Requirements and Architecture

System Requirement Categories

Functional requirements - what the **system must do**, how it must **react to run-time stimuli**

Non-functional - these requirements **qualify functional requirements**

- **Quality attributes** (the “abilities”)
- **Business**
- **Architectural**

Constraints - design decisions with **zero degrees of freedom**; i.e., a design decision that has already been made for you.

Quality Attribute(QA) Considerations

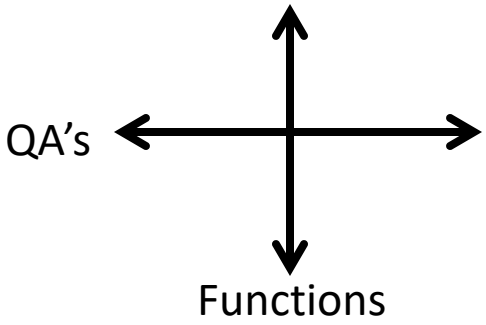
"A QA is a **measureable** or **testable** property of a system that is used to indicate **how well the system satisfies the needs** of its stakeholders." (SAiP p.63)

If a functional requirement is "when the user decides to change runtime preferences a change preferences dialog shall appear":

- a **performance** QA might describe **how quickly** the dialog will appear;
- an **availability** QA might describe **how often this function will fail**, and **how quickly it will be repaired**;
- a **usability** QA might describe how **easy** it is **to learn** this function.

Functionality determines software architecture
true or false?

Functionality and Architecture



Functionality alone does not determine architecture

- Given a set of required functionality, there is no end to the architectures you could create to satisfy that functionality
- So functional design doesn't matter?

Functionality and quality attributes are orthogonal

- Many ways to implement functionality with varying degrees of quality

Architecturally Significant Requirements

Architecture designs build systems that **satisfy requirements**

- But they must look toward current and future needs (known and unknown)

An **architecturally significant requirement** (ASR) is a requirement that will have a **profound effect** on the architecture

ASR Examples?

- **Significant = high cost of change**

How do we find those?

- In requirements documents, right?

NOTE: ASRs are predominately but not exclusively comprised of **quality attributes. Quality is a multi-faceted measure!!**

Why is this hard?

High-quality requirements documents may **not exist**

Emphasis on functionality not quality attributes

Design Quality attributes (ASRs), are high level**.**

- "The system shall be modular"
- "The system shall be **easy to use**"
- "The system shall meet users' performance expectations"

Requirements documents do not include everything useful to the architect

- ASRs often derive from **business goals**
- Development environment

An **architect can't wait** for "finished" requirements

- **Proactive**ly **interview** stakeholders but also ….
- **Suggest** relevant ASR's for discussion

Collaborative Model

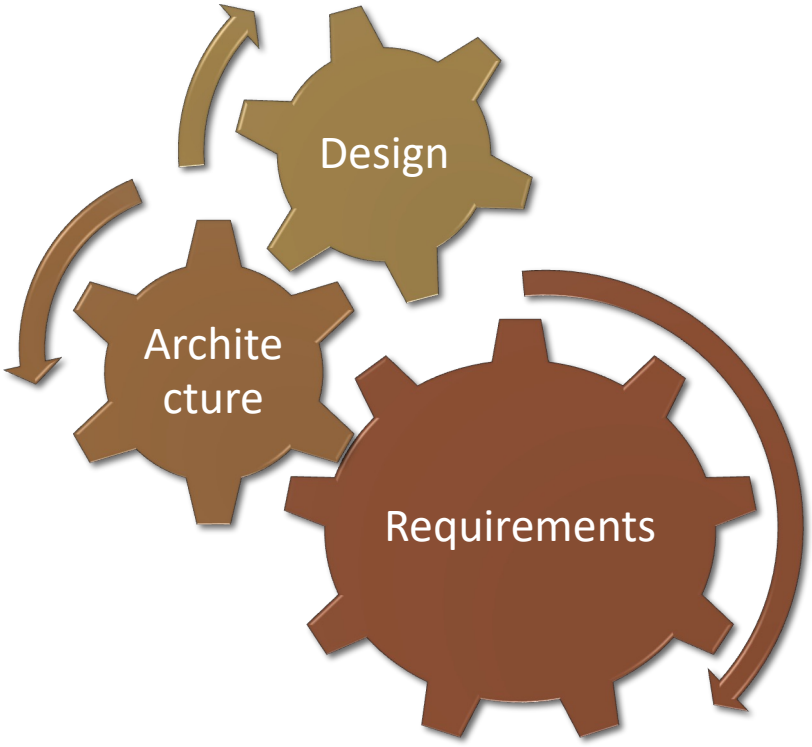
Interplay between **requirements and architecture**

Tradeoffs between system **problem and solutions**

- Challenges the tradition of avoiding solution thinking while discovering requirements

ASRs may only be **recognized** after some architecture **design**

Architecture feedback may **eliminate** infeasible **requirements** or poor cost/value benefits



Characteristics of ASRs

Hard to define and articulate - **general** and **abstract** concepts **users don't understand**

Needed early in the **life cycle** before needs are fully understood

Vaguely described, subjective; e.g., 24/7

Tend to be **neglected initially** – significance not appreciated

- Embedded in the expression of other requirements

Variable – subject to requirements and technology **change**

Situational – significance may depend on **system context**; e.g., scale, legacy systems, technology

Recognizing ASRs

Wide system impact

May involve resolving **requirements tradeoffs**

Strict (constraining, limiting, nonnegotiable) requirements that **dictate a design**

May i**nvalidate** conventional **design** tactic **decisions**

Difficult to achieve technically

Architecture: Quality Attributes

Operational categories

- Availability
- Interoperability
- Reliability
- Usability
- Performance
- Deployability
- Scalability
- Monitorability
- Mobility
- Compatibility
- Security
- Safety

Developmental categories

- Modifiability
- Variability
- Supportability
- Testability
- Maintainability
- Portability
- Localizability
- Development distributability
- Buildability
- Portability

F and N-F and QA => ASRs

Functional	Non-Functional
• Display a list of all products available to purchase • Organize lists by product category • Display name, picture, description, price • Catalogue wide search • Shopping basket • Support all major credit cards • Valid payment through cc agency	• Support at least 1000 transactions per day • Handle peaks of 10 transactions/ second • Allow 5000 concurrent users • 24/7 Availability • Data loss rate of 0% • Order completion to payment confirmation within 5 seconds (95% of the time) • Login complete within 5 seconds

Discussion: What ASRs come from which requirements? Which are ASRs? Which are not?

Business Goals May Drive ASRs

Market share, competition

Product lines

Global markets

Revenue

Cost to develop, deploy, operate, and maintain

Personnel objectives

Liability, safety, reputation

Standards and regulations

Intellectual property

Environmental and sustainability concerns

Business Qualities (System QAs)

Time to market (buy or reuse)

Cost and **benefit** (tradeoffs)

Projected **lifetime** (modifiability, scalability, portability)

Target **market** (product line strategy, reliability, performance, usability)

Rollout schedule (scope and scale, deployment flexibility)

Integration with legacy and external systems (interfaces, constraints, interoperability)

Constraints (safety, security, modifiability, …)

Regulatory requirements

Architectural Qualities

Conceptual integrity – consistency, do similar things in similar ways

Correctness and completeness

Traceability – Can you trace back from your architecture decision to the original requirement?

- **e.g. The architecture requires a FIPS-140 certified encryption aglorithm to be accepted by the Pentagon as a customer**
- **e.g. The architecture requires all forms received to be processed in background without user intervention to satisfy 24/7 operational requirements**
- **e.g. The architecture requires user pages to be translatable to Simplified Chinese and satisfy all Section-508 accessibility requirements (Client usability spec)**
- **e.g. The architecture must allow backups of full data-sets within 1 hour to meet downtime constraints**

Traceability Matrix example

ASR	Original req.	Source
FIPs libraries	FIPS-140 certified encryption aglorithm to be accepted by the Pentagon as a customer	Business Goals
508-Compliance	… Will be sold in China mainland and satisfy all Section-508 accessibility requirements	Marketing goals
Simplified Chinese translation and character sets	… Will be sold in China mainland and satisfy all Section-508 accessibility requirements	Marketing goals

Conceptual Integrity