FACULDADE ENGENHARIA
Departamento de Informática

Qualidade de Software (14450)

Quality Scenarios

(adapted from lecture notes of the “DIT 635 - Software Quality and Testing” unit, delivered by Professor Gregory Gay, at the Chalmers and the University of Gothenburg, 2022)

Today's Goals

- ✧ Examine quality attributes.
  - **Reliability:** Ability to minimize number of visible failures.
    - **Availability:** Ability to mask or repair failures so cumulative service outage is not excessive.
  - **Performance:** Ability to meet timing requirements.
    - **Scalability:** Ability to "grow" the system to process increasing number of requests.
  - **Security:** Ability to protect data and information from unauthorized access
- ✧ How to assess each using **scenarios**.

Scenarios

- A description of an interaction between an external entity and the system. Defines:
  - Event that triggers the scenario.
  - Interaction initiated by the external entity.
  - The response required of the system (defined in terms of quality attributes).
- Similar to use cases or user stories, but examines both quality and functionality.

Scenarios

Capture a range of requirements:

- A set of interactions with users to which a system must respond.
- Processing in response to timed events.
- Peak load situations that could occur.
- Regulator demands.
- Failure response.
- A change that a maintainer might make.
- Any situation that the design must handle.

Scenario Usage

- ✧ Provide input to architecture definition.
  - Help flesh out and find missing requirements.
- ✧ Evaluate system architecture.
  - Force description of execution paths through system
  - Find missing/incompatible interfaces.
- ✧ Communicate with stakeholders.
  - Concrete, easy to understand.
- ✧ Drive the testing process.
  - Help prioritize testing efforts.

Scenario Format

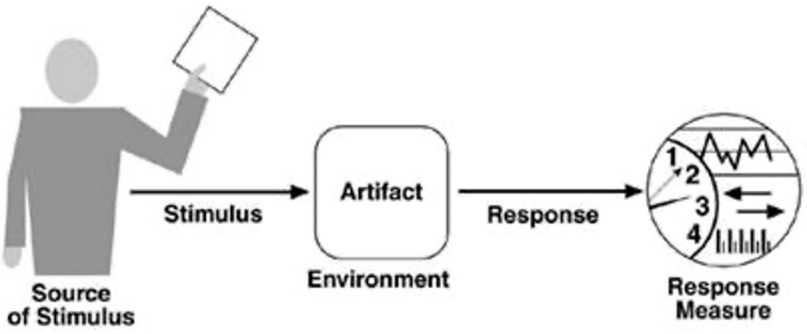
Overview:

Brief description of the scenario.

External Stimulus:

Input or environmental factors that initiate the scenario.

(e.g., user request, infrastructure changes or failures, security attacks)



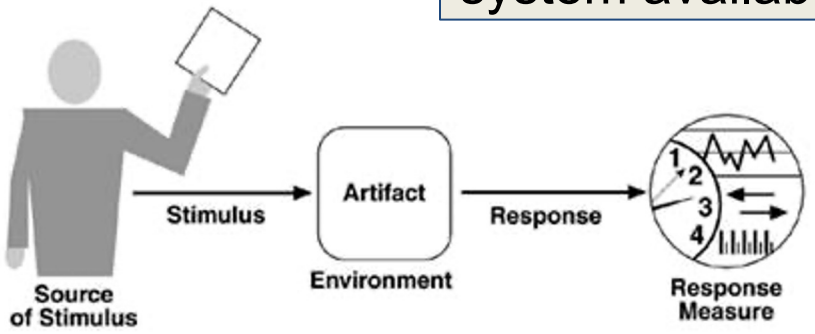
Scenario Format

System State:

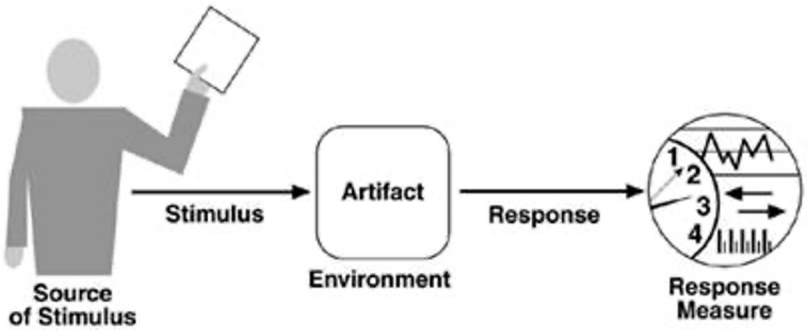
Aspects of internal state that affect quality (e,g., information stored in the system or database, current load)

System Environment:

Significant observations about the external environment (e.g., network connection, external system availability).



Scenario Format



Required System Response:

How does the system respond and meet the quality goal?

(e.g., how should it handle a defined increase in requests?)

Response Measure:

How we judge whether the system meets the quality goal.

(e.g., throughput, timing, availability)

Response Measures

✧ Most quality measurements are non-deterministic.

- **Time-based measures should be probabilistic.**
  - 95% of the time, the response should be N. **(common case)**
  - 99% of the time, the response should be M. **(worst case)**
- For real-time systems (i.e., embedded devices), time measurements should give **absolute bounds**.
  - Look at worst-case scenario.
- For other measures, gives an absolute threshold.

Example - Daily Data Update Increases in Size

- **Overview:** How the system's end-of-day processing behaves when regular data volumes are suddenly greatly exceeded.
- **System state:** The system has summary statistics in its database for data that has been processed, and the system's processing elements are lightly loaded at the current rate of system load.
- **System environment:** The deployment environment is working correctly, and data is arriving at a steady rate of 1,000 to 1,500 items per hour.
- **External Stimulus:** The data update rate on a particular day suddenly increases to 4,000 items per hour.
- **Required system behavior:** When the end-of-day processing starts, the system should process that day's data set for a period until the processing time exceeds a system-configurable limit. At that point, the system should stop processing the data set, discard work in process, leave the previous set of summary statistics in place, and log a diagnostic message (including cause and action taken) to the operational console monitoring system.

Example Availability Scenario

- **Overview:** One client-facing web server fails during transmission of page update.
- **System State:** System is working correctly under normal load.
- **Environment State:** Environment is operating normally.
- **External Stimulus:** Customer has generated a “add item to cart” post, which was routed to Server <X> in transaction pool. <X> crashes during response generation.

Example Availability Scenario

- **Required system behavior:** Response page may be corrupted on client. Load balancer no longer receives heartbeat message from server and removes it from the pool of available servers after 2s of missed messages, or on next request sent to the server. A page reload will be routed to alternate server by load balancer and page will be correctly displayed on client.
- **Response measure:** On client-side page refresh, client state and display contains state after last transaction. Time for re-routed refresh is equivalent to “standard” refresh (<1 second 95% of the time, <3 seconds 99% of the time).

What do we do with Scenarios?

- ✧ System Design
- ✧ Stakeholder Negotiation
- ✧ Exploratory Testing
  - Human experiments with app.
- ✧ Formal Test Cases
  - Assign specific input and check response.

“Good” Scenarios

- Credible
  - Describes a realistic scenario.
- Valuable
  - Can be directly used during architectural definition.
- Specific
  - Addresses a single, concrete situation.
- Precise
  - Intended user of scenario should be clear about the described situation and response.
- Comprehensible
  - Writing should be unambiguous and free of jargon.

Effective Scenario Use

- **Identify a focused scenario set**
  - Too many scenarios can be distracting.
  - Prioritize no more than 15-20.
- **Use distinct scenarios**
  - Avoid having multiple scenarios centered around near-identical events. They are redundant.
  - Consider demands placed on the system.
- **Use scenarios early**
  - Most impactful early in development to focus design activities on most important aspects of the system.

Reliability Scenarios

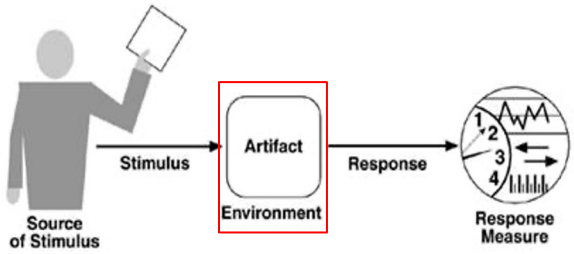
- **The ability to minimize the number of observed failures** (a.k.a. a statistical approximation of correctness).
- Scenarios revolve around one function (or a small set) accessed through an interface.
  - Give context on type of user if it impacts system execution or perceived reliability.

Reliability Metrics

- ✧ POFOD: **(failures/ requests over period)**
- ✧ ROCOF: **(failures / total time observed)**
- ✧ MTBF: **Average time between observed failures.**
- ✧ **Availability is a quality attribute of its own.**

Reliability Scenarios

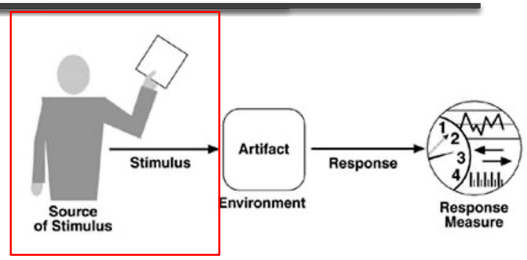
- **Overview:** Highlight the function(s) being used and the context used in. Explain the type of user, if relevant.



- **System state:** Data stored in system or past events may impact reliability (multiple failures may leave system vulnerable)
- **Environment state:** Available resources may impact reliability (resources, networking).

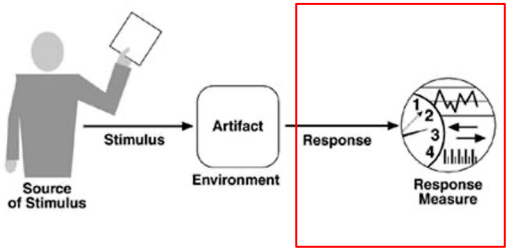
Reliability Scenarios

✧ External Stimulus:



- User or system performs one or more functions.
- State specific action or sequence of actions performed.
- If relevant, explain the type of user and reason they would perceive reliability differently.

Reliability Scenarios



✧ Required Response:

- The functional response of the system.
- Generally, the normal operation of that function.

✧ Reliability Measure:

- ROCOF, POFOD, MTBF
- State probabilistically (generally given as an average)
  - “The average ROCOF will be at most 97 out of 1000 requests.”
  - MTBF already an average.

Example Reliability Scenario

- **Overview:** A user adds an item to the shopping cart.
- **System State:** The system is operating normally, under normal load (500 concurrent users). The shopping cart is empty.
- **Environment State:** The environment is operating normally, with standard connectivity.
- **External Stimulus:** A user selects the “add to cart button” with the quantity set to “1”.
- **Required Response:** The item is successfully added to the shopping cart. The number of items displayed on the cart icon is incremented by one.
- **Response Measure:** Average ROCOF for this function is 15/1000 requests.

Example Reliability Scenario 2

- **Overview:** A “power user” requests summary statistics for data in the spreadsheet.
- **System/Environment State:** The system and environment are operating normally.
- **External Stimulus:** A “power user” clicks the data summary button.
  - A power user is defined as one who manages a large volume of data (> 20000 rows), and accesses this function at least once per hour.

Example Reliability Scenario 2

✧ Required Response:

- The summary statistics are calculated and displayed to the screen. The statistics are also written to a CSV file (appended to the file if it already exists).

✧ Response Measure:

- The MTBF for this function must be at least 8 hours.
  - Power uses expect long, uninterrupted sessions, and expect accurate results on a regular basis.

Availability Scenarios

- ✧ Ability of the system to mask or repair failures such that the outage period does not exceed a required value over a time period.
- ✧ **Measure how the system responds to failure.**
  - What does the system do to return to normal?
  - How long does it take?
- ✧ **Stimuli should always be a failure.**

Availability Scenarios

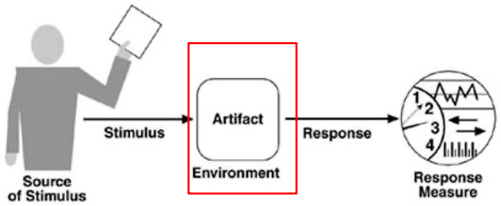
- Response measures should always include a measure of availability:
  - availability threshold
    - (“Availability must be at least 0.9999”)
  - time to detect or repair fault
    - (“95% of the time, the failure is detected within 5ms”)
  - time system in degraded mode
    - (“95% of the time, must be back online within 10 minutes”)

Availability Scenarios

- Scenarios must distinguish physical failures in the system and the software's perception of the failure.
  - Do not assume software is omniscient.
- Scenarios tend to deal with:
  - Failure of internal component or external system.
  - Reconfiguration of physical system.
  - Maintenance or reconfiguration.

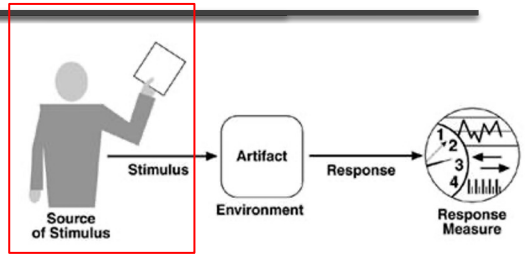
Availability Scenarios

- **Overview:** Be clear about what needs to be available.



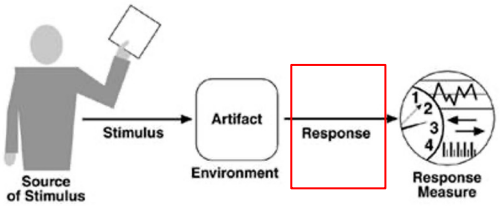
- **System/environment state:** The state of the system when the fault or failure occurs may also affect the desired system response.
  - If the system has already failed and is not in normal mode, it may be desirable to shut it down.
  - If this is the first failure, degradation of response time or functions may be preferred.

Availability Scenarios



- **External Stimulus:** Differentiate internal/external failure - desired response may differ.
- Stimuli is:
  - An ***omission*** (a component fails to respond to an input),
  - A ***crash*** (component repeatedly suffers omission faults)
  - ***timing*** (component responds but response is early/late)
  - ***response*** (component responds with incorrect value).

Availability Scenarios

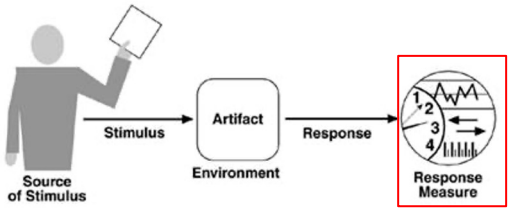


- **Required Response:**
- Many possible reactions.
- Failure must be detected and isolated before recovery.
- Actions include:
  - Logging the failure
  - Notifying users or systems
  - Taking actions to limit the damage
  - Switching to a degraded mode
  - Shutting down external systems
  - Becoming unavailable during repair.

Availability Scenarios

✧ Response measure:

- Can specify an availability percentage
- Can specify a time:
  - to detect the fault
  - to repair the fault
  - times where system must be available
  - duration system must be available



Example Availability Scenario

- **Overview:** How the system handles additional beer taps being added to the dispensing system.
- **System/environment state:** System and environment operating normally.
- **External Stimulus:** A user powers up a new Kegboard on the network with six additional taps.
- **Required system behavior:** The kegboards send init messages to the central Kegbot server. The server interrogates the kegboards and adds the additional taps to the inventory of taps. The system continues to service the existing taps without interruption.
- **Response measure:** There is no interruption of service to existing taps. Within 1 second, the new kegboard is added to the administrative interface on the KegBot web server for administration configuration.

Example Availability Scenario 2

- **Overview:** How the server-side components handle non-response from external payment system.
- **System state:** System is operating under heavy load (>10000 concurrent users).
- **Environment state:** External payment processing system has exceeded load limits.
- **External Stimulus:** Multiple (5+) queries to the external system have gone without response without any successful responses between.

Availability Scenario 2

- **Required system behavior:** The system will cease to allow any further orders until the external system responds to a heartbeat message. An error page will be displayed to all clients to prevent further order attempts. Once the external system has returned for sufficient time (100 responses over at least a five minute period), normal operations will be resumed.
- **Response measure:** Following detection of the failure, all order attempts will be stopped within one minute (90% of the time) and two minutes (99% of the time). Once the failure is resolved, normal operations will be resumed within one minute (following the five minutes of successful responses) 90% of the time and three minutes 99% of the time.

Performance Scenarios

- ✧ Measure system performance (not user).
- ✧ Begins with an event arriving at the system.
  - Responding requires resources to be consumed.
- ✧ Arrival pattern for events can be:
  - Periodic (at regular time intervals)
  - Stochastic (events arrive according to a distribution)
  - Sporadic (unknown timing, but known properties)
    - “No more than 600 per minute”
    - “At least 200 ms between arrival of two events”

Performance Measurements

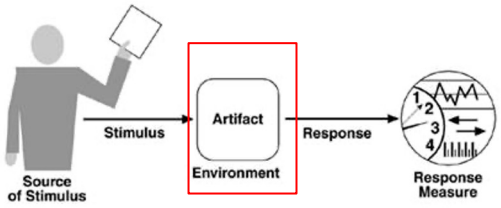
- **Latency:** The time between the arrival of the stimulus and the system's response to it.
- **Response Jitter:** The allowable variation in latency.
- **Throughput:** Number of transactions system can process in a unit of time.
- **Deadlines in processing:** Points where processing must have reached a particular stage.
- **Number of events not processed** because the system was too busy to respond.

Specifying Response Time

- ✧ Response time targets require a defined load.
  - One transaction in 3s is easy if that is the only request.
    - Can you still hit 3s if there are 500 transactions per second?
  - Must specify a clearly-defined response time goal.
    - Define when a transaction starts and ends.
- ✧ Not all requests take the same amount of time, even with constant load.

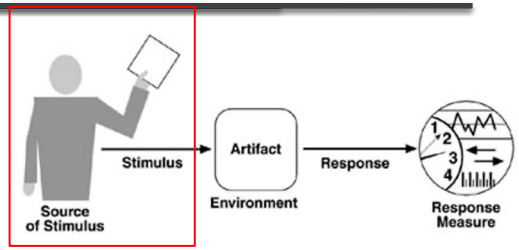
Performance Scenarios

- **Overview:** Description of the scenario.



- **System State:** System can be in various levels of load (normal, emergency, peak load, or overload).
- **Environment State:** Be clear on conditions that can impact performance.
  - Limited resources (disc, memory, CPU)
  - Networking conditions

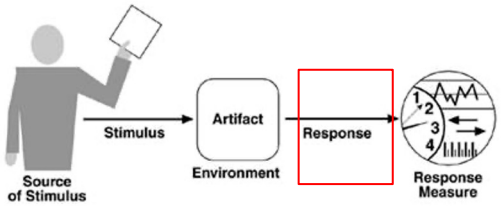
Performance Scenarios



- **External Stimulus:** Stimuli arrive from external or internal sources.
  - The stimuli are event arrivals.
  - Arrival pattern can be periodic, stochastic, or sporadic, characterized by numeric parameters.
  - Be clear on number, duration, concurrency of stimuli.

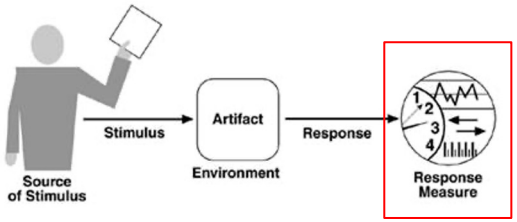
Performance Scenarios

- **Required system behavior:**
  - System must process arriving events
  - May cause change in system
    - (e.g., shift from normal to overload mode).
  - May cause change in environment
    - (e.g., reduction of available memory)



Performance Scenarios

✧ Response measure:



- Time to process arriving events (**latency or a deadline**)
- Variation in latency time (**jitter**)
- Number of events that can be processed within a time interval (**throughput**)
- Characterization of the events that cannot be processed (**miss rate**).

Example Performance Scenario

- **Overview:** Check system responsiveness for adding items to shopping cart under normal conditions.
- **System State:** Normal load (less than 20 customer requests per second).
- **Environment State:** System is communicating over good internet connection to client.
- **External Stimulus:** Customer adds product to cart.

Example Performance Scenario

- **Required system behavior:** Web page refreshes. Icon on right side of web page displays last item added to cart. If item is out of stock, cart icon has exclamation point overlay on top of cart icon.
- **Response measure:** In 95% of requests, web page is loaded and displayed to user within 1 second. In 99.9% of requests, web page is loaded and displayed to user within 5 seconds.

Example Performance Scenario 2

- **Overview:** Ensure that credit card processing can still meet throughput targets when many users are competing for resources.
- **System state:** System is operating under heavy load (10000 concurrent users are logged in).
- **Environment state:** Environment is operating normally. Load balancer distributes user requests approximately evenly between 100 servers.
- **External stimulus:** A large number of credit card processing requests come within a short window (8500 requests within a one minute window).

Example Performance Scenario 2

- **Required system behavior:** Each server maintains a queue of requests and processes requests as resources become available. The load balancer distributes requests to servers, favoring servers with shorter queues. All requests are completed successfully.
- **Response measure:** All 8500 requests are completed within two minutes, 85% of the time. All requests are completed within three minutes 99% of the time.

Scalability Scenarios

- ✧ **The ability to efficiently use available resources.**
- ✧ Scenarios assessing scalability directly deal with impact of **adding or removing resources**.
- ✧ Performance measures to reflect:
  - Changes to performance.
  - Changes to availability.
  - Load assigned to existing and new resources.

Example Scalability Scenario

- **Overview:** Addition of new hardware improves credit card transaction speed.
- **System/environment state:** Before addition of new hardware, 95% of credit card transactions were completed within 10 seconds, 99.9% within 15s. Additional server has doubled threads available for processing requests. System is under normal load, with normal connectivity. Environment is operating normally.
- **External Stimulus:** Customer completes a purchase.

Example Scalability Scenario

- **Required system behavior:** Order confirmation is displayed, with a list of items purchased, expected arrival date, and total cost of items.
- **Response measure:** In 95% of requests, web page is loaded and displayed to user within 5 second. In 99.9% of requests, web page is loaded and displayed to user within 7.5 seconds.

Example Scalability Scenario 2

- **Overview:** Addition of additional VMs improves availability of account authorization service.
- **System/environment state:** Before addition of 10000 VMs to the available pool, the average availability of the authorization service was 97.35% per week.
  - The VM pool has now been increased by 150%.
  - System is under normal load.
  - Environment is operating normally.
- **External stimulus:** A user submits their username and password for authentication. The password is correct.

Example Scalability Scenario 2

✧ Required system behavior:

- Authentication completes successfully.
- A session is established, and the user's customized homepage is displayed on the client browser.

✧ Response measure:

- The average availability of the authorization service is increased to at least 99% per week.

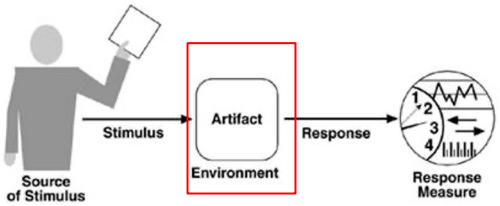
Security Scenarios

- **Ability to protect data from unauthorized access while still providing service to authorized users.**
  - Stimuli are attacks from external systems/users or demonstrations of policies (log-in, authorization).
- Responses:
  - Auditing, logging, reporting, analyzing.
- Measures:
  - time to detect/stop attack; time to identify attacker; impact of breach

Assessing Security

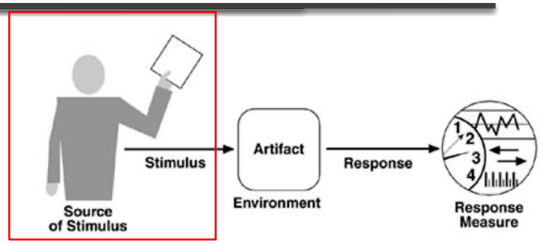
- ✧ No universal metrics for measuring “security”.
  - Present specific attack types and specify response.
- ✧ Response assessed by appropriate metrics.
  - Time to identify attacker.
  - Amount of data protected.
  - Time to stop attack.
  - Number of accounts compromised.
  - Availability during attack.

Security Scenarios



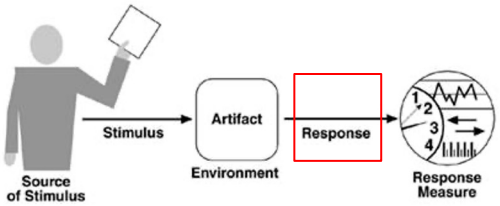
- **Overview:** Description of the scenario.
  - Be clear about goal (examine attack response, demonstrate policy)
- **System/environment state:**
  - The attack can come when the system is online or offline
  - Connected to or disconnected from a network
  - Behind a firewall or open to a network
  - Fully operational, partially operational, or not operational.

Security Scenarios



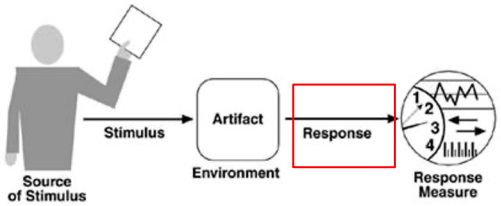
- **External Stimulus:**
  - Source may be either a human or external system.
    - May have been previously identified or may be unknown.
    - Attacker may be from outside or inside organization.
  - The stimulus is an **attack**
    - Unauthorized attempt to display data
    - Change or delete data
    - Access services
    - Change system's behavior, or reduce availability.

Security Scenarios



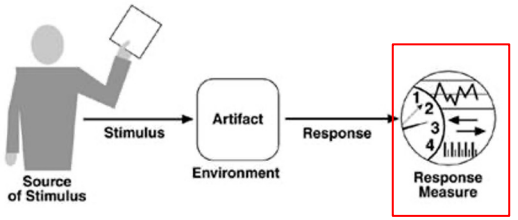
- **Required response:**
  - Should ensure that:
    - Data/services protected from unauthorized access.
    - Data/services not manipulated without authorization.
    - Parties identified and cannot repudiate involvement.
    - Data/resources/services available for legitimate use.

Security Scenarios



- **Required response:**
  - The system should track activities by:
    - Recording attempts to access or modify data, resources, or services.
    - Notifying appropriate entities (people or systems) when attack is occurring.

Security Scenarios



✧ Response Measure:

- How much of system is compromised when particular component or data compromised.
- How much time passed before attack was detected.
- How many attacks were resisted.
- How long it took to recover from a successful attack.
- How much data were vulnerable to a particular attack.
- How long it takes to identify attacker.

Example Security Scenario

- **Overview:** A disgruntled employee at a remote location attempts to change their pay rate.
- **System state:** System is operating normally. 100 active users are logged in.
- **Environment state:** Environment is operating normally.
- **External stimulus:** An employee has discovered the location of a configuration file storing all employee pay rates. They log in (using their credentials) and use a stolen passkey to open the locked file. They modify the file with a new rate and save changes.

Example Security Scenario

- **Required system behavior:** The system maintains an audit trail. The user is able to modify the file, as they have the passkey. However, the log records the date, time, identify of user, and modification made. System administrators are informed of the modification.
- **Response measure:** The correct data is restored within a day and the source of tampering has been identified and reported.

Example Security Scenario 2

- **Overview:** A user attempts to authenticate with the beer dispensing system (to purchase beer) but the authentication fails due to unrecognized authorization token.
- **System state:** System is operating normally, without problems.
- **Environment state:** There is a valve installed on the tap. There is a flow meter installed on the tap. There is a buzzer installed on the Kegboard. Authentication hardware (RFID or one-wire) is installed on the Kegboard. There is no pour in progress.
- **External stimulus:** A user presents an authorization token to the authentication sensor on the Kegboard.

Example Security Scenario 2

- **Required system behavior:** The authorization token is unrecognized, and the valve is not opened. An audible sound is played from the buzzer, indicating authentication failure.
- **Response measure:** Authorization fails within 2 seconds 95% of the time (within 5 seconds 99% of the time). Buzzer sounds within 5 seconds 95% of the time (7 seconds, 99%). No beer is dispensed.

Key Points

- Defining and applying scenarios ensures that desired quality attributes are shown.
- Scenarios define how the system responds to factors that affect quality properties.
- Should include the initial system state and environment state, external stimulus, required system response, and how to assess response.


HOW
WHY
WHERE
WHEN
WHERE
WHY
WHERE
WHO
WHEN
WHAT
WHAT
WHY
WHAT
WHEN
WHO
WHEN
WHERE
WHY
HOW