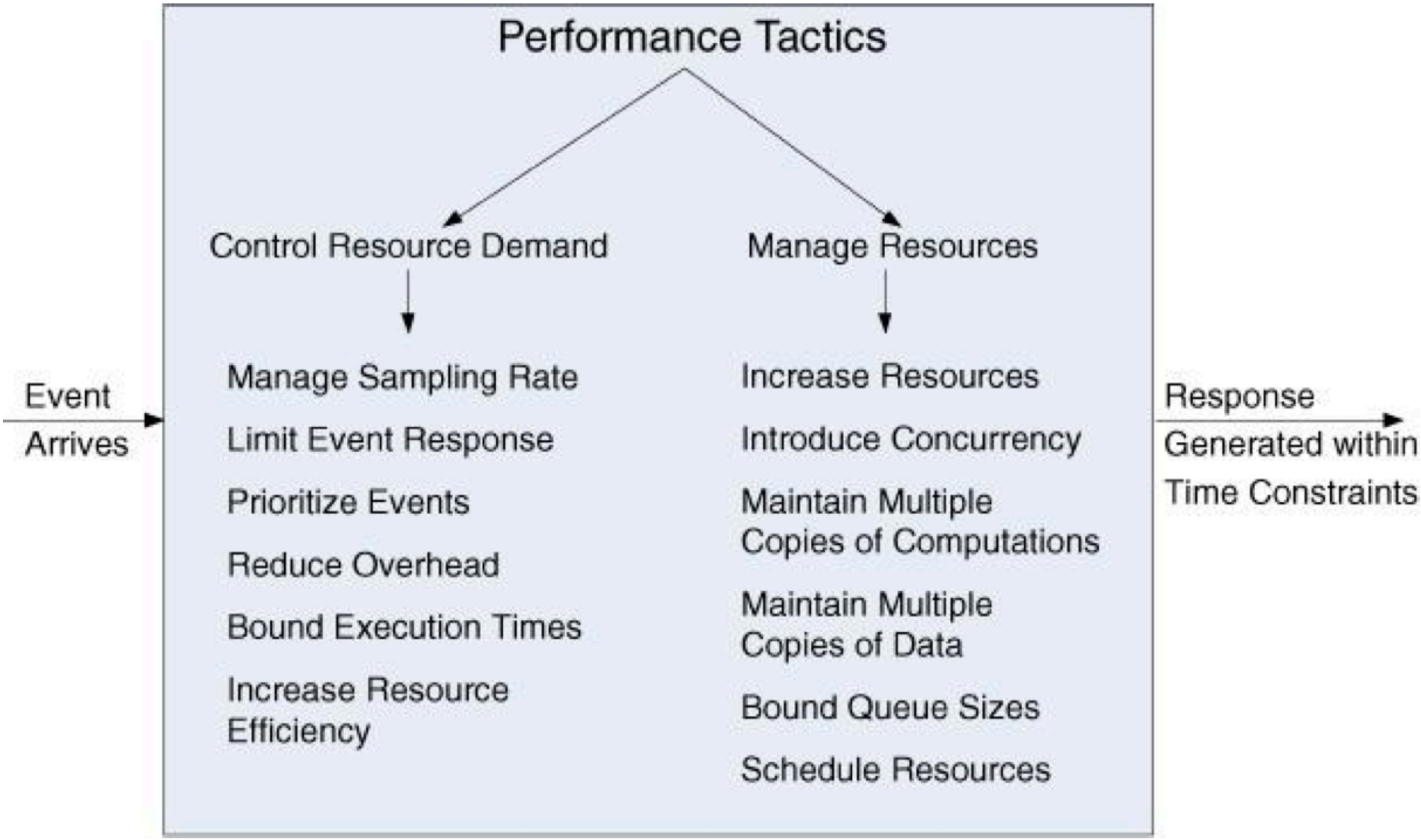


Source	Stimulus	Environment	Response	Measure
Internal to the system	Periodic events	Normal mode	Process events	Latency
External to the system	Sporadic events	Overload mode	Change level of service	Deadline
	Bursty events	Reduced capacity mode		Throughput
	Stochastic events	Emergency mode		Jitter
		Peak mode		Miss rate
				Data loss

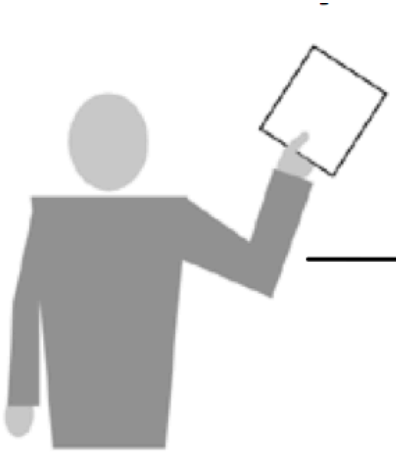
Sample performance scenario

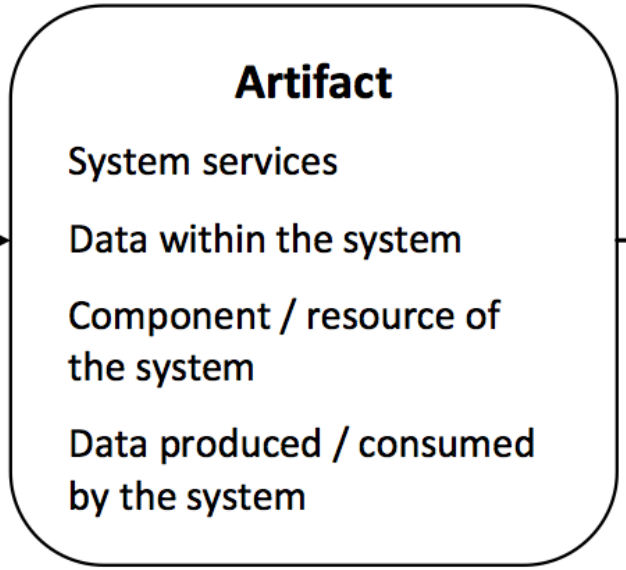


Performance tactics



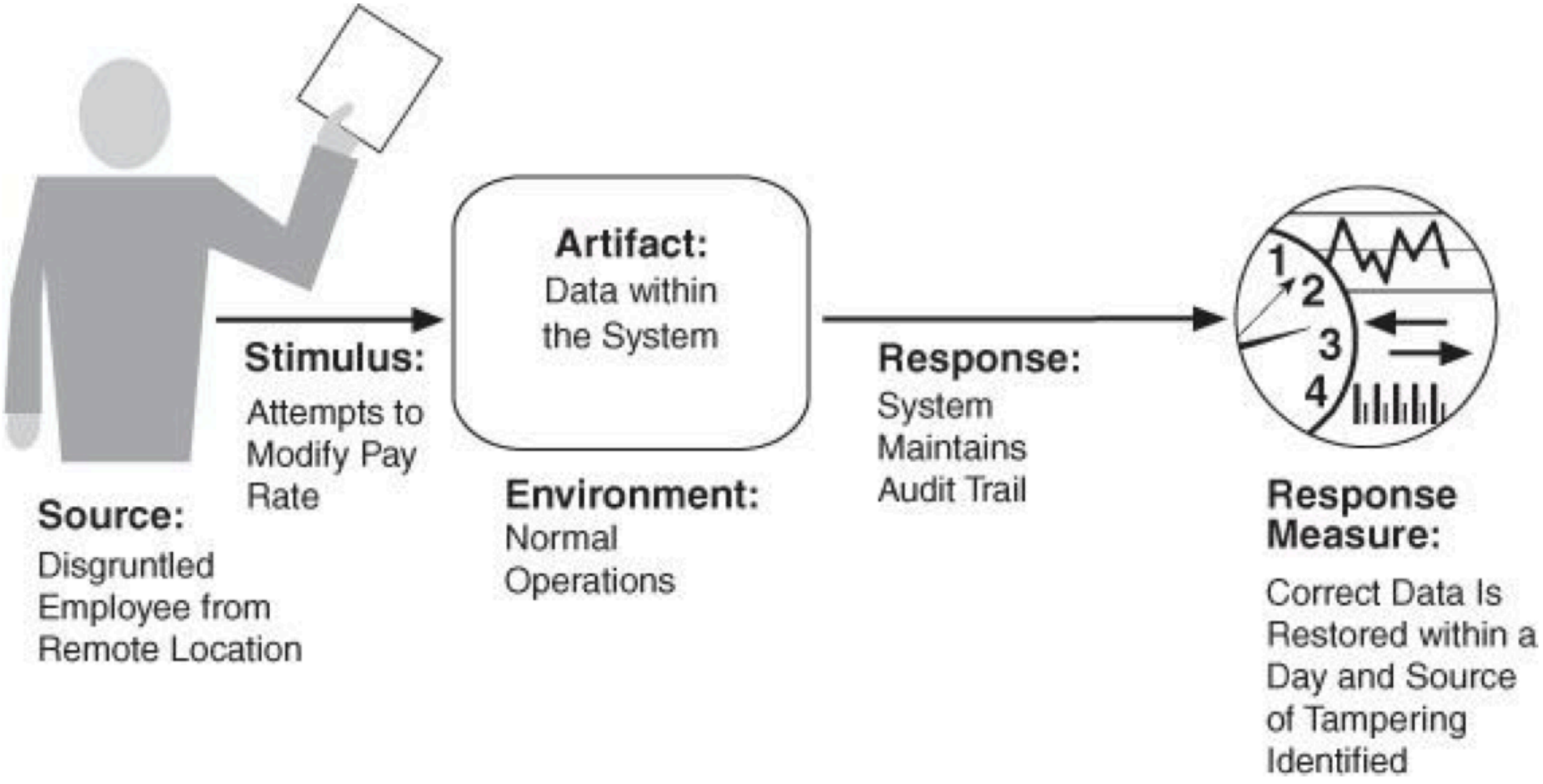
General security scenario



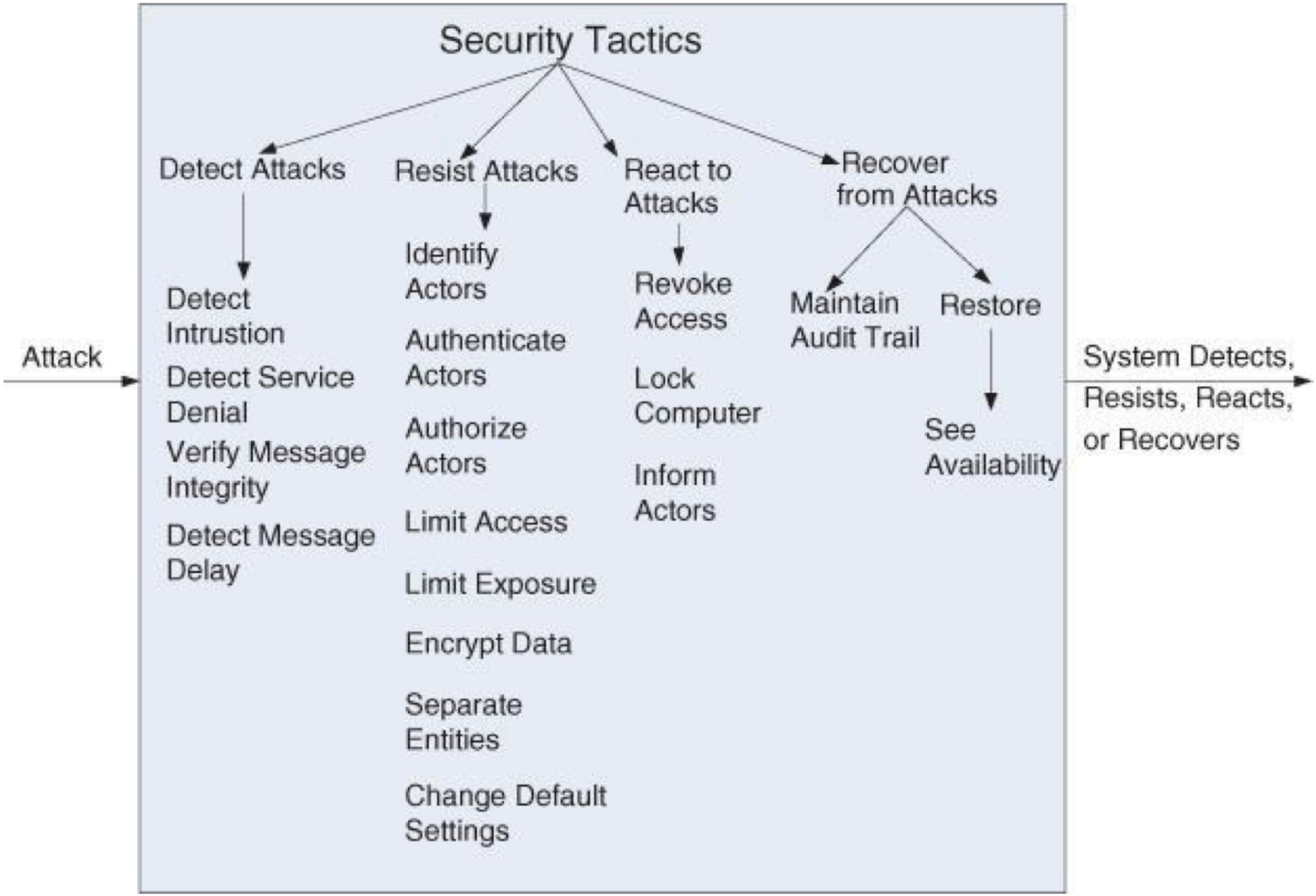


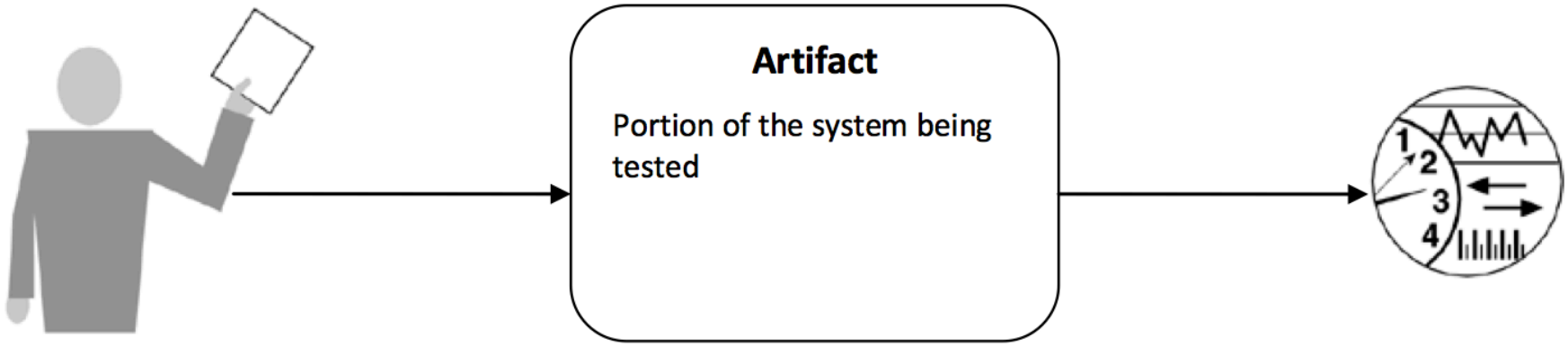
Source	Stimulus	Environment	Response	Measure
Identified user	Attempt to display data	Normal mode	Process events	Latency
Unknown user	Attempt to modify data	Overload mode	Change level of service	Deadline
Hacker from outside the organization	Attempt to delete data	Reduced capacity mode		Throughput
Hacker from inside the organization	Access system services	Emergency mode		Jitter
	Change system's behavior	Peak mode		Miss rate
	Reduce availability			Data loss

Sample security scenario



Security tactics

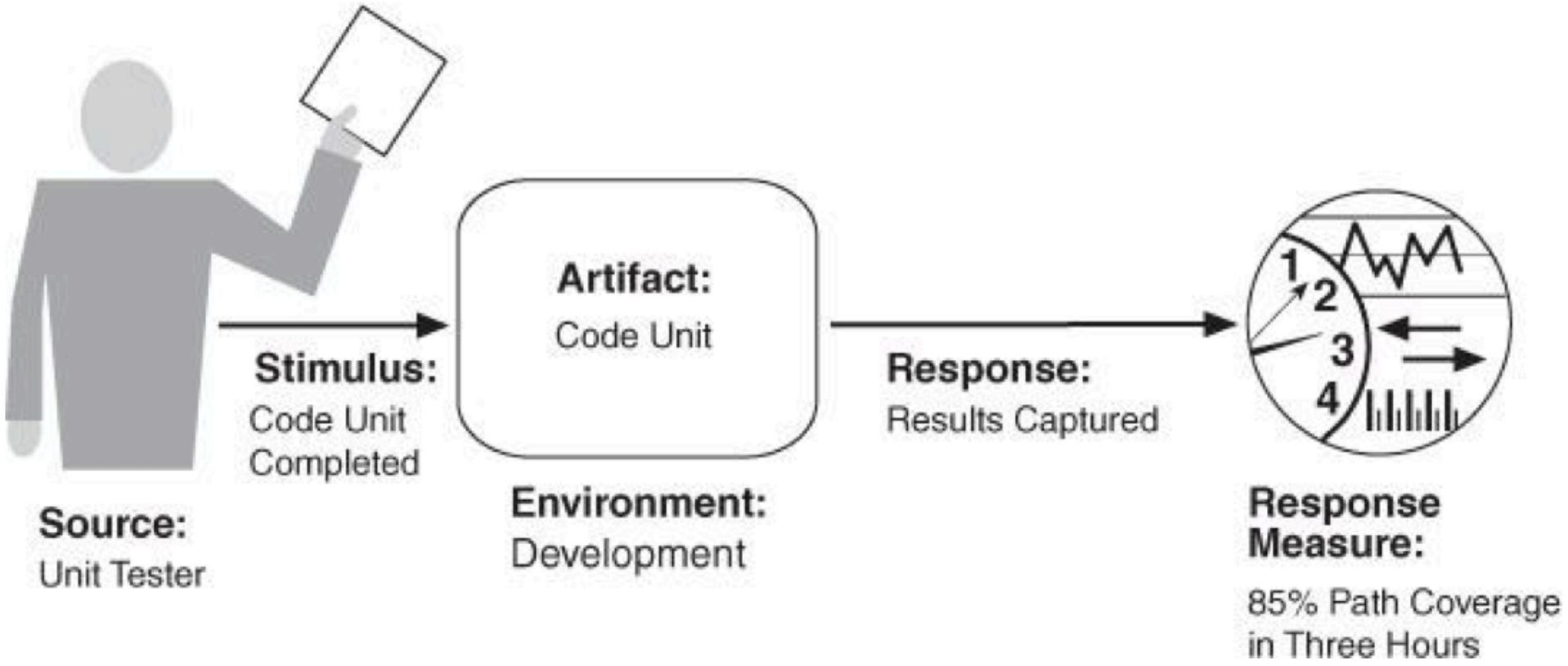




Source	Stimulus	Environment	Response	Measure
Unit tester	Execution of tests due to completion of code increment	Design time	Execute test suite & capture results	Effort to find fault
Integration tester		Development time	Capture cause of fault	Effort to achieve coverage %
System tester		Compile time	Control & monitor state of the system	Probability of fault being revealed by next test
Acceptance tester		Integration time		Time to perform tests
End user		Deployment time		Effort to detect faults
Automated testing tools		Run time		Length of longest dependency chain
				Time to prepare test environment
				Reduction in risk exposure

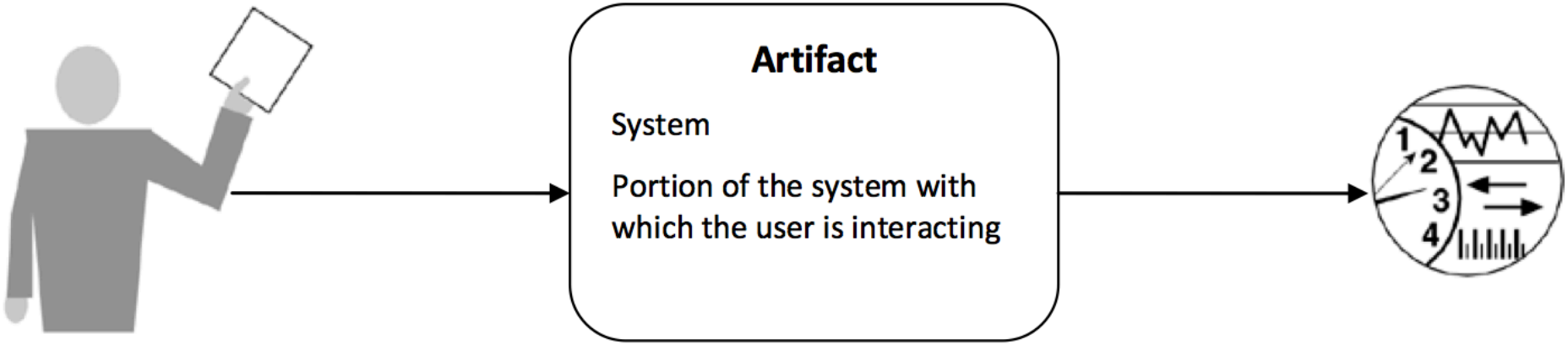
General testability scenario

Sample testability scenario




Testability Tactics
Control and Observe System State
Limit Complexity
Tests Executed
Specialized Interfaces
Record/ Playback
Localize State Storage
Abstract Data Sources
Sandbox
Executable Assertions
Limit Structural Complexity
Limit Nondeterminism
Faults Detected

General usability scenario



Source

End user (possibly special role)

Stimulus

Use the system efficiently

Learn to use the system

Minimize impact of errors

Adapt the system

Configure the system

Environment

Runtime

Configuration time

Response

Provide features needed

Anticipate the user's needs

Measure

Task time

Number of errors

Number of tasks accomplished

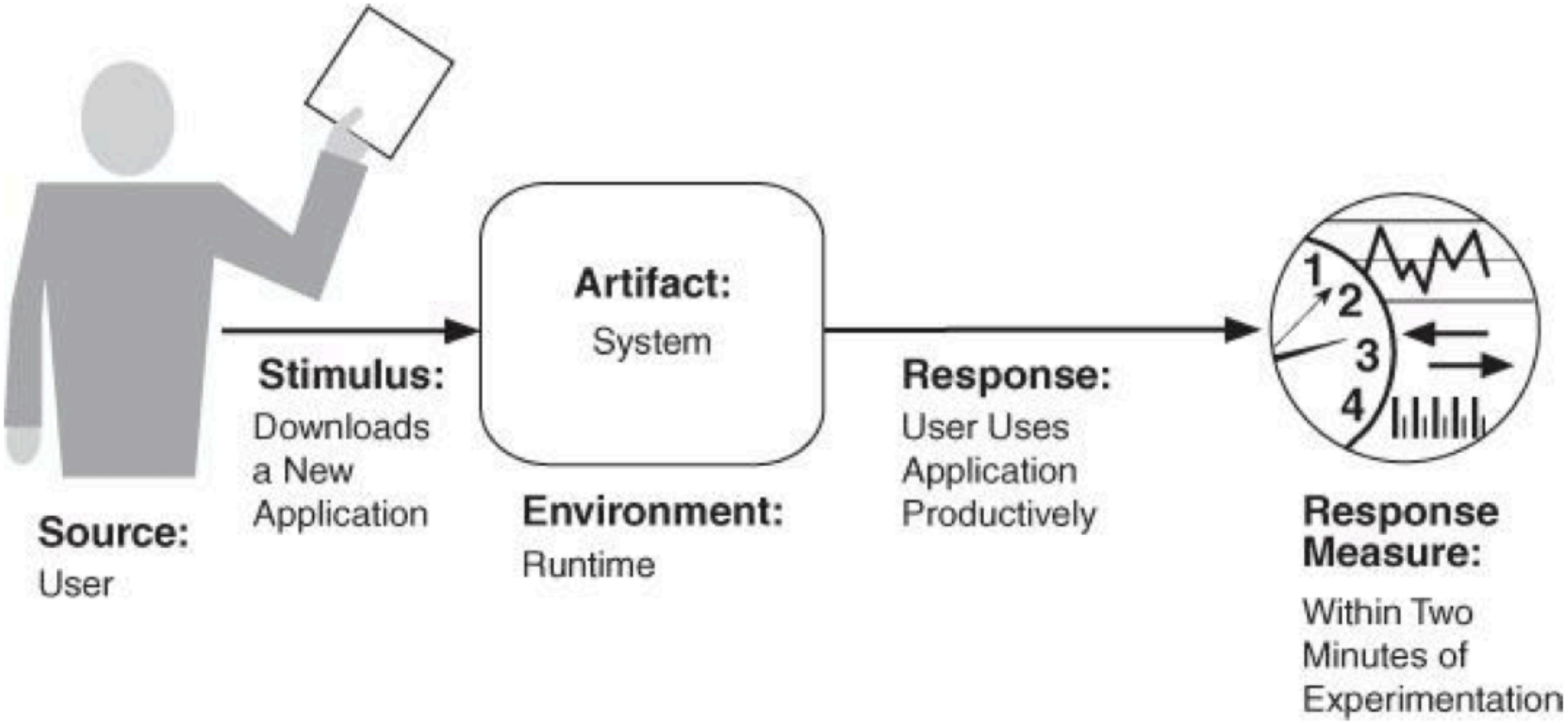
User satisfaction

Gain of user knowledge

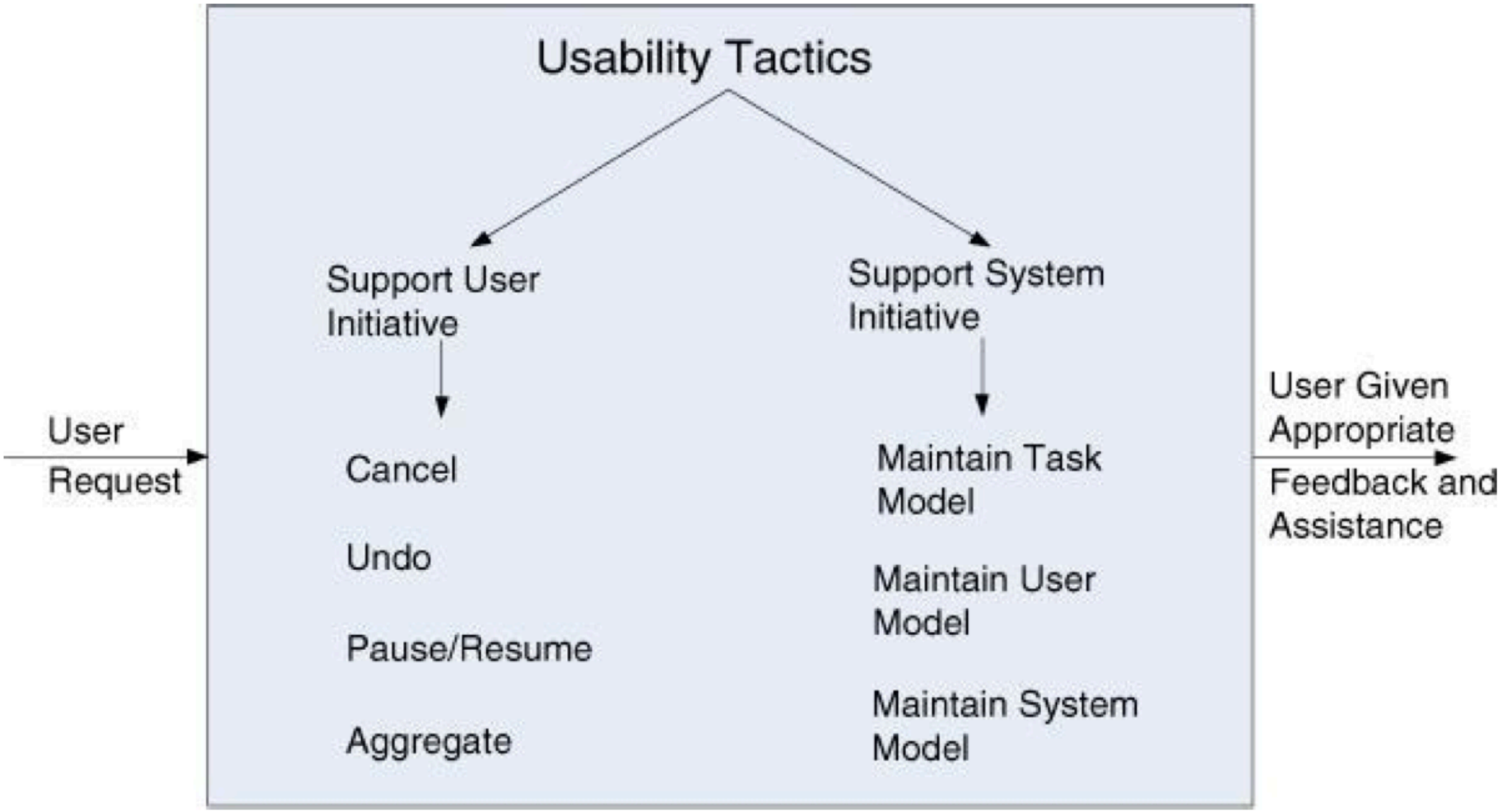
Ratio of successful operations to total operations

Amount of time / data lost when error occurs

Sample usability scenario



Usability tactics



References

- Bass, *Software architecture in practice*, 3° ed
  http://www.ece.ubc.ca/~matei/EECE417/BASS/

Questions?