Understanding Quality Attributes

1. Functionality and Architecture

- Functionality and quality attributes are orthogonal.
- What is functionality? It is the ability of the system to do the work for which it was intended. A task requires that many or most of the system's elements work in a coordinated manner to complete the job. If the elements have not been assigned the correct responsibilities or have not been endowed with the correct facilities for coordinating with other elements
- Functionality may be achieved through the use of any of a number of possible structures. In fact, if functionality were the only requirement, the system could exist as a single monolithic module with no internal structure at all. Instead, it is decomposed into modules to make it understandable and to support a variety of other purposes. In this way, functionality is largely independent of structure. Software architecture constrains its allocation to structure when other quality attributes are important.

2. Architecture and Quality Attributes

- Achieving quality attributes must be considered throughout design, implementation, and deployment.
- Architecture is critical to the realization of many qualities of interest in a system, and these qualities should be designed in and can be evaluated at the architectural level.
- Architecture, by itself, is unable to achieve qualities. It provides the foundation for achieving quality, but this foundation will be to no avail if attention is not paid to the details.
- Types of quality attributes:
  1. Qualities of the system
  2. Business qualities
  3. Qualities that are about the architecture itself

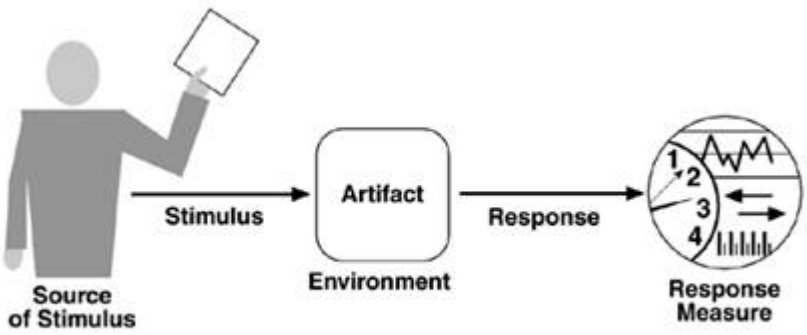
3. System Quality Attributes

QUALITY ATTRIBUTE SCENARIOS

A quality attribute scenario is a quality-attribute-specific requirement. It consists of six parts.

- **Source of stimulus**. This is some entity (a human, a computer system, or any other actuator) that generated the stimulus.
- **Stimulus**. The stimulus is a condition that needs to be considered when it arrives at a system.
- **Environment**. The stimulus occurs within certain conditions. The system may be in an overload condition or may be running when the stimulus occurs, or some other condition may be true.
- **Artifact**. Some artifact is stimulated. This may be the whole system or some pieces of it.
- **Response**. The response is the activity undertaken after the arrival of the stimulus.
- **Response measure**. When the response occurs, it should be measurable in some fashion so that the requirement can be tested.

Quality attribute parts

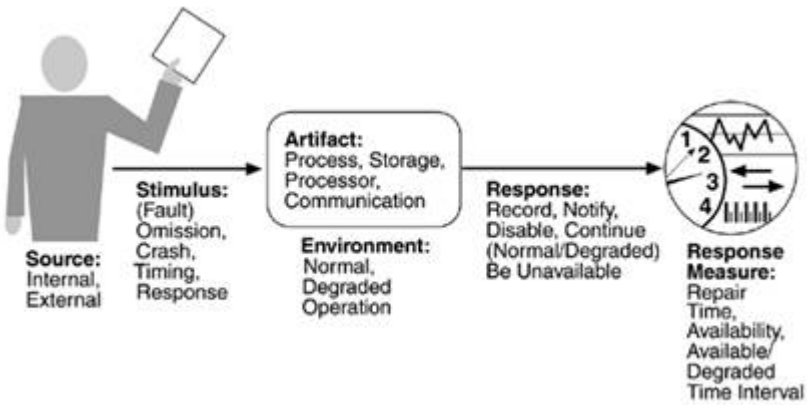


4. Quality Attribute Scenarios in Practice

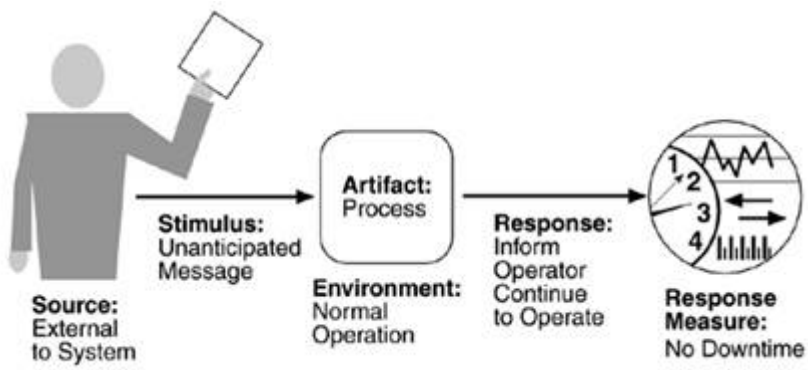
AVAILABILITY

- Availability is concerned with system failure and its associated consequences.
- A system failure occurs when the system no longer delivers a service consistent with its specification.
- Such a failure is observable by the system's users—either humans or other systems.

Availability General Scenarios



Sample availability scenario



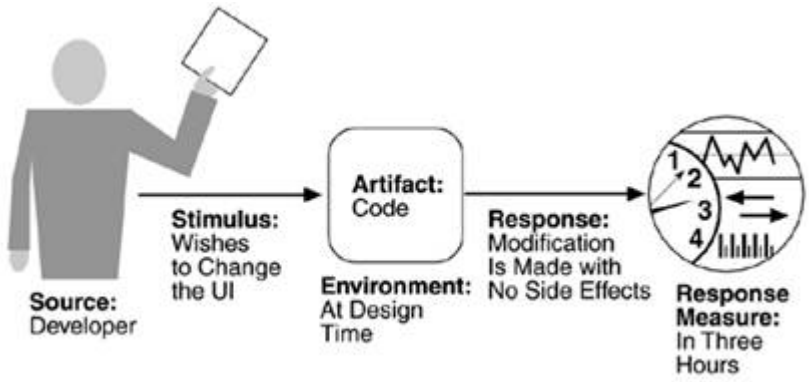
MODIFIABILITY

Modifiability is about the cost of change.

Modifiability General Scenario Generation

Portion of Scenario	Possible Values
Source	End user, developer, system administrator
Stimulus	Wishes to add/delete/modify/vary functionality, quality attribute, capacity
Artifact	System user interface, platform, environment; system that interoperates with target system
Environment	At runtime, compile time, build time, design time
Response	Locates places in architecture to be modified; makes modification without affecting other functionality; tests modification; deploys modification
Response Measure	Cost in terms of number of elements affected, effort, money; extent to which this affects other functions or quality attributes

Sample modifiability scenario



PERFORMANCE

- Performance is about timing. Events (interrupts, messages, requests from users, or the passage of time) occur, and the system must respond to them.
- One of the things that make performance complicated is the number of event sources and arrival patterns.
- A performance scenario begins with a request for some service arriving at the system. Satisfying the request requires resources to be consumed. While this is happening the system may be simultaneously servicing other requests.
- An arrival pattern for events may be characterized as either periodic or stochastic.

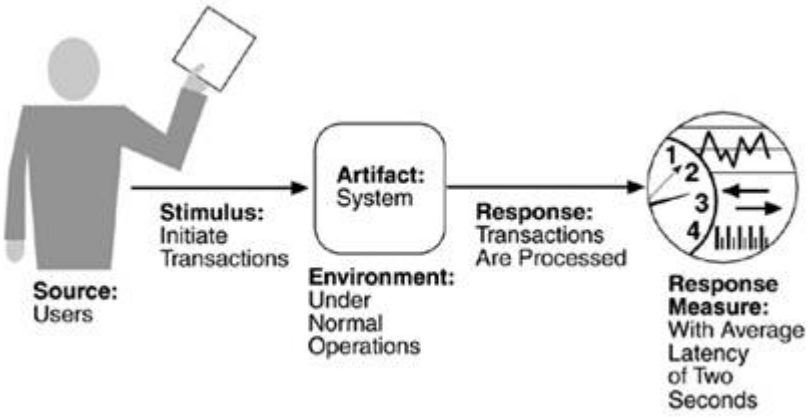
Performance General Scenario Generation

Portion of Scenario	Possible Values
Source	One of a number of independent sources, possibly from within system
Stimulus	Periodic events arrive; sporadic events arrive; stochastic events arrive
Artifact	System
Environment	Normal mode; overload mode

Performance General Scenario Generation

Portion of Scenario	Possible Values
Response	Processes stimuli; changes level of service
Response Measure	Latency, deadline, throughput, jitter, miss rate, data loss

Sample performance scenario



SECURITY

- Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legitimate users.
- Security can be characterized as a system providing nonrepudiation, confidentiality, integrity, assurance, availability, and auditing.

1. **Nonrepudiation** is the property that a transaction (access to or modification of data or services) cannot be denied by any of the parties to it.
2. **Confidentiality** is the property that data or services are protected from unauthorized access.
3. **Integrity** is the property that data or services are being delivered as intended.
4. **Assurance** is the property that the parties to a transaction are who they purport to be.
5. **Availability** is the property that the system will be available for legitimate use.
6. **Auditing** is the property that the system tracks activities within it at levels sufficient to reconstruct them.

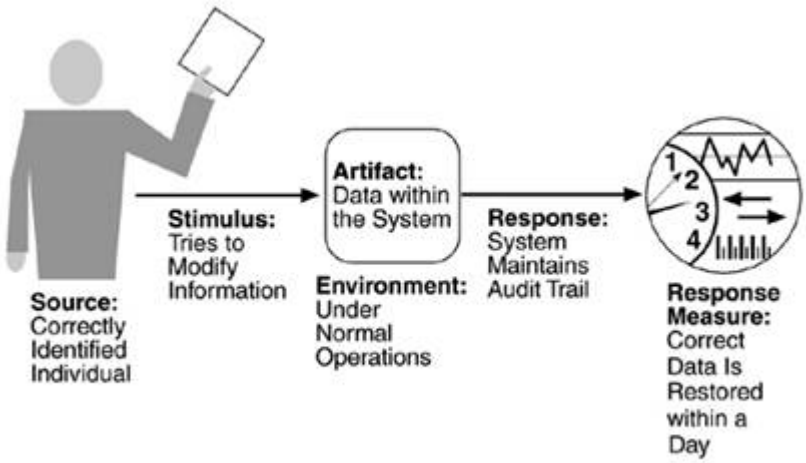
Security General Scenario Generation

Portion of Scenario	Possible Values
Source	Individual or system that is correctly identified, identified incorrectly, of unknown identity who is internal/external, authorized/not authorized with access to limited resources, vast resources
Stimulus	Tries to display data, change/delete data, access system services, reduce availability to system services
Artifact	System services; data within system

Security General Scenario Generation

Portion of Scenario	Possible Values
Environment	Either online or offline, connected or disconnected, firewalled or open
Response	Authenticates user; hides identity of the user; blocks access to data and/or services; allows access to data and/or services; grants or withdraws permission to access data and/or services; records access/modifications or attempts to access/modify data/services by identity; stores data in an unreadable format; recognizes an unexplainable high demand for services, and informs a user or another system, and restricts availability of services
Response Measure	Time/effort/resources required to circumvent security measures with probability of success; probability of detecting attack; probability of identifying individual responsible for attack or access/modification of data and/or services; percentage of services still available under denial-of-services attack; restore data/services; extent to which data/services damaged and/or legitimate access denied

Sample security scenario



TESTABILITY

- Software testability refers to the ease with which software can be made to demonstrate its faults through testing.
- Testability refers to the probability that it will fail on its next test execution.
- For a system to be properly testable, it must be possible to control each component's internal state and inputs and then to observe its outputs.

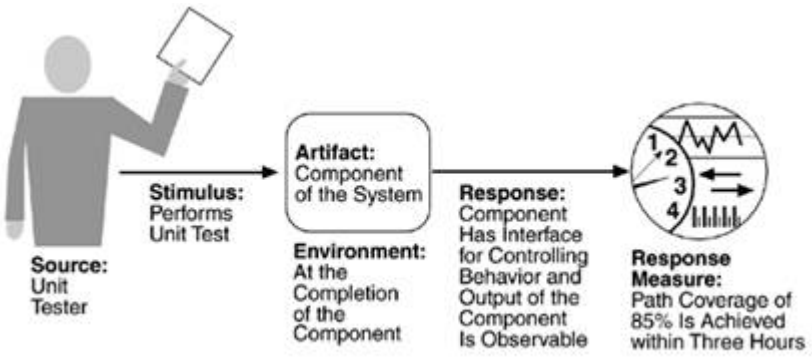
Testability General Scenario Generation

Portion of Scenario	Possible Values
Source	Unit developer, Increment integrator, System verifier, Client acceptance tester, System user
Stimulus	Analysis, architecture, design, class, subsystem integration completed; system delivered
Artifact	Piece of design, piece of code, complete application
Environment	At design time, at development time, at compile time, at deployment time

Testability General Scenario Generation

Portion of Scenario	Possible Values
Response	Provides access to state values; provides computed values; prepares test environment
Response Measure	Percent executable statements executed Probability of failure if fault exists Time to perform tests Length of longest dependency chain in a test Length of time to prepare test environment

Sample testability scenario



USABILITY

Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides. It can be broken down into the following areas:

- **Learning system features**. If the user is unfamiliar with a particular system or a particular aspect of it, what can the system do to make the task of learning easier?
- **Using a system efficiently**. What can the system do to make the user more efficient in its operation?
- **Minimizing the impact of errors**. What can the system do so that a user error has minimal impact?
- **Adapting the system to user needs**. How can the user (or the system itself) adapt to make the user's task easier?
- **Increasing confidence and satisfaction**. What does the system do to give the user confidence that the correct action is being taken?

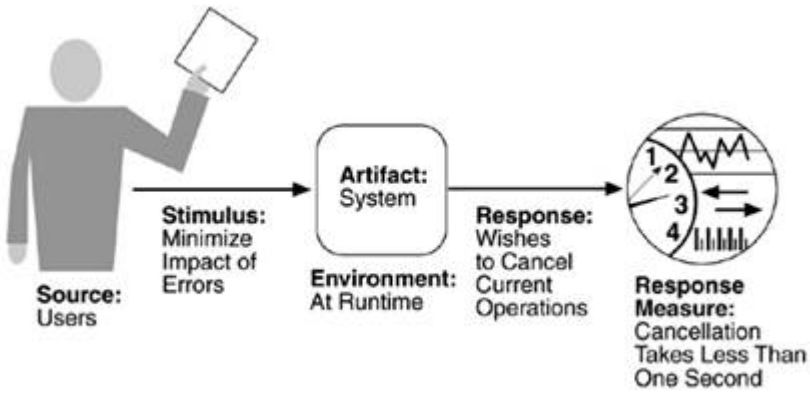
Usability General Scenario Generation

Portion of Scenario	Possible Values
Source	End user
Stimulus	Wants to learn system features; use system efficiently; minimize impact of errors; adapt system; feel comfortable
Artifact	System
Environment	At runtime or configure time

Usability General Scenario Generation

Portion of Scenario	Possible Values
Response	System provides one or more of the following responses: to support "**learn system features**": help system is sensitive to context; interface is familiar to user; interface is usable in an unfamiliar context to support "**use system efficiently**": aggregation of data and/or commands; re-use of already entered data and/or commands; support for efficient navigation within a screen; distinct views with consistent operations; comprehensive searching; multiple simultaneous activities to "**minimize impact of errors**": undo, cancel, recover from system failure, recognize and correct user error, retrieve forgotten password, verify system resources to "**adapt system**": customizability; internationalization to "**feel comfortable**": display system state; work at the user's pace
Response Measure	Task time, number of errors, number of problems solved, user satisfaction, gain of user knowledge, ratio of successful operations to total operations, amount of time/data lost

Sample usability scenario



COMMUNICATING CONCEPTS USING GENERAL SCENARIOS

Quality Attribute Stimuli

Quality Attribute	Stimulus
Availability	Unexpected event, nonoccurrence of expected event
Modifiability	Request to add/delete/change/vary functionality, platform, quality attribute, or capacity
Performance	Periodic, stochastic, or sporadic
Security	Tries to display, modify, change/delete information, access, or reduce availability to system services
Testability	Completion of phase of system development
Usability	Wants to learn system features, use a system efficiently, minimize the impact of errors, adapt the system, feel comfortable

5. Other System Quality Attributes

- Scalability
- Portability
- Interoperability

6. Business Qualities

- Time to market.
- Cost and benefit.
- Projected lifetime of the system.
- Targeted market.
- Rollout schedule.
- Integration with legacy systems.

7. Architecture Qualities

- Conceptual integrity is the underlying theme or vision that unifies the design of the system at all levels. The architecture should do similar things in similar ways. Conceptual integrity is the most important consideration in system design.
- Correctness and completeness are essential for the architecture to allow for all of the system's requirements and runtime resource constraints to be met. A formal evaluation is the architect's best hope for a correct and complete architecture.
- Buildability allows the system to be completed by the available team in a timely manner and to be open to certain changes as development progresses.