Name : Unit - 2 Roll No. :
Subject : Class :
Batch : Division :

* Undesstaing Quality Attributes

In the Architecture Business Cycle, business consideuations determine qualities that must be accommodated in a system's architecture.

These qualities are over and above that of functionality, which is the basic statement of the system's capabilities, services, and behavior.

★ Functionality And Architecture

Functionality and quality attributes ase osthogonal. If functionality & quality attributes were not osthogonal, the choice of function would dictate the level of security or performance or availability or usability.

Clearly though, it is possible to independently choose a desired level of each. This is not to say that any level of any level of any quality attribute is achievable with any function.

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.

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 purpose. In this way, functionality is largely independent of structure. Software architecture constrains its allocation to structure when other quality attributes are important.

* Architecture & Quality Attributes

Achieving quality attributes must be considered throughout design, implementation, and deployment. No quality attribute is entirely dependent on design, nor is it entirely dependent on implementation or deployment.

- Usability involves both architectural & nonarchitectural aspects. The nonarchitectural aspects include making the user interface clear & easy to use. Should you provide a radio button or a check box? What screen layout is most intuitive? Although these details matter tremendously to the end user & influence usability.

- Modifiability is determined by how functionality is divided (architectural) & by coding techniques within a module (nonarchitectural).

- Performance involves both architectural & nonarchitectural dependencies. It depends partially on how much comm^2 is necessary among components (architectural), partially on what functionality & on how shared resources are allocated & the choice of algorithms to implement selected functionality.

Note -

1. Architecture is critical to the realization of many qualities of interest in a system,

2. Architecture by itself, is unable to achieve qualities.

* Classes of Quality attributes →

1. Quality of the system. Mainly focus on availability, modifiability, performance, security, testability, & usability.

2. Business qualities (such as time to market) that are affected by the architecture.

3. Qualities, such as conceptual integrity that are about the architecture itself although they indirectly affect other qualities, such as modifiability.

Name : _____ Roll No. : _____

Subject : _____ Class : _____

Batch : _____ Division : _____

★ System Quality Attributes

- The definitions provided for an attribute are not operational. It is meaningless to say that a system will be modifiable. Every system is modifiable with respect to one set of changes & not modifiable with respect to another.

- A system failure an aspect of availability, an aspect of security, or an aspect of usability? All three attribute communities would claim ownership of a system failure.

- The performance community has "events" arriving at a system, the security community has "attacks" arriving at a system, the availability community has "failures" of a system, & the usability community has "user input."

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.

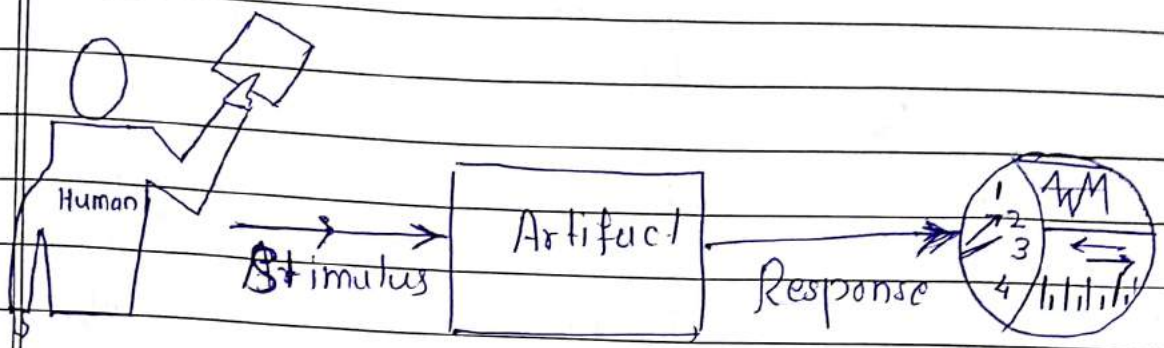
- Environment. The stimulus occurs within certain conditions. The system may be in an overload condition or may be running when the stimulus occurs, or some other condition may be true.

- 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.

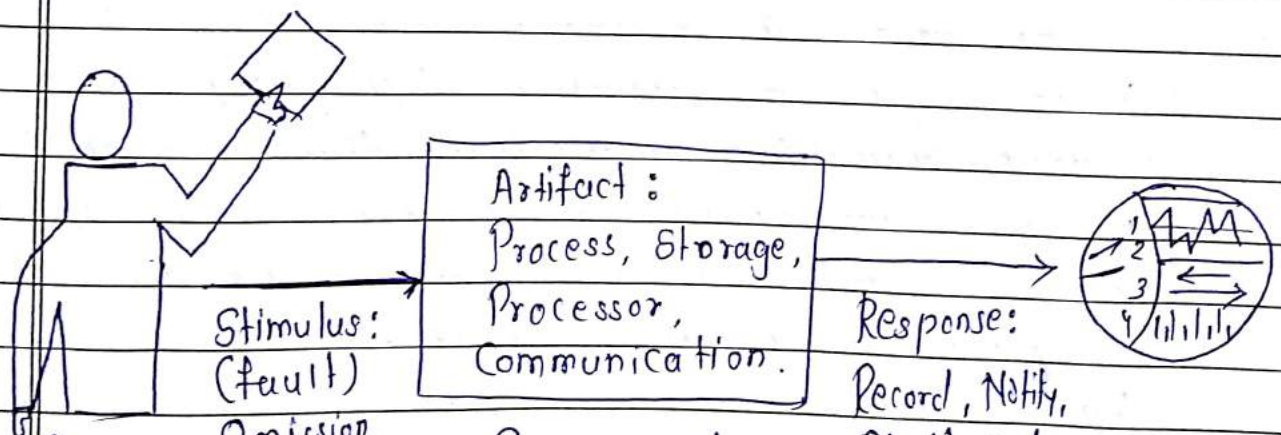
figure ① shows the parts of a quality attribute scenario.



Source of Stimulus | Environment | Response Measure

fig. ① Quality attribute Part.

figure ② derived the general scenario by instantiating each of the parts.



Source: Internal, External

Stimulus: (fault) Omission, Crash, Timing, Response.

Environment: Normal, Degraded Operation.

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

Fig ② Availability general scenarios.

figure (3) shows the pieces of this derived scenario.

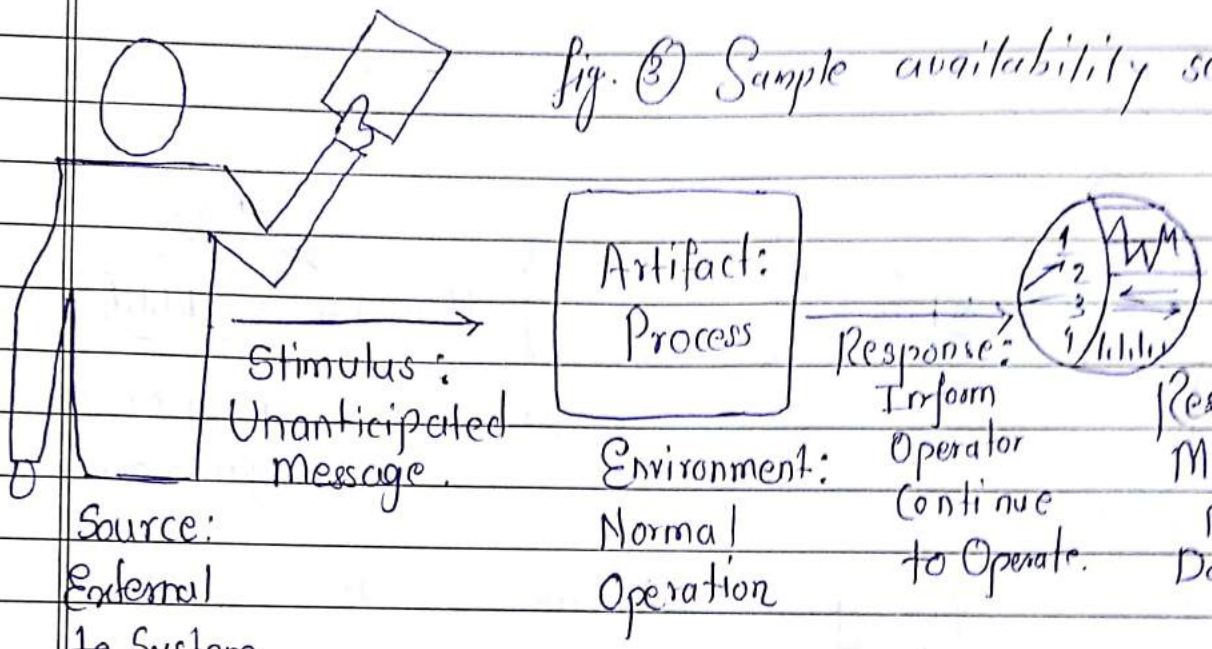


fig. (3) Sample availability scenario

The source of the stimulus is imp since differing response may be required depending on what it is.

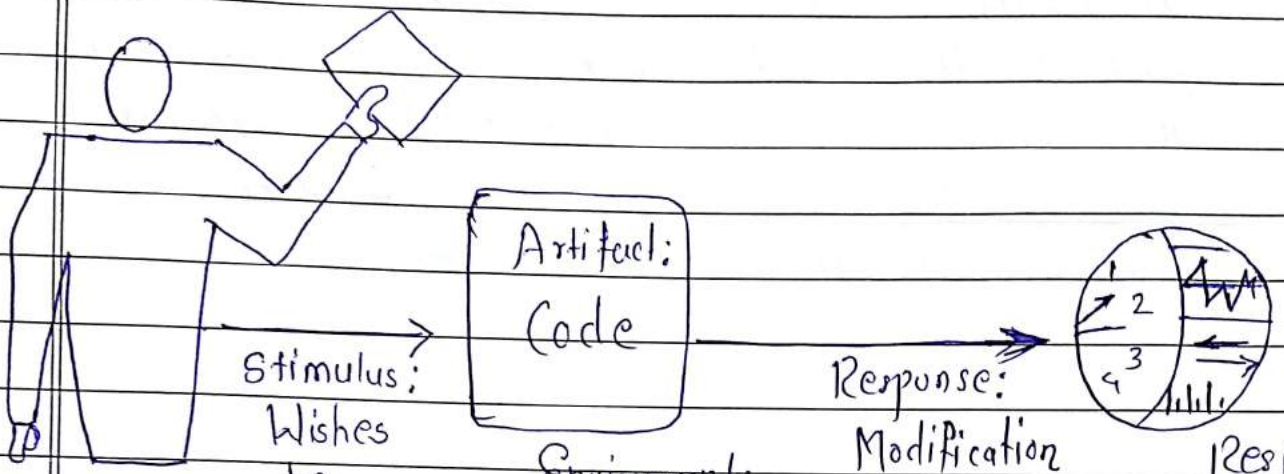
for ex-, a request from a trusted source may be treated differently from a request from an untrusted source in a security scenario. The environment may also affect the response, in that an event arriving at a system may be treated differently if the system is already overloaded. The artifact that is stimulated is less important as a requirement. It is almost always the system, & we explicitly call it out for two reasons.

Name : _____ Roll No. : _____

Subject : _____ Class : _____

Batch : _____ Division : _____

Figure (4) illustrates sample scenario



A collection of concrete scenarios can be used as the quality attribute requirements for a system. Each scenario is concrete enough to be meaningful to the architect, & the details of the response are meaningful enough so that it is possible to test whether the system has achieved the response. When eliciting requirements, we typically organize general scenarios by quality attributes, if the same scenario is generated by two different attributes, one can be eliminated.

Quality Attribute Scenarios in Practice

General scenarios provide a framework for generating a large number of generic, system independent, quality-attribute-specific scenarios. Each is potentially but not necessarily relevant to the system you are concerned with. To make the general scenarios useful for a particular system.

Availability :→

Availability is concerned with system failure & 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's - either humans or other systems.

Now to differentiate bet^n failures & faults. A fault may become a failure if not corrected or masked. That is, a failure is observable by the system's user & a fault is not. When a fault does become observable, it becomes a failure. For ex, a fault can be choosing

the wrong algorithm for a computation, resulting in a miscalculation that causes the system to fail.

The availability of a system is the probability that it will be operational when it is needed. This is typically defined as

$$\alpha = \frac{\text{mean time to failure}}{\text{mean time to failure} + \text{mean time to repair}}$$

* Availability General Scenarios

- Source of stimulus. We differentiate betⁿ internal & external indications of faults or failure since the desired system response may be different. In our ex, the unexpected message arrives from outside the system.
- Stimulus. A fault of one of the following classes occurs.
  - omission. A component fails to respond to an i/p
  - crash. The component repeatedly suffers omission faults.
  - timing. A component responds but the response is early or late.

- response. A component responds with an incorrect value.

- Artifact. This specifies the resource that is required to be highly available, such as a processor, communication channel, process, or storage.

- Environment. The state of the system when the fault or failure occurs may also affect the desired system response. for ex, if the system has already seen some faults & is operating in other than normal mode, it may be desirable to shut it down totally.
  However, if this is the first fault observed, some degradation of response time or function may be preferred. In our ex, the system is operating normally.

- Response. There are a number of possible reactions to a system failure. These include logging the failure, notifying selected users or other systems, switching to a degraded mode with either less capacity or less function, shutting down external systems, or becoming unavailable during repair.

- Response measure. The response measure can specify an availability percentage, or it can specify a time to repair.

Batch :

* Modifiability :° →

1. What can change (the artifact)?

A change can occur to any aspect of a system, most commonly, the functions that the system computes, the platform the system exists on (the hardware, operating system, middleware, etc.), the environment within which the system operates (the systems with which it must interoperate, the protocols it uses to communicate with the rest of the world, etc.), the qualities the system exhibits (its performance, its reliability, & even its future modifications), & its capacity (number of users supported, number of simultaneous operations.

2. When is the change made and who makes it (the environment)?

A change was made to source code. That is, a developer had to make the change, which was tested & then deployed in a new release. An end user changing the screen saver is clearly making a change to one of the aspects of the system. Equally clear, it is not in the same category as changing

the system so that it can be used over the Web rather than on a single machine. Changes can be made to the implementation (by modifying the source code), during compile (using compile-time switches), during build (by choice of libraries), during compile configuration setup (by a range of techniques, including parameter setting) or during execution (by parameter setting). A change can also be made by a developer, an end user, or a system administrator.

* Modifiability General Scenario

Portion of Scenario	Possible values
Source	This portion specifies who makes the changes - the developer, a system administrator, or an end user.
Stimulus	This portion specifies the changes to be made. A change can be the addition, deletion, modification and vary functionality, quality attribute, capacity.
Artifact	This portion specifies what is to be change ~~can~~ the functionality of a system, its platform, its user interface,

~~its~~ its user interface, its environment, or another system with which it interoperates.

Environment — This portion specified when the change can be made - design time, compile time, build time, initiation time or runtime. In our ex-, the modification is to occur at design time.

Response. — Whoever makes the change must understand how to make it, & then make it, test it & deploy it. In our example, the modification is made with no side effects.

Response measure. — All of the possible responses take time & cost money, & so time & cost are the most desirable measures. Extent to which this affects other functions or quality attributes. In ex., the time to perform the modification should be less than three hours.

* Performance →

Performance is about timing. Events (interrupts, messages, requests from users, or the passage of time) occur, & the system must respond to them. There are a variety of characterizations of event arrival and the response but basically performance is concerned with how long it takes the system to respond when an event occurs,

Figure 4.5 illustrate the performance general scenario

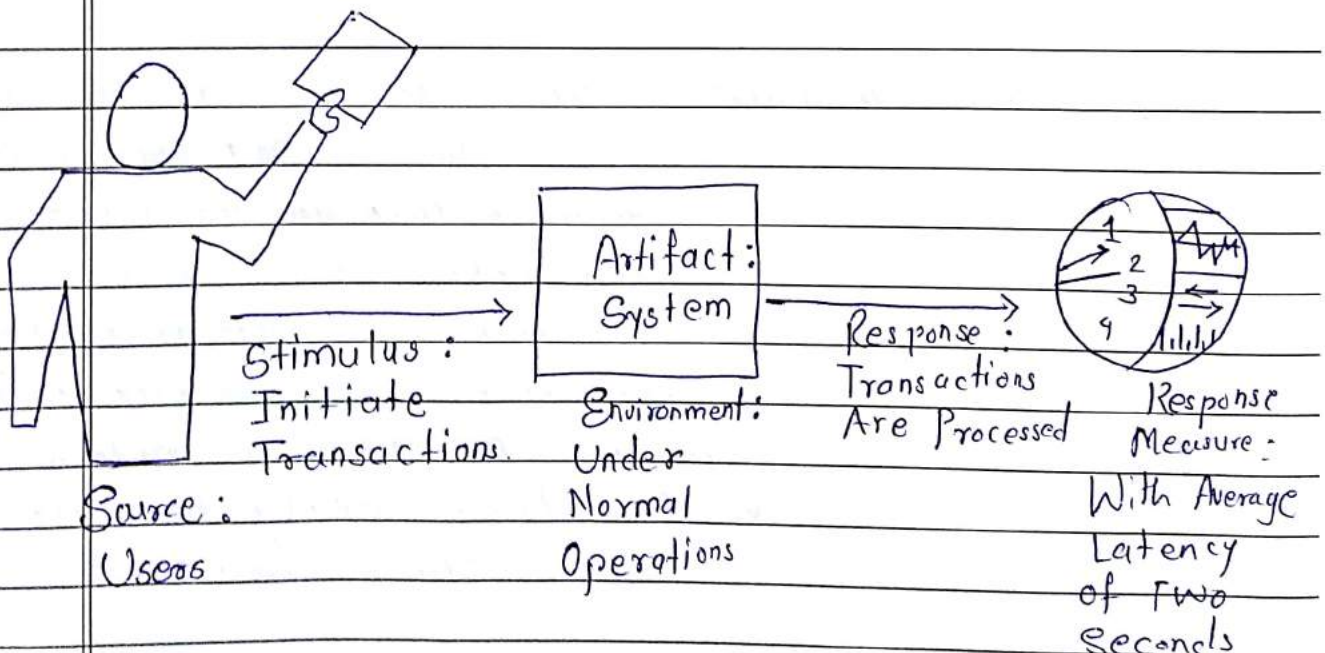


fig: Sample performance Scenario

SANDIP FOUNDATION'S
SANDIP INSTITUTE OF TECHNOLOGY & RESEARCH CENTRE
Mahiravani, Trimbak Road, Tal. & Dist. : Nashik - 422213, Maharashtra, India.

Name : _____ Roll No. : _____
Subject : _____ Class : _____
Batch : _____ Division : _____

* Security :→

Security is a measure of the system's ability to resist unauthorized usage while still providing its services to legi-timate users.

Security can be characterized as a system providing nonrepudiation, confidentiality, integ-rity, assurance, availability & auditing. For each term, a definition & an example.

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. This means you cannot deny that you ordered that item over the Internet if, in fact, you did.

2. Confidentiality is the property that data or services are protected from unauthorized access. This means that a hacker cannot access your income tax returns on a government computer.

3. Integrity is the property that data or services are being delivered as intended. This means that your grade has not been changed since your instructor assigned it.

4. Assurance is the property that the parties to a transaction are who they purport to be. This means that, when a customer sends a credit card number to an Internet merchant, the merchant is who the customer thinks they are.

5. Availability is the property that the system will be available for legitimate use. This means that a denial-of-service attack won't prevent your ordering this book.

6. Auditing is the property that the system tracks activites within it at levels sufficient to reconstruct them. This means that, if you transfer money out of one account to another account in Switzerland, the system will maintain a record of that transfer.

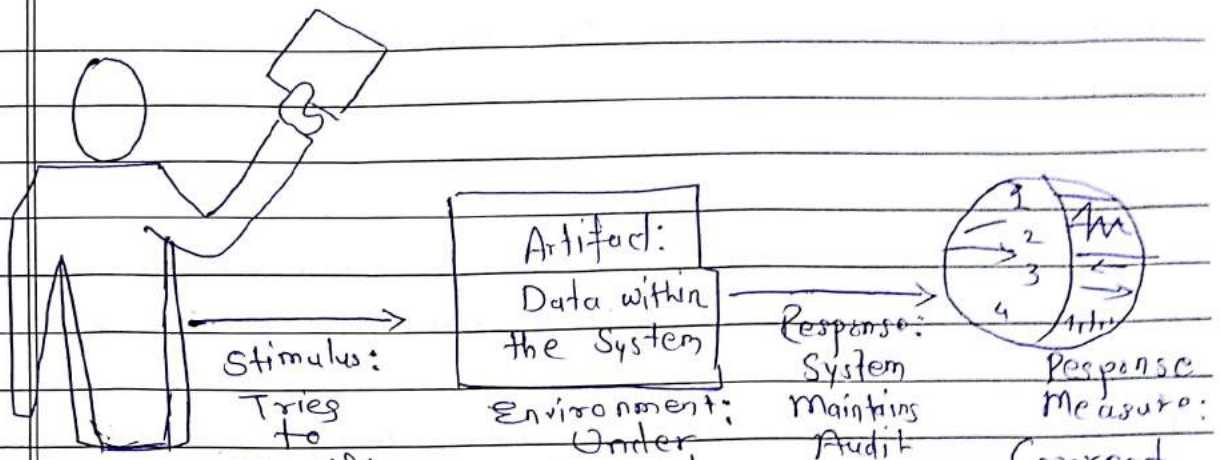


fig: Sample Security Scenario.

★ 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.

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 &/or services, allows access data &/or services, grants or withdraws permission to access/modify data/services by identity, stores data in an unreadable format, recognizes an unexplainable high demand for services, & informs a user or another system & 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 &/or services, percentage of services still available under denial of services attacks, restore data/services, extent to which data/services damaged &/or legitimate access denied.