Quality Attributes

Arnav Gupta

December 10, 2024

Contents

1 Quality Attributes

Includes reliability, availability, portability, scalability, and performance.

Information Capture and Dissemination Environment (ICDE): software system for providing intelligent assistance to analysts and researchers

ICDE automatically captures and stores data of actions performed by a user. Data can later by used by 3rd parties to offer intelligent help.

ICDE business goals:

- encourage 3rd party tool developers
  - integration: simple and reliable programmatic access to data store for 3rd party tools
  - portability: heterogeneous platform support for running 3rd party tools
  - reliability: allow 3rd party tools to communicate with ICDE users from a remote machine
- promote ICDE concept to users
  - scalability: scale data collection and data store components to support up to 150 users at a single site
  - scalability: low-cost deployment for each ICDE user workstation

Quality attributes must be made precise/measurable for a given system design.

1.1 Performance

Different ways to measure performance include: throughput, response time, deadlines

Enterprise applications often have strict performance requirements.

1.1.1 Throughput

Measure of amount of work an application must perform in unit time.

Required throughput can be average or peak.

Many systems have low average but high peak throughput requirements.

1.1.2 Response Time

Measure of latency an application exhibits in processing a request.

Usually measured in milliseconds, and often an important metric for users.

Required response time can be guaranteed or average.

1.1.3 Deadlines

Something must be completed before some specified time. Most often associated with batch jobs in IT systems.

1.1.4 Issues

Overheads of trapping user events must be imperceptible to ICDE users (response time). Solution is to decouple user event capture from storage using a queue.

1.2 Scalability

Main scalability issues are request load, connections, data size, and deployments

1.2.1 Request Load

How an application behaves when simultaneous load grows.

Ideal solution, without additional hardware capacity is: as the load increases, throughput remain constant and response time per request increases only linearly.

Can scale by adding more hardware: scale-out $\rightarrow$ horizontal or scale-up $\rightarrow$ vertical

In reality, applications will exhibit a decrease in throughput and a subsequent exponential increase in response time:

- increased load causes increased resource contention (CPU, network, memory)

- each request consumes some additional resource (buffer space, locks, etc) in the application, and eventually these are exhausted

1.2.2 Connections

When the number of simultaneous connections to an application increases. Each connections consumes some resources and this can exceed the maximum number of connections.

1.2.3 Data Size

How an application behaves as the data it processes increases in size.

1.2.4 Deployment

How effort to install/deploy an application increases as installation base grows, by installing new users or new servers.

Solutions typically revolve around automatic download/installation.

1.2.5 ICDE

Scalability often overlooked but is a major cause of application failure. Also hard to predict/test/validate.

Reliance on proven designs and technologies is essential.

For ICDE, application should be capable of handling a peak load of 150 concurrent requests from ICDE clients. Relatively easy to simulate user load to validate this.

1.3 Modifiability

Modifications to a software system during its lifetime are certain. Modifiable systems are easier to change/evolve.

Assess modifiability in context of how a system is likely to change. No need to facilitate changes that are unlikely to occur, so avoid over-engineering.

Modifiability measures how easy it may be to change an application to cater for new functional or non-functional requirements.

Must estimate cost and effort of modifiability.

Scenarios could involve new features, replacing components, or porting applications.

1.3.1 Analysis

Impact not easy to quantify.

Best possible is:

- convincing impact analysis of changes needed
- demo of how solution can accommodate modification without change

Minimizing dependencies increases modifiablity. Changes isolated to single components are likely to be less expensive than those that affect other components as well.

1.3.2 ICDE

Range of events trapped and stored by ICDE client to be expanded.

3rd party tools communicate new message types.

Change DB or server technology user.

1.4 Security

Difficult and specialized multi-faceted quality attribute.

Much technology available, but requires deep knowledge of approaches and solution.

Authentication: applications can verify identity of users and other applications with which they communicate

Authorization: authenticated users and applications have defined access rights to system resources

Encryption: messages sent to/from application are encrypted

Integrity: ensures message contents are not altered in transit

Non-repudiation: message sender has proof of delivery and receiver is assured of sender's identity, so neither can subsequently refute participation in message exchange

Approaches include:

- internet application security (SSL, PKI)
- authentication and authorization in Java (JAAS)
- OS security
- DB security

1.4.1 ICDE Requirements

Authentication of ICDE users and 3rd party ICDE tools to ICDE server.

Encryption of data to ICDE server from 3rd party tools/users executing remotely over an insecure network.

1.5 Availability

Requirement for most IT applications.

Measured by the proportion of required time it is usable.

Related to reliability, unreliable $\Rightarrow$ poor availability.

Period of loss of availability determined by:

- time to detect failure
- time to correct failure
- time to restart application

Strategies for high availability:

- eliminate single points of failure
- replication and failover
- automatic detection and restart

Recoverability: capability to reestablish performance levels and recover affected data after an application or system failure

1.5.1 ICDE

Achieve 100% availability during business hours.

Plenty of scope for downtime for system upgrade, backup, and maintenance.

Include mechanisms for component replication and failover.

1.6 Integration

Ease with which application can be incorporated into broader application context.

Achieved by programmatic APIs and data integration.

1.6.1 Strategies

Expose application data for access by other components.

Offer services to read/write application data through an abstracted interface.

1.6.2 ICDE

Revolve around need to support 3rd party analysis tools.

Well-defined and understood mechanism for 3rd party tools to access data in the ICDE data store.

1.7 Other Quality Attributes

1.7.1 Portability

If an application can be easily executed on a different software/hardware platform to the one it has been developed for.

1.7.2 Testability

The ease at which an application can be tested.

1.7.3 Supportablity

The ease at which an application can be supported once it is deployed.

2 Design Trade-offs

Quality attributes interact with and affect each other.

Architects must create solutions that make sensible compromises since it is not possible to satisfy all competing requirements.

Still, must satisfy all stakeholder needs, which is difficult.