Quality Attribute Generic Scenarios

Different quality attributes

**Availability** is concerned with system failure and duration of system failures. System failure means when the system does not provide the service for which it was intended.

**Modifiability** is about the cost of change, both in time and money.

**Performance** is about timeliness. Events occur and the system must respond in a timely fashion.

**Security** is the ability of the system to prevent or resist unauthorized access while providing access to legitimate users. An attack is an attempt to breach security.

**Testability** refers to the ease with which the software can be made to demonstrate its faults or lack thereof. To be testable the system must control inputs and be able to observe outputs.

**Usability** is how easy it is for the user to accomplish tasks and what support the system provides for the user to accomplish this. Dimensions:
- Learning system features
- Using the system efficiently
- Minimizing the impact of errors
- Adapting the system to the user’s needs
- Increasing confidence and satisfaction

Quality attribute scenarios

A Quality Attribute Scenario is a quality-attribute-specific requirement. There are 6 parts:

1. Source of stimulus (e.g., human, computer system, etc.)
2. Stimulus – a condition that needs to be considered
3. Environment - what are the conditions when the stimulus occurs?
4. Artifact – what elements of the system are stimulated.
5. Response – the activity undertaken after arrival of the stimulus
6. Response measure – when the response occurs it should be measurable so that the requirement can be tested.
Availability

General availability scenario

Source
Internal to system
External to system

Stimulus
Crash
Omission
Timing
No response
Incorrect response

Environment
Normal operation
Startup
Shutdown
Repair mode
Degraded (failsafe) mode
Overloaded operation

Response
Prevent the failure
Log the failure
Notify users / operators
Disable source of failure
Temporarily unavailable
Continue (normal / degraded)

Measure
Time interval available
Availability %
Detection time
Repair time
Degraded mode time interval
Unavailability time interval

Sample availability scenario

Source:
Heartbeat Monitor

Stimulus:
Process

Environment:
Normal Operation

Response:
Inform Operator
Continue to Operate

Measure:
No Downtime
**Availability tactics**

You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Availability: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html)
Availability tactics: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec2.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec2.html)
Modifiability

**General modifiability scenario**

Source
- End-user
- Developer
- System-administrator

Stimulus
- Add / delete / modify functionality, quality attribute, capacity or technology

Environment
- Runtime
- Compile time
- Build time
- Initiation time
- Design time

Response
- Make modification
- Test modification
- Deploy modification

Measure
- Cost in effort
- Cost in money
- Cost in time
- Cost in number, size, complexity of affected artifacts
- Extent affects other system functions or qualities
- New defects introduced

Sample modifiability scenario

Source: Developer
Stimulus: Wishes to change the UI
Environment: Design and unit tested
Response: Change made and unit tested
Measure: In three hours

Artifact
- Code
- Data
- Interfaces
- Components
- Resources
- Configurations...

Artifact: Code
Response: Change made and unit tested
Measure: In three hours
Modifiability tactics

More information
You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Modifiability: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html
Modifiability tactics: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec3.html
Performance

*General performance scenario*

![Diagram showing the relationship between source, stimulus, environment, response, and artifact.]

**Source**
- Internal to the system
- External to the system

**Stimulus**
- Periodic events
- Sporadic events
- Bursty events
- Stochastic events

**Environment**
- Normal mode
- Overload mode
- Reduced capacity mode
- Emergency mode
- Peak mode

**Response**
- Process events
- Change level of service

**Measure**
- Latency
- Deadline
- Throughput
- Jitter
- Miss rate
- Data loss

*Sample performance scenario*

![Diagram showing a sample scenario with a source, stimulus, artifact, environment, response, and measure.]

**Source**
- Users

**Stimulus**
- Initiate Transactions

**Environment**
- Normal Operation

**Response**
- Transactions Are Processed

**Measure**
- Average Latency of Two Seconds
**Performance tactics**

![Performance Tactics Diagram]

**More information**

You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Performance: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html)
Performance tactics: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec4.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec4.html)
Security

**General security scenario**

![Diagram](image)

**Artifact**
- System services
- Data within the system
- Component/resource of the system
- Data produced/consumed by the system

**Source**
- Identified user
- Unknown user
- Hacker from outside the organization
- Hacker from inside the organization

**Stimulus**
- Attempt to display data
- Attempt to modify data
- Attempt to delete data
- Access system services
- Change system’s behavior
- Reduce availability

**Environment**
- Normal mode
- Overload mode
- Reduced capacity mode
- Emergency mode
- Peak mode

**Response**
- Process events
- Change level of service

**Measure**
- Latency
- Deadline
- Throughput
- Jitter
- Miss rate
- Data loss

**Sample security scenario**

![Diagram](image)

**Source:** Disgruntled Employee from Remote Location

**Stimulus:** Attempts to modify pay rate

**Artifact:** Data within the System

**Environment:** Normal Operations

**Response:** System maintains audit trail

**Response Measure:**
- Correct data is restored within a day and source of tampering is identified
Security tactics

More information
You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Security: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html)
Security tactics: [http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec5.html](http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec5.html)
Testability

**General testability scenario**

![Diagram of testability scenario](image)

<table>
<thead>
<tr>
<th>Source</th>
<th>Stimulus</th>
<th>Environment</th>
<th>Response</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit tester</td>
<td>Execution of tests due to completion of code increment</td>
<td>Design time</td>
<td>Execute test suite &amp; capture results</td>
<td>Effort to find fault</td>
</tr>
<tr>
<td>Integration tester</td>
<td></td>
<td>Development time</td>
<td>Capture cause of fault</td>
<td>Effort to achieve coverage %</td>
</tr>
<tr>
<td>System tester</td>
<td></td>
<td>Compile time</td>
<td>Control &amp; monitor state of the system</td>
<td>Probability of fault being revealed by next test</td>
</tr>
<tr>
<td>Acceptance tester</td>
<td></td>
<td>Integration time</td>
<td></td>
<td>Time to perform tests</td>
</tr>
<tr>
<td>End user</td>
<td></td>
<td>Deployment time</td>
<td></td>
<td>Effort to detect faults</td>
</tr>
<tr>
<td>Automated testing tools</td>
<td></td>
<td>Run time</td>
<td></td>
<td>Length of longest dependency chain</td>
</tr>
</tbody>
</table>

- **Artifact**: Portion of the system being tested
- **Artifact**: Portion of the system being tested
- **Artifact**: Portion of the system being tested
- **Artifact**: Portion of the system being tested
- **Artifact**: Portion of the system being tested
Sample testability scenario

Source: Unit Tester

Stimulus: Code Unit
Completed

Environment: Development

Artifact: Code Unit

Response: Results Captured

Response Measure:
86% Path Coverage
in Three Hours

Testability tactics

Testability Tactics

Control and Observe System State

- Specialized Interfaces
- Record/Playback
- Localize State Storage
- Abstract Data Sources
- Sandbox
- Executable Assertions

Limit Complexity

- Limit Structural Complexity
- Limit Nondeterminism

More information
You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Testability: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html
Testability tactics: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec6.html
Usability

General usability scenario

Source
End user (possibly special role)

Stimulus
Use the system efficiently
Learn to use the system
Minimize impact of errors
Adapt the system
Configure the system

Environment
Runtime
Configuration time

Response
Provide features needed
Anticipate the user’s needs

Measure
Task time
Number of errors
Number of tasks accomplished
User satisfaction
Gain of user knowledge
Ratio of successful operations to total operations
Amount of time / data lost when error occurs

Sample usability scenario

Source: User

Stimulus: Downloads a New Application

Environment: Runtime

Response: User Uses Application Productively

Response Measure: Within Two Minutes of Experimentation
Usability tactics

More information
You can find more information in the book “Software architecture in Practice”. Please note that the website of the book contains an outdated version (version 2) while this document is based on version 3.

Usability: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch04lev1sec4.html
Usability tactics: http://www.ece.ubc.ca/~matei/EECE417/BASS/ch05lev1sec7.html
Table 1 – Availability Generic Scenario.

<table>
<thead>
<tr>
<th>Availability</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Internal to system or external to system</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Crash, omission, timing, no response, incorrect response</td>
</tr>
<tr>
<td>Artifact</td>
<td>System’s processors, communication channels, persistent storage</td>
</tr>
<tr>
<td>Environment</td>
<td>Normal operation; degraded (failsafe) mode</td>
</tr>
<tr>
<td>Response</td>
<td>Log the failure, notify users/operators, disable source of failure, continue (normal/degraded)</td>
</tr>
<tr>
<td>Response Measure</td>
<td>Time interval available, availability%, repair time, unavailability time interval</td>
</tr>
</tbody>
</table>

Table 2 – Modifiability Generic Scenario.

<table>
<thead>
<tr>
<th>MODIFIABILITY</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>End-user, developer, system-administrator</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Add/delete/modify functionality or quality attr.</td>
</tr>
<tr>
<td>Artifact</td>
<td>System user interface, platform, environment</td>
</tr>
<tr>
<td>Environment</td>
<td>At runtime, compile time, build time, design-time</td>
</tr>
<tr>
<td>Response</td>
<td>Locate places in architecture for modifying, modify, test modification, deploys modification</td>
</tr>
<tr>
<td>RespMeasure</td>
<td>Cost in effort, money, time, extent affects other system functions or qualities</td>
</tr>
</tbody>
</table>
Table 3 – Performance Generic Scenario.

<table>
<thead>
<tr>
<th>Performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>A number of sources both external and internal</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Periodic events, sporadic events, bursty, stochastic events</td>
</tr>
<tr>
<td>Artifact</td>
<td>System, or possibly a component</td>
</tr>
<tr>
<td>Environment</td>
<td>Normal mode; overload mode; reduced capacity mode</td>
</tr>
<tr>
<td>Response</td>
<td>Process stimuli; change level of service</td>
</tr>
<tr>
<td>RespMeasure</td>
<td>Latency, deadline, throughput, capacity jitter, miss rate, data loss</td>
</tr>
</tbody>
</table>

Table 4 – Security Generic Scenario.

<table>
<thead>
<tr>
<th>Scenario Portion</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>User/system who is legitimate/imposter/unknown with full/limited access</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Attempt to display/modify data; access services</td>
</tr>
<tr>
<td>Artifact</td>
<td>System services, data</td>
</tr>
<tr>
<td>Environment</td>
<td>Normal operation; degraded (failsafe) mode</td>
</tr>
<tr>
<td>Response</td>
<td>Authenticate user; hide identity of user; grant/block access; encrypt data; detect excessive demand...</td>
</tr>
<tr>
<td>RespMeasure</td>
<td>Time /effort/resources to circumvent security measures with probability of success</td>
</tr>
</tbody>
</table>
Table 5 – Testability Generic Scenario.

<table>
<thead>
<tr>
<th>Scenario Portion</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Unit developer, increment integrator, system verifier, client acceptance tester, system user</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Analysis, architecture, design, class, subsystem integration, system delivered</td>
</tr>
<tr>
<td>Artifact</td>
<td>Piece of design, piece of code, complete system</td>
</tr>
<tr>
<td>Environment</td>
<td>At design time, at development time, at compile time, at deployment time</td>
</tr>
<tr>
<td>Response</td>
<td>Provide access to state data values, observes results, compares</td>
</tr>
<tr>
<td>RespMeasure</td>
<td>% coverage; prob. of failure; time to perform tests; length of time to prepare test environment</td>
</tr>
</tbody>
</table>

Table 6 – Usability Generic Scenario.

<table>
<thead>
<tr>
<th>Scenario Portion</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>End user</td>
</tr>
<tr>
<td>Stimulus</td>
<td>Wants to: learn system, use system, recover from errors, adapt system, feel comfortable</td>
</tr>
<tr>
<td>Artifact</td>
<td>System</td>
</tr>
<tr>
<td>Environment</td>
<td>At runtime, or configure time, install-time</td>
</tr>
<tr>
<td>Response</td>
<td>(see below)</td>
</tr>
<tr>
<td>RespMeasure</td>
<td>Task time, number of errors, number of tasks accomplished, user satisfaction, gain of user knowledge, amount of time/data lost</td>
</tr>
</tbody>
</table>