Quality Attributes

Software Architecture
3rd Year – Semester 1
Lecture 11

Software Requirements

- Expectations from the Software/System
- Stakeholders/users/roles may have different requirements
  - End User: Needs the functionality work without issues, Wish the system to be easy to use
  - Project Manager: Low Cost
  - Admin/Security/Maintenance: May have their own specialized wishes
  - Sometimes there may be conflicts of requirements
- Needs Vs. Wants

Requirements…

Q: When to identify requirements?
A: At the start of the Project

Q: Who is responsible for requirements?
A: Business Analyst + Architect

Software Architect must identify “Architectural Significant Requirements” (ASR)
Software Architect should find a balance between needs & wants

Q: Why the Requirements should be clear?
A: Changes are costly, specially if the changes come in later stages

Types of Requirements

- Functional
  - What the system must do
  - How it must behave
  - How it must react
- Non-Functional
  - Quality Attributes
  - Business Requirements
  - Constraints

Exercise

- Write down the requirements for a Calculator on a Mobile Phone
- Functional
  - Basic + - * / operations
  - Support for Square root, etc…
  - Support for Brackets
- Non Functional
  - Should work for Android and iOS and later on Windows
  - Buttons should be placed in Number pad order
  - Support to add more operations later on (e.g. Log, Sin, Cos, …)
  - Landscape UI Support ??
  - How many digits to display on the UI ?? (e.g. 0.33333333333333333333)

Definition of Quality & Functionality

- Functionality: The capability of the software product to provide functions which meets stated and implied needs when software is used under specified conditions
- Quality: The extend to which a product satisfies stated and implied needs when used under specified conditions

Functionality & Quality

- Functionality does not determine the Architecture
- For a given set of Functional Requirement you can create an endless amount of Architectures to Satisfy the requirements
- Functionality and Quality Attributes are Orthogonal (difference dimensions)

Non-Functional Requirements

- Non-Functional requirements are not directly linked to any specific function
- They are qualifications that typically cover business and system quality requirements and have a big influence on the architecture
- When they are forgotten at the beginning of a project, it often results in major problems in later stages of the project

Quality Attributes (overview)

- Qualifications of the functional requirements or of the overall product
  - Runtime Qualities – Performance
  - User Qualities – User Friendliness / UX
- Measurable and Testable properties of the system that are used to indicate how well the system satisfies the needs of the stakeholders
- There are many Quality Attributes and the list keeps growing…

Business Requirements

- Business or strategic decisions
  - Cost
  - Time to Market
  - System Lifetime
- Strategic Trade-offs are made
  - E.g. Building Brand New Software Vs. Purchasing Existing Software

Constraints

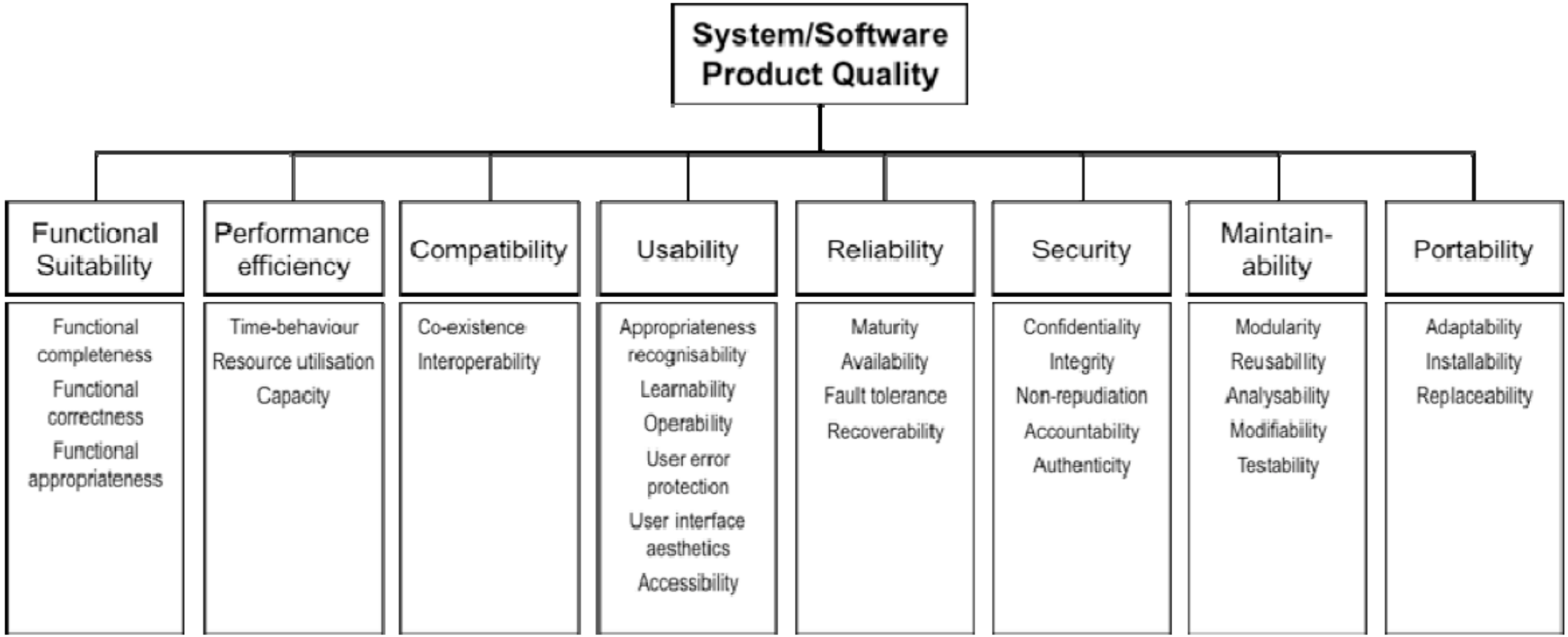
- Decisions/Agreements arrived beforehand
- No freedom to change and No Trade-offs can be made
  - E.g. Must use Open Source Software for Development

Software Architect has to make sure the Architecture Adheres to the constraints

Quality Attributes

Design Qualities	Run-time Qualities	System Qualities	User Qualities
• Modifiability • Reusability • Maintainability	• Performance • Reliability • Availability • Security • Scalability • Interoperability	• Testability • Supportability	• Usability • Accessibility

Quality Attribute – ISO 25010 Classification



Nature of Quality Attributes

- Quality Attributes may be related to one use case or part/module of the system or system as a whole
- Quality Attributes typically relate to different phases of the system life cycle such as design time and runtime qualities
- Quality Attributes may impact on each other
  - Performance is affected by almost all the other Quality Attributes

Role of an Architect on Quality Attributes

- Understand the Importance and Priority of the Quality Requirements of the System
- Evaluate and make Trade-Offs to meet the Quality Levels to satisfy the Stakeholders

Issues with Quality Attributes Definitions

- Grouped differently by different authors
- Overlapping (or Similar) Attributes
  - Integrity & Security
- Confusion with definition/meaning of Attributes
  - Performance: Responsiveness, Efficiency (CPU % Use)

Quality Attribute: Availability

- System Services are available when and where the users need to use them
- Proportion of time that the system is functional and working
- Affected by
  - System Errors
  - Maintenance Tasks
  - Infrastructure Failures
  - Malicious Attacks
  - System Load
  - Dependent Services
- Can be measured as a percentage of the total system downtime over a predefined period

Quality Attribute: Interoperability

- Ability of a system to exchange information successfully by communicating with other systems
- Interoperability Considerations
  - Syntactic: Ability to communicate between systems using specified data & communication protocols
  - Semantic: Ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results

Quality Attribute: Modifiability

- Cost to make a change
  - Lower cost to change → higher in Modifiability
- Change Types
  - Functional Change
  - Environment Change (Host OS/Platform)
  - Protocols (Communication Protocols, etc…)
  - Dependencies (MySQL → Oracle Vs. MySQL → MongoDB)

Quality Attribute: Performance

- Performance is an indication of the responsiveness of a system to execute any action within a given time interval.
- It can be measured in terms of latency or throughput. Latency is the time taken to respond to any event.
- Throughput is the number of events that take place within a given amount of time.

Quality Attribute: Reliability

- Ability of a system to remain operational over time
- Probability that a system will not fail to perform its intended functions over a specified time interval
- How users and other systems can be dependable of a given software, protocol, hardware, etc...
- Affected by other attributes
  - Availability
  - Accuracy
  - Predictability

Quality Attribute: Reusability

- Capability for components and subsystems to be suitable for use in other applications and in other scenarios
- Minimizes the duplication of components and also the implementation time

- E.g. Software Libraries

Quality Attribute: Scalability

- Ability of a system to either handle increases in load without impact on the performance of the system, or the ability to be readily enlarged
  - Load
  - Functions
  - Geographic
- Methods
  - Horizontal: Add more Servers
  - Vertical: Add/Improve hardware (e.g. CPU) of the same Server

Quality Attribute: Security

- Capability of preventing Malicious Attacks
  - Virus
- Preventing unauthorized usage
- Protecting System Assets
  - Data
  - Hardware

Question:

Denial of Service: Is this a Security issue?
Is this only a Security issue?

Quality Attribute: Testability

- Create test criteria for the system and its components, and to execute these tests in order to determine if the criteria are met
  - How much can be tested?
  - How much time it takes to test?
- Testability makes it more likely that faults in a system can be isolated in a timely and effective manner

Quality Attribute: Usability

- How well the application meets the requirements of the user and consumer by being intuitive, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience.
- Usability Considerations
  - 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

Architectural Attributes & Considerations

- 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 : 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

Quality Attribute Scenarios (QAS)

- Universal for Formal way to express Quality Attributes
- The goal of QAS is to capture and document unambiguous and testable requirements
  - Document similar to Use Case scenarios

NEXT LECTURE:

Template to express Quality Attributes

How to apply it

Concrete System Specific Qualities

Tactics

- An architectural tactic is a means of satisfying a quality attribute response measure by manipulating some aspect of a quality attribute model through architectural decisions

NEXT LECTURE:

Ways to improve Quality Attributes

Tactics Framework