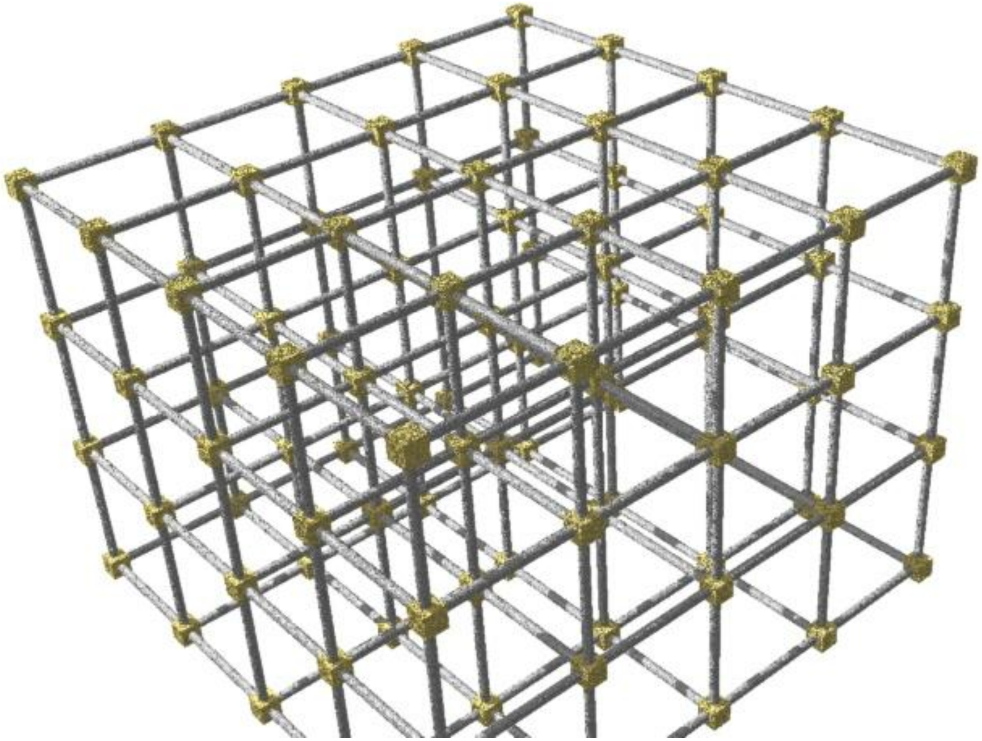
# Scenario Generation Tables



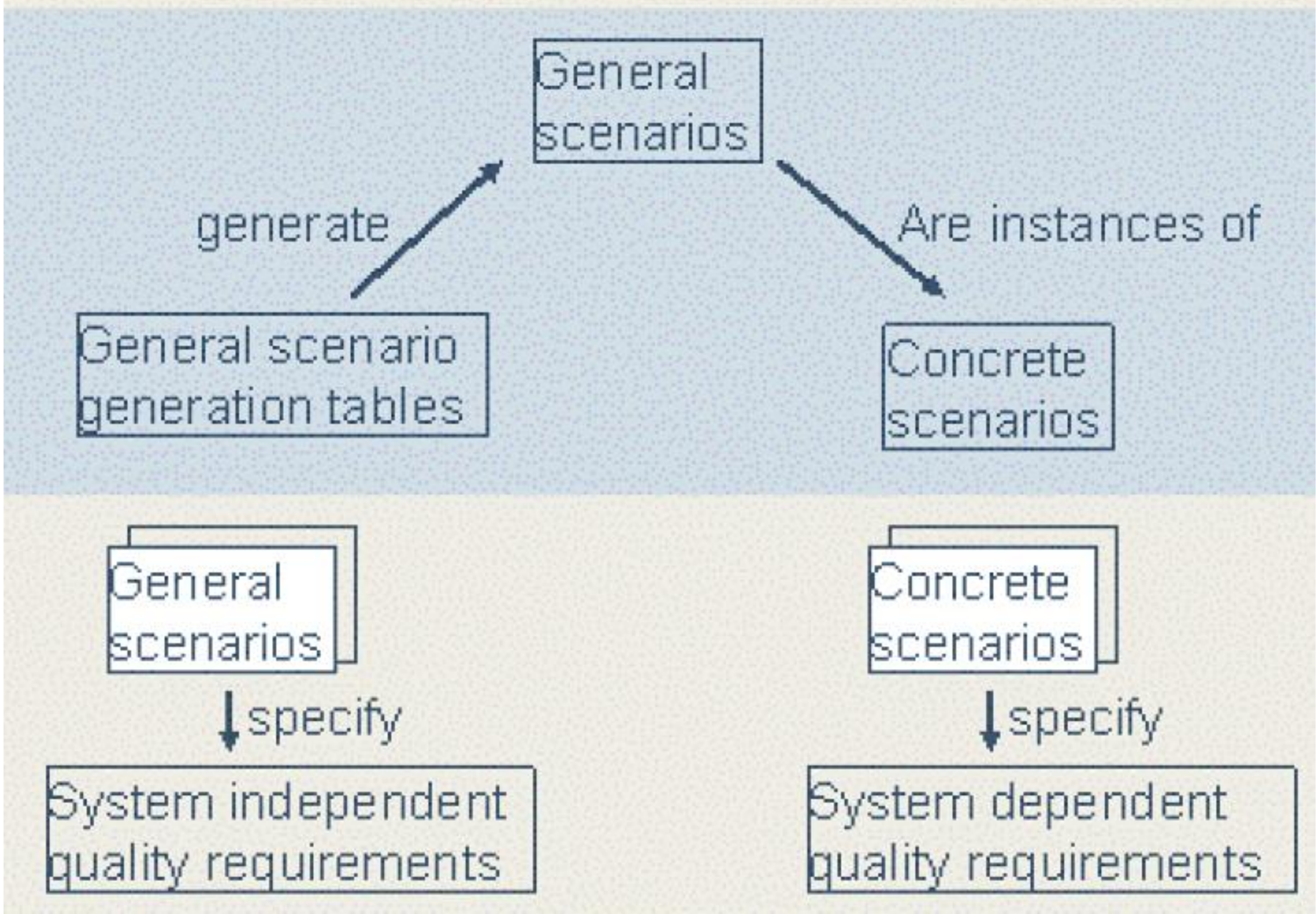
The general scenario approach provides a framework for generating a large number of system-independent, quality-attribute-specific scenarios.

- A *general scenario* is a precise system-independent specification of a type of quality attribute Requirement

*A periodic event from an independent source arrives at the system under normal conditions. The system has to process the stimulus within certain latency.*

- The general scenario provides a template for a creating a system-specific *concrete scenarios*.

- *An event from sensor X arrives every 10 milliseconds (ms) at a system that operates under normal conditions. The system has to process the stimulus within 1 ms.*



General scenarios

generate

Are instances of

General scenario generation tables

Concrete scenarios

General scenarios

specify

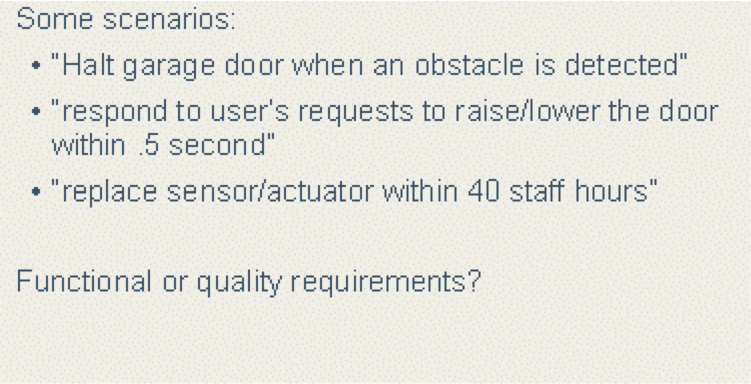
System independent quality requirements

Concrete scenarios

specify

System dependent quality requirements

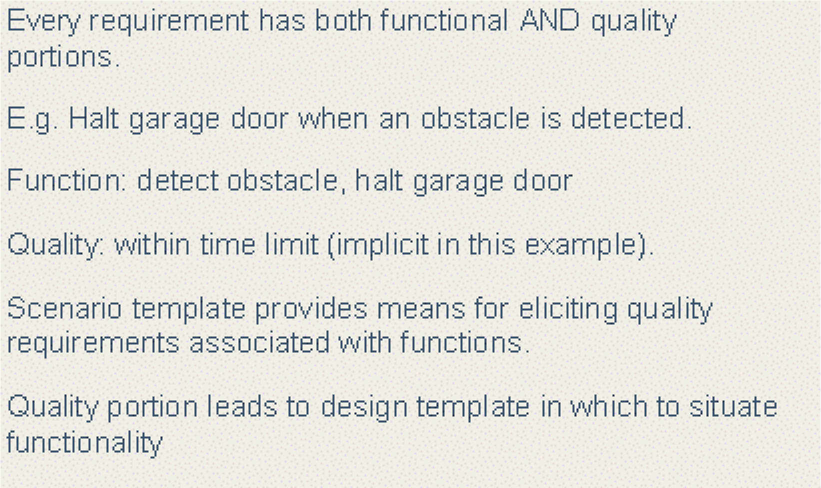
- A general scenario is system independent and can, potentially, pertain to any system.



Some scenarios:

- "Halt garage door when an obstacle is detected"
- "respond to user's requests to raise/lower the door within .5 second"
- "replace sensor/actuator within 40 staff hours"

Functional or quality requirements?



Every requirement has both functional AND quality portions.

E.g. Halt garage door when an obstacle is detected.

Function: detect obstacle, halt garage door

Quality: within time limit (implicit in this example).

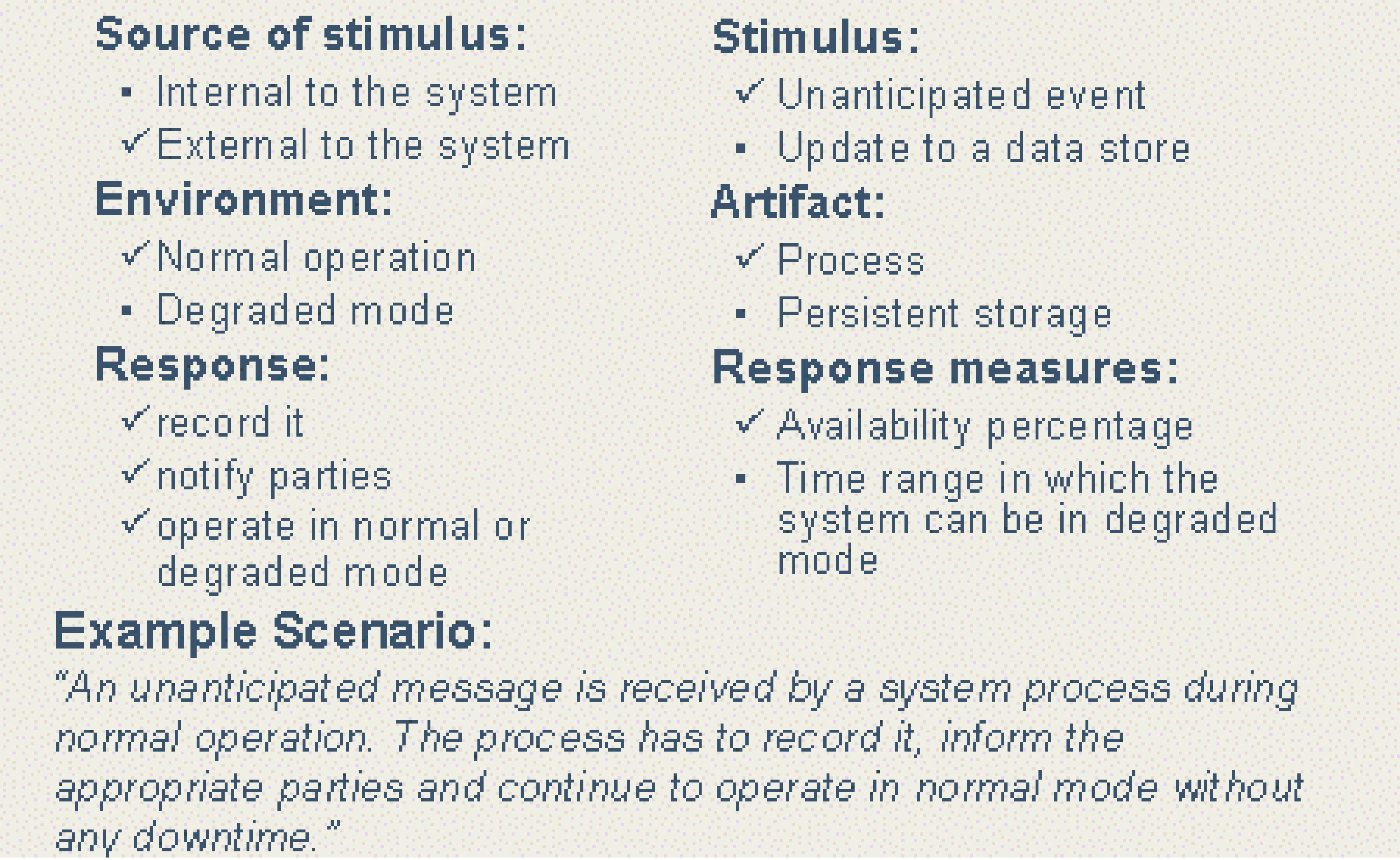
Scenario template provides means for eliciting quality requirements associated with functions.

Quality portion leads to design template in which to situate functionality

To make the general scenarios useful, you must make them system specific. Concrete Quality Attribute Scenarios is specific to the particular system under consideration.

# Table for Generation of General Availability Scenario

| Portion of Scenario | Possible Values |
|---|---|
| Source | Internal to the system; external to the system |
| Stimulus | Fault; omission, crash, timing, response |
| Artifact | System’s processors, communication channels, persistent storage processes |
| Environment | Normal operation; degraded mode (i.e., fewer features, a fall-back solution. |
| Response | System should detect event and do one or more of the following:<br>• record it<br>• notify appropriate parties, including the user and other systems<br>• disable sources of events that cause fault or failure<br>• be unavailable for a prespecified interval<br>• continue to operate in normal or degraded mode |
| Response Measure | Time interval when the system must be available; availability time; time interval in which system can be in degraded mode; repair time |



**Source of stimulus:**

- ▪ Internal to the system
- ✓ External to the system

**Environment:**

- ✓ Normal operation
- ▪ Degraded mode

**Response:**

- ✓ record it
- ✓ notify parties
- ✓ operate in normal or degraded mode

**Stimulus:**

- ✓ Unanticipated event
- ▪ Update to a data store

**Artifact:**

- ✓ Process
- ▪ Persistent storage

**Response measures:**

- ✓ Availability percentage
- ▪ Time range in which the system can be in degraded mode

**Example Scenario:**

*"An unanticipated message is received by a system process during normal operation. The process has to record it, inform the appropriate parties and continue to operate in normal mode without any downtime."*

Generate a possible general scenario by choosing one or more entries from each list and combining them

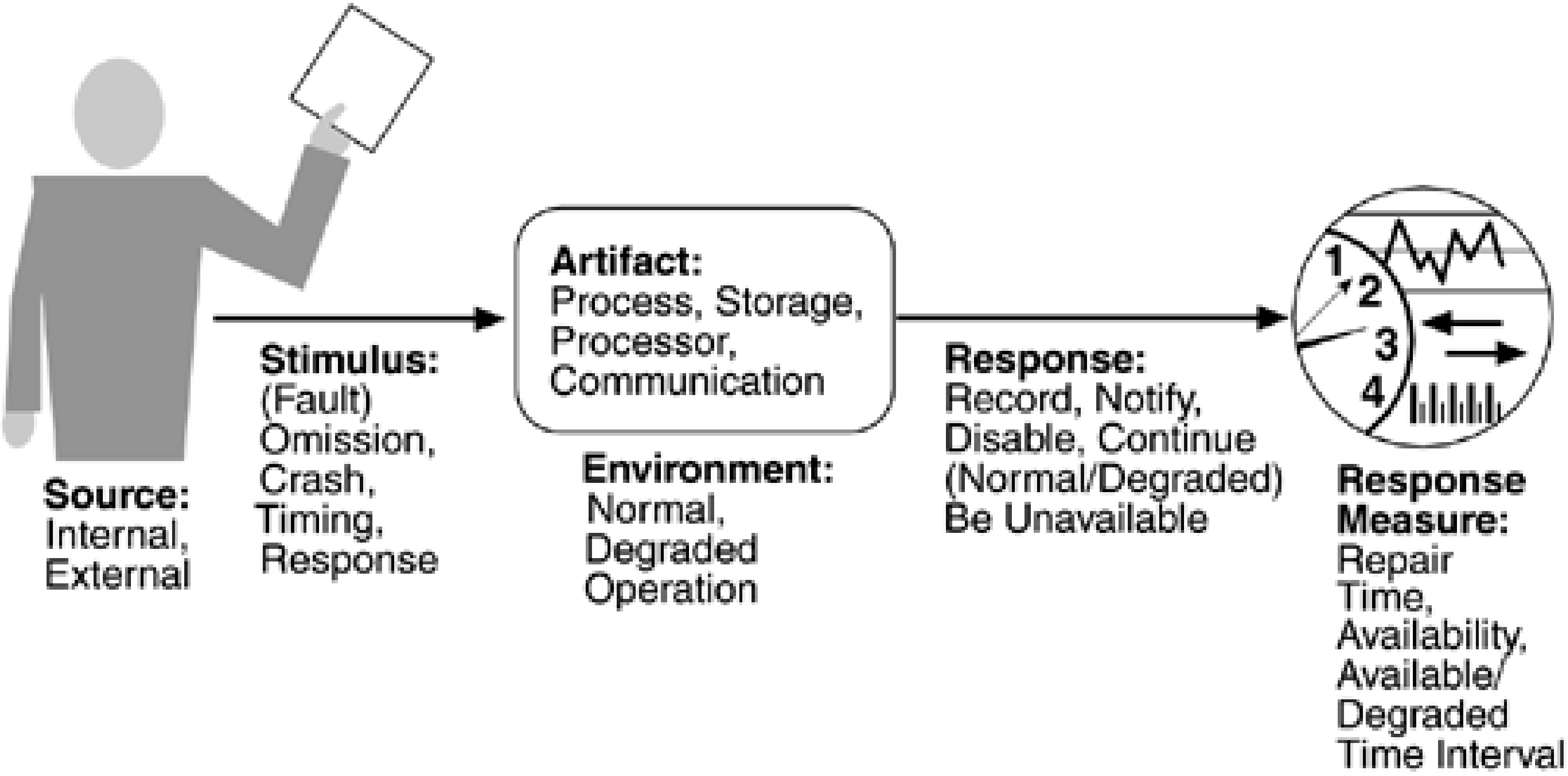
Not all:

- general scenarios are relevant to specific system
- generated scenarios make sense

Make each scenario system specific (concrete scenario)

May be multiple concrete scenarios for each general scenario. e.g., modify function.

# General Scenario for Availability



**Source:** Internal, External

**Stimulus:** (Fault) Omission, Crash, Timing, Response

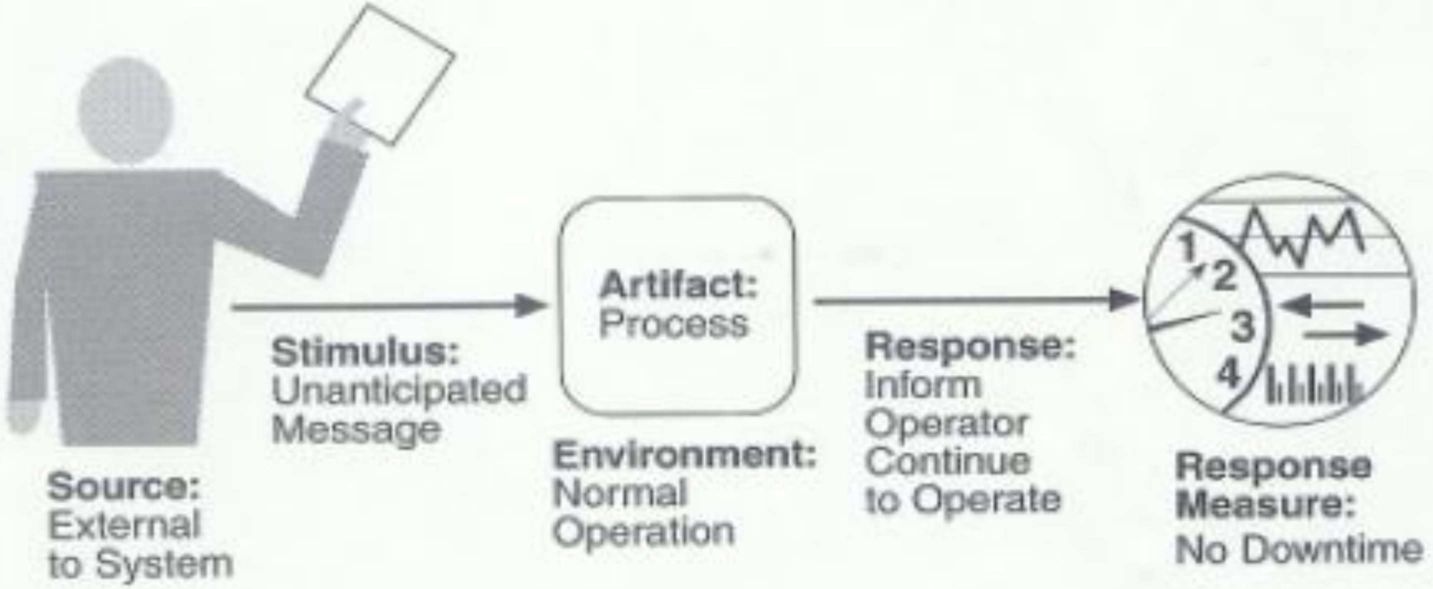
**Artifact:** Process, Storage, Processor, Communication

**Environment:** Normal, Degraded Operation

**Response:** Record, Notify, Disable, Continue (Normal/Degraded) Be Unavailable

**Response Measure:** Repair Time, Availability, Available/ Degraded Time Interval

# Sample Concrete Availability Scenario



**Source:** External to System

**Stimulus:** Unanticipated Message

**Artifact:** Process

**Environment:** Normal Operation

**Response:** Inform Operator Continue to Operate

**Response Measure:** No Downtime

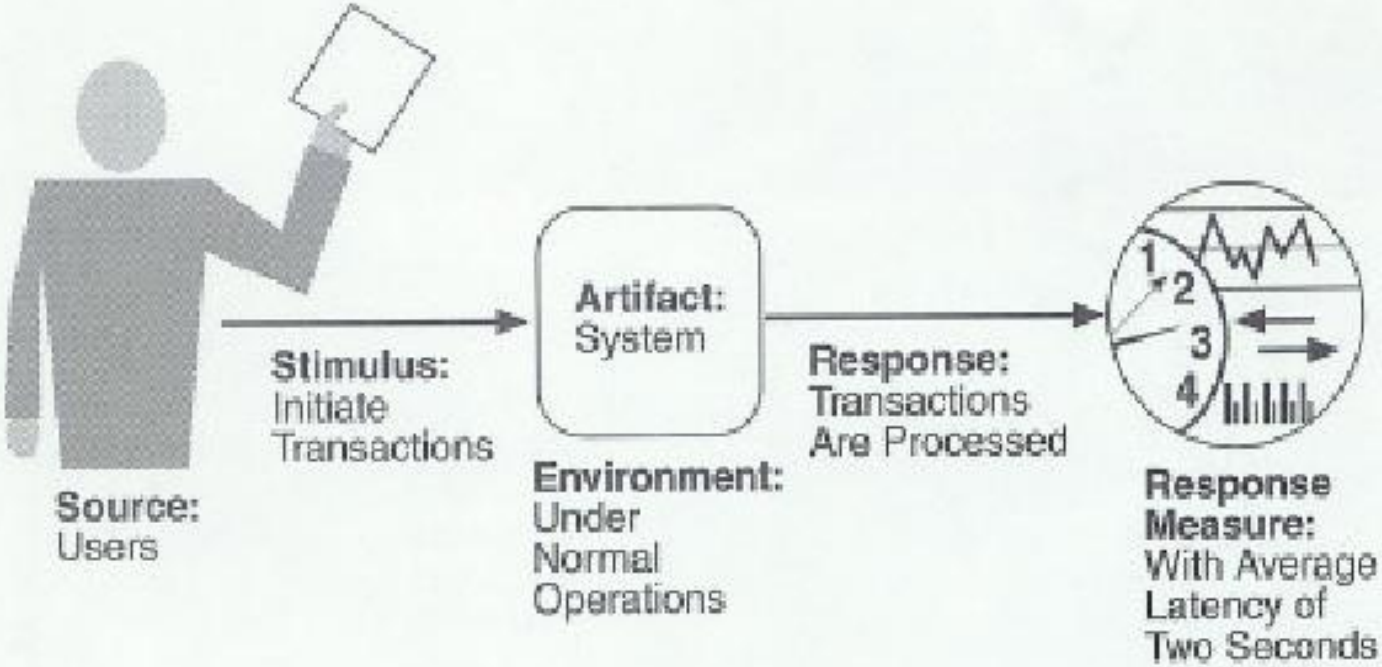
Sample availability scenario. An unanticipated external message is received by a process during normal operation. The process informs the operator of the receipt of the message and the system continues to operate with no downtime.

The tables serve as checklist to ensure that all possibilities have been considered.

# Table for Generation of General Performance Scenario

| Scenario Portion | Possible Values |
|---|---|
| Source | A number of sources both external and internal |
| Stimulus | Periodic events, sporadic events, stochastic events |
| Artifact | System, or possibly a component |
| Environment | Normal mode; overload mode |
| Response | Process stimuli; change level of service |
| RespMeasure | Latency, deadline, throughput, jitter, miss rate, data loss |

# Sample Performance Scenario

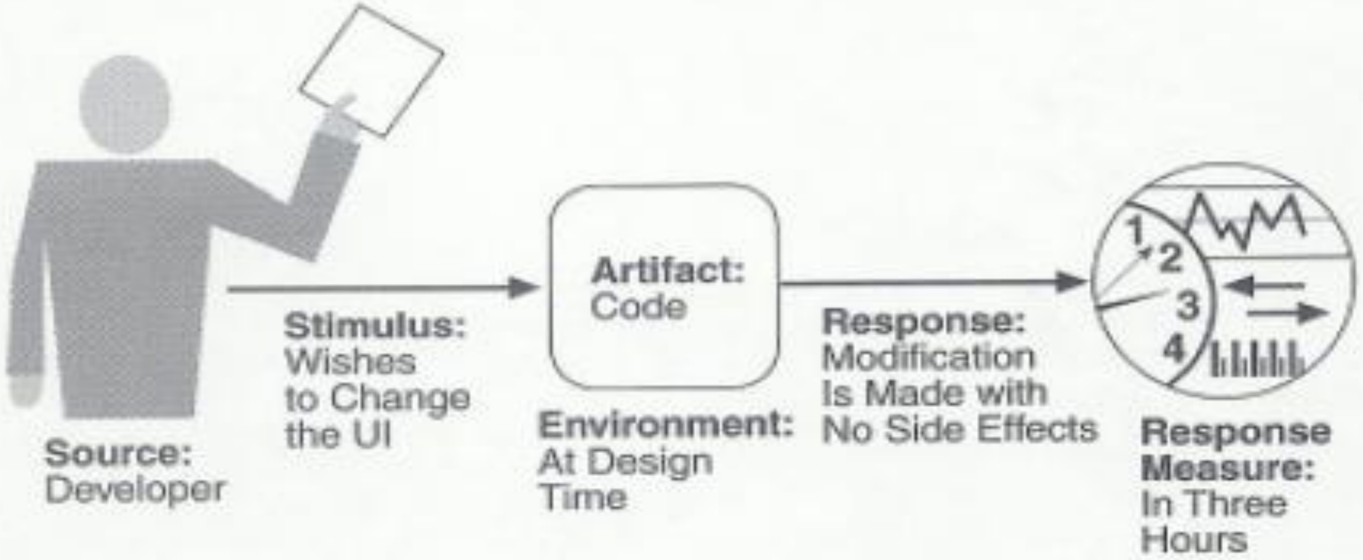


Sample performance scenario. Users initiate 1,000 transactions per minute stochastically under normal operations and these transactions are processed with an average latency of two seconds.

# Table for Generation of General Modifiability Scenario

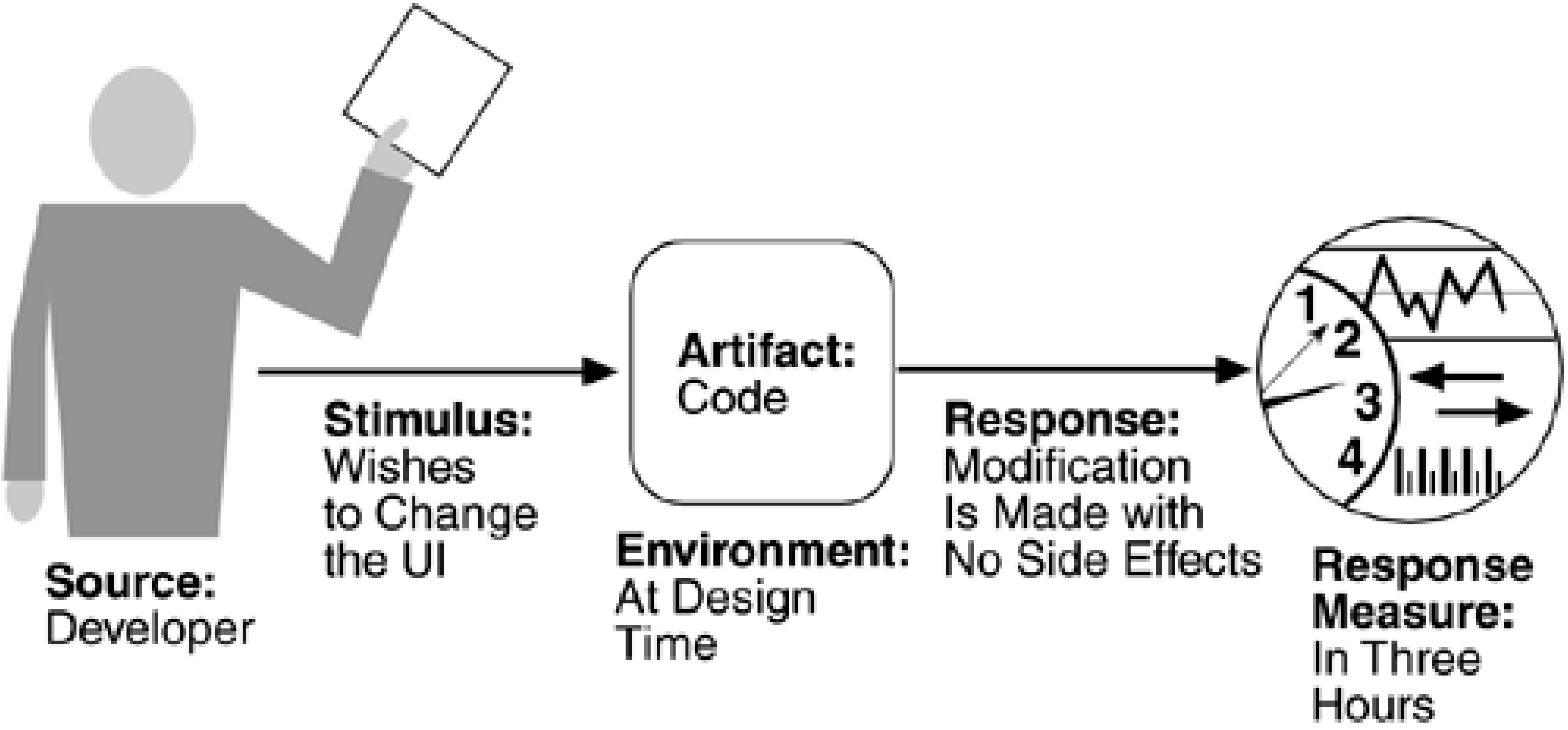
| Portion of Scenario | Possible Values |
|---|---|
| Source | End user, developer, system administrator |
| Stimulus | Wishes to add/delete/modify/vary functionality, quality attribute, capacity |
| Artifact | System user interface, platform, environment; system that inter-operates with target system |
| Environment | At runtime, compile time, build time, design time |
| Response | Locates places in architecture to be modified;<br>makes modification without affecting other functionality;<br>tests modification;<br>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



Sample modifiability scenario. A developer wishes to change the UI code at design time; modification is made with no side effects in three hours.

# Sample Modifiability Scenario



**Source:** Developer

**Stimulus:** Wishes to Change the UI

**Artifact:** Code

**Environment:** At Design Time

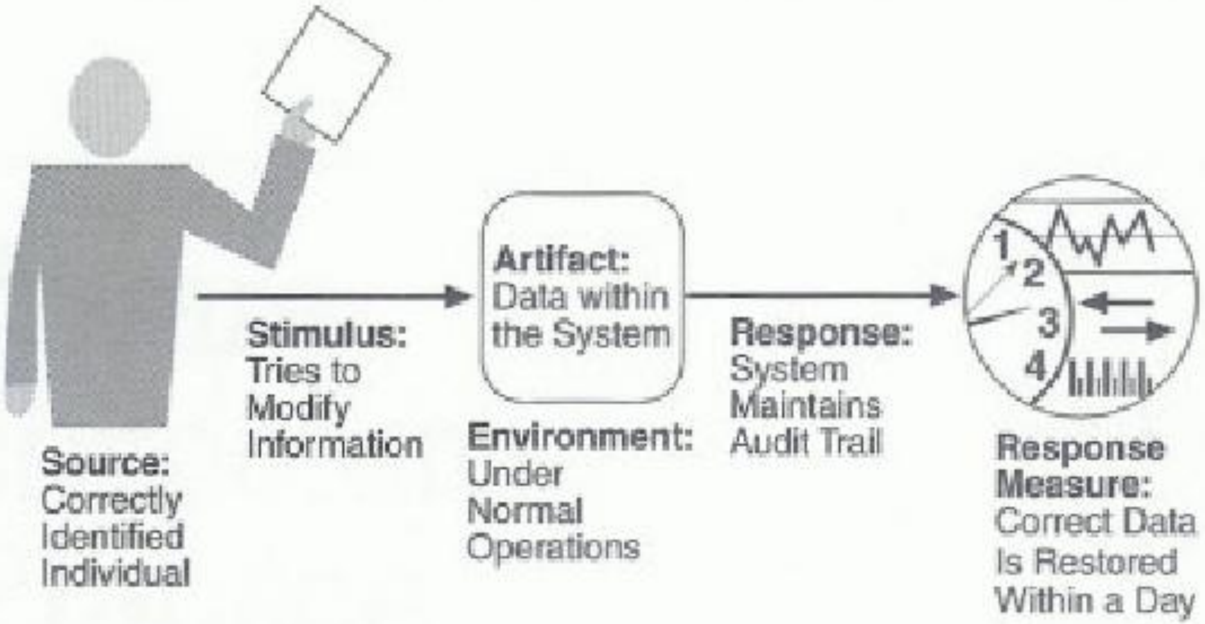
**Response:** Modification Is Made with No Side Effects

**Response Measure:** In Three Hours

# Table for Generation of General Security 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 |
| 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

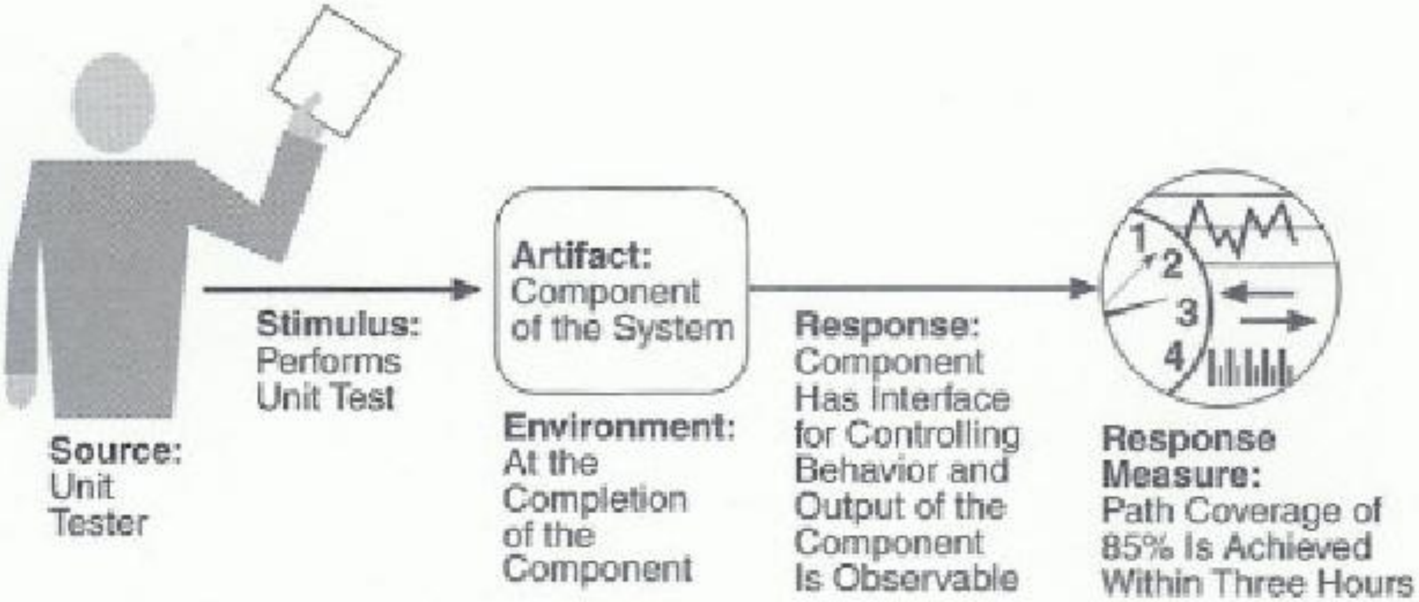

Source:
Correctly Identified Individual
Stimulus:
Tries to Modify Information
Artifact:
Data within the System
Environment:
Under Normal Operations
Response:
System Maintains Audit Trail
Response Measure:
Correct Data Is Restored Within a Day
1
2
3
4

Sample security scenario. A correctly identified individual tries to modify system data from an external site; system maintains an audit trail and the correct data is restored within one day.

# Table for Generation of General Testability Scenario

| Scenario Portion | Possible Values |
|---|---|
| Source | Unit developer, increment integrator, system verifier, client acceptance tester, system user |
| Stimulus | Analysis, architecture, design, class, subsystem integration, system delivered |
| Artifact | Piece of design, piece of code, complete system |
| Environment | At design time, at development time, at compile time, at deployment time |
| Response | Provide access to state data values, observes results, compares |
| RespMeasure | Coverage; prob of failure, time to perform tests, length of time to prepare test environment |

# Sample Testability Scenario



**Source:** Unit Tester

**Stimulus:** Performs Unit Test

**Artifact:** Component of the System

**Environment:** At the Completion of the Component

**Response:** Component Has Interface for Controlling Behavior and Output of the Component Is Observable

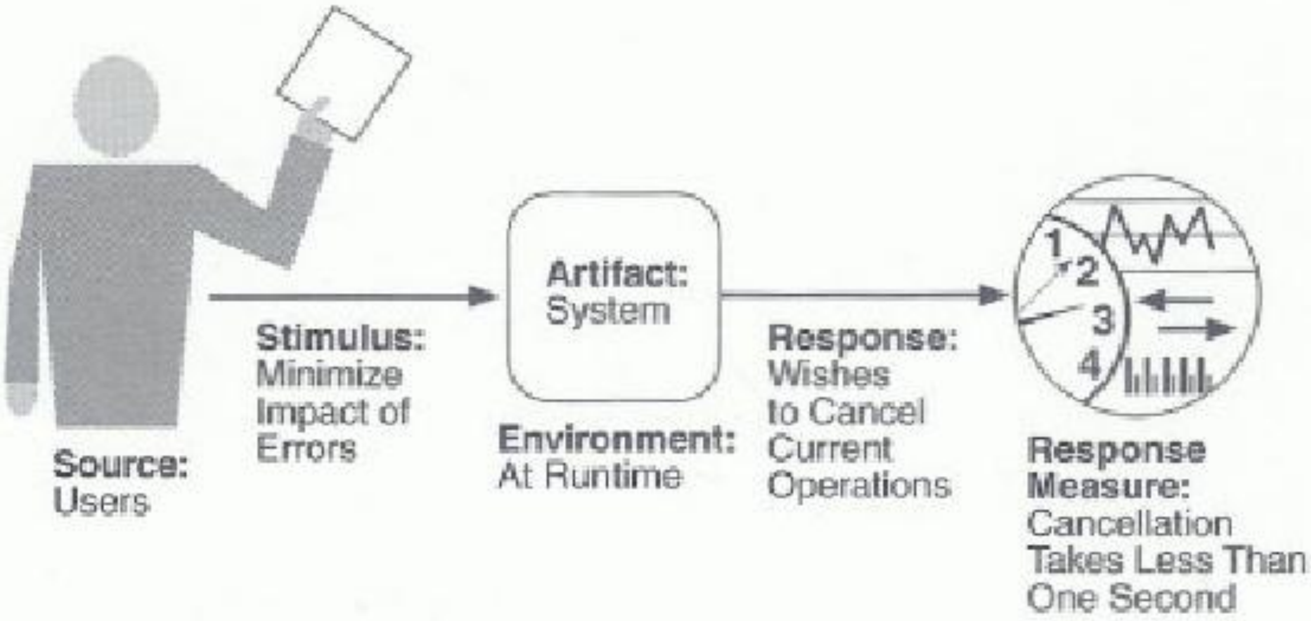
**Response Measure:** Path Coverage of 85% Is Achieved Within Three Hours

Sample testability scenario. A unit tester performs a unit test on a completed system component that provides an interface for controlling its behavior and observing its output; 85% path coverage is achieved within three hours.

# Table for Generation of General Usability 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 configuration time** |
| **Response** | **System provides one or more of the following responses:**<br>• **to support "learn system features" – help system is sensitive to context; interface is familiar to user; interface is usable in an unfamiliar context**<br>• **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**<br>• **to minimize "impact of errors" – undo, cancel, recover from system failure, recognize and correct user error, retrieve forgotten password, verify system resources**<br>• **to "adapt system" – customizability; internationalization**<br>• **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, ration of successful operations to total operations; amount of time/data lost** |

# Sample Usability Scenario



**Source:** Users

**Stimulus:** Minimize Impact of Errors

**Artifact:** System

**Environment:** At Runtime

**Response:** Wishes to Cancel Current Operations

**Response Measure:** Cancellation Takes Less Than One Second

Sample usability scenario. A user, wanting to minimize the impact of an error, wishes to cancel a system operation at runtime; cancellation takes place in less than one second.