Common Quality Attributes

Availability	Need to be there when required i.e. can always get a call through
Performance	Control resource demand tactics i.e. robot voice doesn't start mid call
Modifiability	Can change an update system i.e. want to add video calls
Reliability	System does not fail i.e. call does not drop
Security	System is not vulnerable i.e. people can't listen in
Testability	Possible to test in any way required

Common Stakeholders

- User
- Service Provider
- Analysis
- Developer

Common Design Patterns

Layered	distinct layers, can only interact with adjacent layers
MVC	UI and application isolation, user interface changes often
Blackboard	think /r/place
N-tier Architecture	See Layered

Types of Structures

Modular	Static structures, focus on how functionality is divided up
Component & Connector	Runtime structures focused on interactions
Allocation	Mapping to Environments

Scenarios

Portion of Scenario	**Possible Values**
Source	Internal or external to the system (e.g. user)
Stimulus	Arrival of a periodic, sporadic or stochastic event
Artefact	System or one or more components in the system
Environment	Operational mode: normal, emergency, peak overload, overload
Response	Process events, change level of service
Response Measure	Latency, deadline, throughput, jitter, miss rate

Availability

Acronyms & Measures

- mtbf — mean time between failures
- mttr — mean time to repair
- usually defined to be the probability it will be there when you ask it to work:

$$\frac{mtbf}{mtbf + mttr}$$

Availability Tactics

Fault Detection

- Ping/Echo;
- System monitor (watchdog/heartbeat);
- Exception detection (system detections);
- Voting (triple modular redundancy)

Fault Recovery

Preparation & Recovery

- Active/Passive Redundancy;
- Spare;
- Exception handling (error codes, exception classes);
- Software upgrade

Reintroduction

- Shadow;
- State Resynchronisation (graceful restart);
- Rollback (un/co-ordinated checkpointing);
- Escalating restart;
- Non-stop forwarding

Fault Prevention

- Removal from service;
- Transactions (ACID);
- Process monitor
- Exception prevention (exception classes; smart pointers; wrappers)

Performance Tactics

Control Resource Demand

- Manage sampling rate
- Limit event response
- Prioritise events (priority queues)
- Reduce overhead
- Bound execution times (timeouts)
- Increase resource efficiency (CDNs?)

Manage Resources

- Increase resources (vertical scaling)
- Introduce concurrency (horizontal scaling)
- Replication
- Bound queue sizes
- Schedule resources (auto-scaling?)

Modifiability Tactics

Reduce Size of a Module

- Split a Responsibility (Single Responsibility principle)

Increase Cohesion

- Maintain semantic coherence;
- Abstract common services

Reduce Coupling

- Use encapsulation, intermediaries & wrappers;
- Raise abstraction level;
- Restrict communication paths

Defer Binding

- Use runtime registration;
- Use start-up time binding;
- Use runtime binding

Security Tactics

Detect Attacks

- Detect intrusion (IDS)
- Detect service denial (DoS)
- Verify message integrity (checksums?)
- Detect message delay

Resist Attacks

- Identify actors (Accountability)
- Authenticate actors
- Authorise actors
- Limit access
- Limit exposure
- Encrypt data
- Separate entities
- Change default settings

React to Attacks

- Revoke access
- Lock computer
- Inform actors

Recover from Attacks

- Maintain audit trail
- Restore (See Availability tactics)

Testability Tactics

Control and Observe System State

- Specialised interfaces
- Record/playback
- Localise state storage
- Abstract data sources
- Sandbox
- Executable assertions

Limit Complexity

- Limit structural complexity
- Limit non-determinism

Lifecycles

- V Model
  - "Verification & Validation" model
  - Extension of waterfall model
  - Testing planned in parallel with development
  - Coding phase joins these two sides
- RUP
  - "Rational Unified Process"
  - Kind of like a Gantt chart
- Spiral Model
  - Risk driven process model
  - Iterative design model
- Agile SCRUM
  - 2–4 week sprint
  - Daily scrum meetings
  - Most important issues tackled first as they are on top of backlog