Quality Attributes

Thilina Halloluwa
tch@ucsc.cmb.ac.lk

Session objectives

- Quality and functionality
- Quality and architecture
- Quality attribute scenario
- Quality Attribute Scenarios in Practice

Functional requirements

- What is functionality ?
  - The ability of the system to do what it was intended.
  - Need a coordinated effort.

A **house** is a home, building, or structure that functions as a habitat for humans . A shelter against heat cold, rain, invaders, enemies, etc

functionality is independent of the structure.

Functional requirements Contd..

- Functional requirements:
  - Describe what a system should do.
  - Mostly come from the customer.
  - Can be described by a use case model.

Non-functional requirements

- Examples?
- Requires tradeoffs
  - Weight vs Speed (The heavier a car , slower it accelerate)
- More architecture-dependent than functional requirements.
  - Often determined by the architect and stakeholders within the organization.
- Can be described in terms of standard quality attributes.

Quality and functionality

- Functionality
  - the ability of the system to do the work for which it was intended.
- Quality attributes
  - Business considerations determine qualities that must be accommodated in a system's architecture.
  - The mapping of a system's functionality onto software structures determines the architecture's support for qualities.
  - Examples?
- **Quality and functionality (hw)**
  - **Orthogonal or complement?**

Quality attributes

System qualities	• availability, modifiability, performance, security, testability, usability, others.
Business qualities	• time to market, cost and benefit, product lifetime, target market, rollout schedule, integration, others.
Architectural qualities	• conceptual integrity, correctness and completeness.

Activity....

- Usability
  - Radio button or Check box
  - Screen layout
  - Font
  - Have ability to cancel or undo operations
  - Ability to reuse data
- Modifiability
  - Module division
  - Coding technique used
- Performance
  - Component communication
  - Shared resource allocation
  - Choice of algorithm?
  - Algorithm is implemented?

Quality and architecture

- Usability (How easy it is to use/learn)
  - Radio button or Check box
  - Screen layout
  - Font
  - Have ability to cancel or undo operations
  - Ability to reuse data

Quality and architecture

- Modifiability
  - Module division
  - Coding technique used
- Performance
  - Component communication
  - Shared resource allocation
  - Choice of algorithm?
  - Algorithm is implemented?

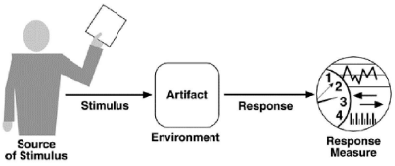
- Architecture alone can't achieve qualities it provides the foundation.

Current issues on definitions

- Non operational attributes
  - It might be modifiable for some changes , but for another set, it might not.
- Overlapping attribute concerns
  - A system might fail because of availability , security or usability?
- Attributes link to their own vocabulary.
  - Security – attacks
  - Performance- events
  - Availability- failures
  - Usability- user inputs

Quality attribute scenario

- Why?
- Parts
  1. Stimulus source
  2. Stimulus
  3. Environment
  4. Artifact
  5. Response
  6. Response measure
- General and concrete scenarios



A collection of concrete scenarios can be used as the quality attribute requirements for a system.

Quality attribute scenario

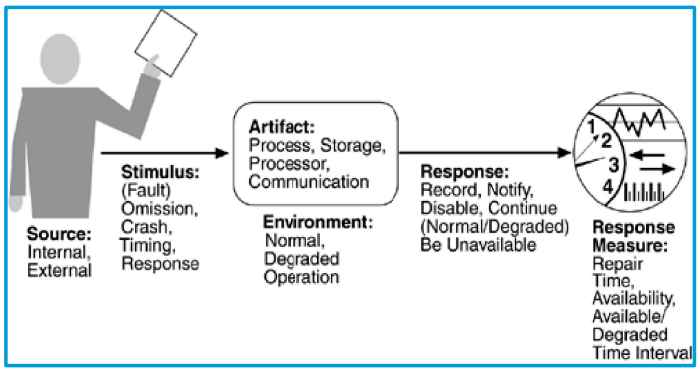
- Source — Who makes the change (an entity)
- Stimulus — The change occurred
- Artifact — What is affected
- Environment — A certain condition
- Response — What to do when stimulus arrive
- Response measure — Measurement

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 continues to operate with no downtime."

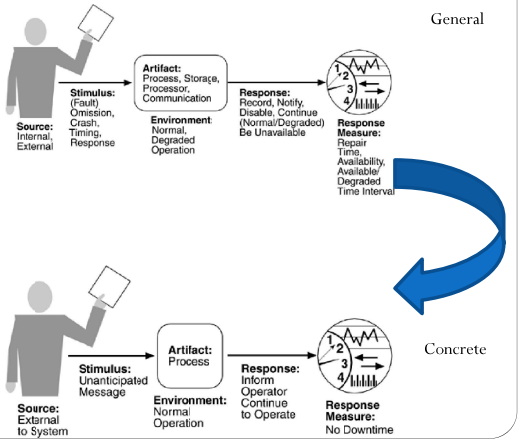
1. Stimulus source 2. Stimulus 3. Environment 4. Artifact
5. Response 6. Response measure

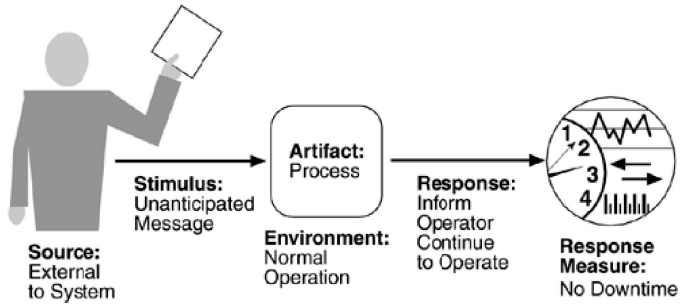
Availability scenario



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 continues to operate with no downtime."





Scenario parts

1. Stimulus source
   - Trusted and untrusted sources
2. Stimulus
3. Environment
   - Overloaded or idle
4. Artifact
5. Response
   - Value should be explicitly defined
6. Response measure
   - It is the "measure" that makes a scenario operational.

Quality Attribute Scenarios in Practice

- Examples we are going to cover
  1. AVAILABILITY
  2. MODIFIABILITY
  3. PERFORMANCE
  4. SECURITY
  5. **TESTABILITY**
  6. **USABILITY**

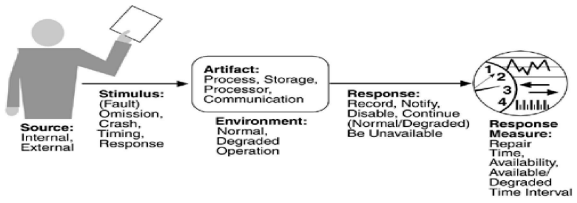
Quality attribute scenarios vs. use case scenarios

AVAILABILITY

Concerned with system failure and it's consequences

- Faults and failures
  - Using the wrong algorithm for computation
  - Miscalculation / incorrect output
- Concerns on failure
  - Frequency
  - Results
  - Non-operative time
  - Prevention
  - Notifications
- $availability = \frac{mean\ time\ to\ failure}{mean\ time\ to\ failure + mean\ time\ to\ repair}$

General availability scenario



Portion of Scenario	Possible Values
Source	Internal to the system; external to the system
Stimulus	Fault: omission, crash, timing, response
Artifact	Systèmes processors, communication charnels, Storage, or process
Environment	Normal operation; Degraded mode (i.e., fewer features, a fall back solution)

General availability scenario

Portion of Scenario	Possible Values
Response	System should detect event and do one or more of the following: ❖record it ❖notify appropriate parties, including the user and other systems ❖disable sources of events that cause fault or failure according to defined rules ❖be unavailable for a prespecified interval, where interval depends on criticality of system ❖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

The concreate scenario (Availability)

Portion of Scenario	Possible Values
Source	External to the system
Stimulus	Unanticipated Message
Artifact	Process
Environment	Normal operation
Response	Notify appropriate parties (Inform the operator and continue)
Response Measure	No downtime

Modifiability

All about changes and have 3 concerns,

What
Who
When

- What can change? **artifact**
  - Functions
  - Platform
  - Environment
  - Protocols
  - Qualities

Modifiability

- When are changes made? **Environment**
- Who makes the changes? **Source**

- During the ,
  - Implementation
  - Compilation
  - Build
  - Configuration
  - Execution

- A Developer
- End User
- System Administrator

General Modifiability scenario

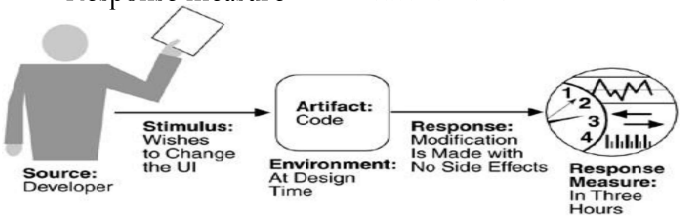
Portion of Scenario	Possible Values
Source	End user, developer, system administrator
Stimulus	The change to be made :Wishes to add/delete/modify/vary functionality, quality attribute, capacity
Artifact	System user interface, platform, environment; system that interoperates with target system
Environment	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

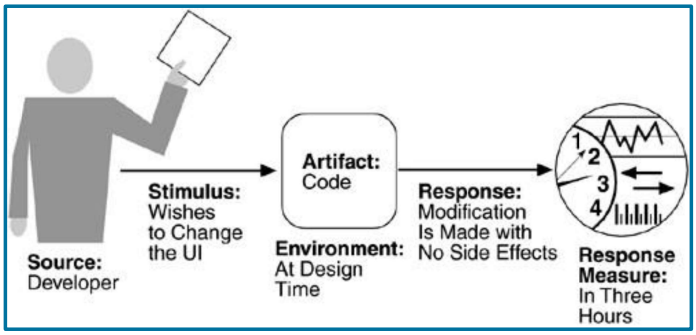
Q1

A developer wishes to change the user interface. This change will be made to the code at design time, it will take less than three hours to make and test the change, and no side-effect changes will occur in the behavior.

Question 1

- Source — Who makes the change
- Stimulus — The change to be
- Artifact — What is to be changed
- Environment — When the change can be made
- Response — How to make it
- Response measure — Measurement





Performance

- The performance quality is concerned with response times and similar measures for various events.
- Checks how long it takes the system to respond for an event.

General Performance scenario

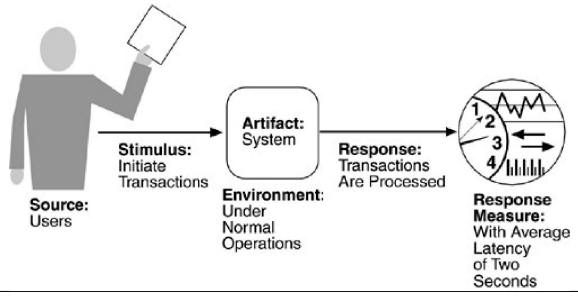
Portion of Scenario	Possible Values
Source	One of a number of independent sources, possibly from within system
Stimulus	Periodic events arrive; sporadic events arrive; stochastic events arrive
Artifact	System
Environment	Normal mode; overload mode
Response	Processes stimuli; changes level of service
Response Measure	Latency, deadline, throughput, jitter, miss rate, data loss

Performance response measures

- Latency – arrival and response
- Deadlines in processing
- Throughput
- Jitter
  - delay between the invocation (or arrival) of a task, and its release (when it actually starts to execute).
  - Operating systems: Say that the kernel preempts user processes at regular intervals (10 ms), and that you want a task to run every 23 ms. The OS scheduler will not be able to satisfy your request exactly. Theoretically, your task should run at times 0, 23, 46… In practice, it will run at 0, 30, 50…The variation of the actual release times is jitter (0, 7, 4, … in this example).
- Miss rate
- Data loss

Concrete 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."



Security

- Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legitimate users
- An attempt to breach security is called an attack and have many forms
  - unauthorized attempt to access data
  - modify data
  - intended to deny services to legitimate users.

Characterization

- Nonrepudiation
  - Transaction cannot be denied by any of the parties
- Confidentiality
  - Data or services are protected from unauthorized access.
- Integrity
  - Data or services are being delivered as intended.
- Assurance or authenticity
  - The parties to a transaction are who they purport to be
- Availability (no denial of service)
  - The system will be available for legitimate use
- Auditing
  - The system tracks activities

Security scenario

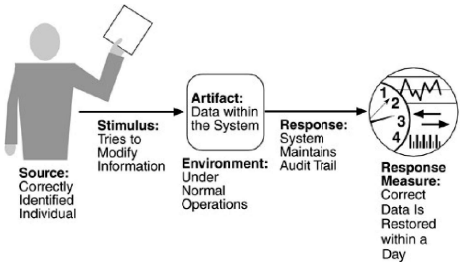
Portion of Scenario	Possible Values
Source	Human or system that is correctly identified, identified incorrectly, of unknown identity who is internal/external , authorized/ not authorized with access to limited resources
Stimulus	**Unauthorized** person Tries to display data, change/delete data, access system services, reduce availability to system services (an attack or an attempt to beak security)
Artifact	System services; data within system
Environment	Either online or offline, connected or disconnected from a network , firewalled or open network

General Security scenario

Portion of Scenario	Possible Values
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

Security concrete 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



Usability

- Can be broken down to..
  - How easy it is to learn the features of the system
  - How efficiently the user can use the system
  - How well the system handles user errors
  - How well the system adapts to user needs
  - To what degree the system gives the user confidence in the correctness of its actions.

General usability scenario

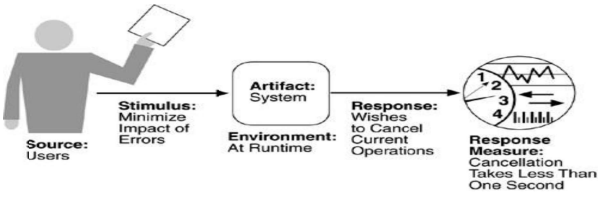
Portion of Scenario	Possible Values
Source	End user is always the source
Stimulus	Wants to learn system features; use system efficiently; minimize impact of errors; adapt system; feel comfortable
Artifact	System
Environment	At runtime or configure time

General usability scenario

Portion of Scenario	Possible Values
Response	System provides one or more of: ❖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

Concrete 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



Testability

Testability refers to the ease with which software can be made to demonstrate its faults through testing

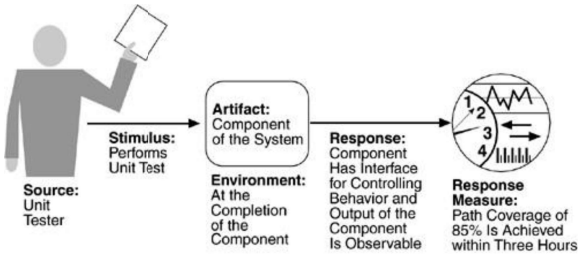
- Detecting failure modes
- 40% of the cost of a large project is spent on testing
- If a system to be properly tested ,need to control the internal state of and inputs to each unit, then observe the corresponding output of that unit.

General Testability scenario

Portion of Scenario	Possible Values
Source	Unit developer; Increment integrator; System verifier; Client acceptance tester; System user
Stimulus	**A milestone in the development process is met** : 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
Response	Provides access to state values; provides computed values; prepares test environment
Response Measure	Percentage of the 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

Testability Concrete Scenario

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



Other scenarios

- Additional quality attributes:
  - Scalability
  - Portability
  - And so on…
- The process is the same: we create a general scenario and then create a set of specific scenarios for the effort at hand.
- Most of the other commonly discussed quality attributes can be expressed in terms of the previous ones. For example, portability is a specialized form of modifiability, and data integrity is a specialized form of security.

Business quality attributes

- Business quality requirements analysis relates the business quality scenarios to the architecture.
  - Time to market
    - If there is competitive pressure or a short window of opportunity.
  - Cost and benefit: The development effort will naturally have a budget that must not be exceeded
    - In-house architectural expertise is cheaper than outside expertise.
  - Projected lifetime of the system
    - If the system is intended to have a long lifetime, modifiability, scalability, and portability become important.. But building in the additional infrastructure will usually compromise time to market.

Business quality attributes

- Rollout schedule:
  - If functionality is planned to increase over time, the architecture needs to be customizable and flexible.
- Integration with legacy systems:
  - If the new system has to *integrate* with existing systems, care must be taken to define appropriate integration mechanisms

Architectural quality attributes

- Architectural quality attributes are also similar to system quality attributes, but concerned with aspects of the architecture itself.
  - Conceptual integrity : The architecture should do similar things in similar ways.
  - Correctness and completeness is concerned with checking the architecture for errors and omissions.
  - 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

Case Study : Garage Door Opener

- A real-time system responsible for raising and lowering the garage door, via
  - Switch button
  - Remote control
  - Home information system
- It is possible to diagnose problems of opener from the home information system (HIS)

Case Study : Garage Door Opener

- What can be the main Quality attributes…??

- Performance
- Availability
- Modifiability
- Usability

Write a Performance scenario..

- Reacting to obstacles [Performance]

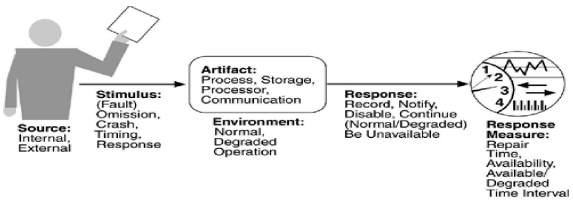
If an obstacle (person/object) is detected by garage door during descent, it must halt or re-open within 0.1 second and report to HIS and user interface.

Write a Performance senario..

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

Write a Availability scenario...

- Door commands [Availability]
- Remote control; HIS; button.
- Open, close, halt, diagnosis
- Detect an stop execution



Write a Modifiability scenario…

- Processors can change due to either obsolescence or changes in the marketplace.

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