

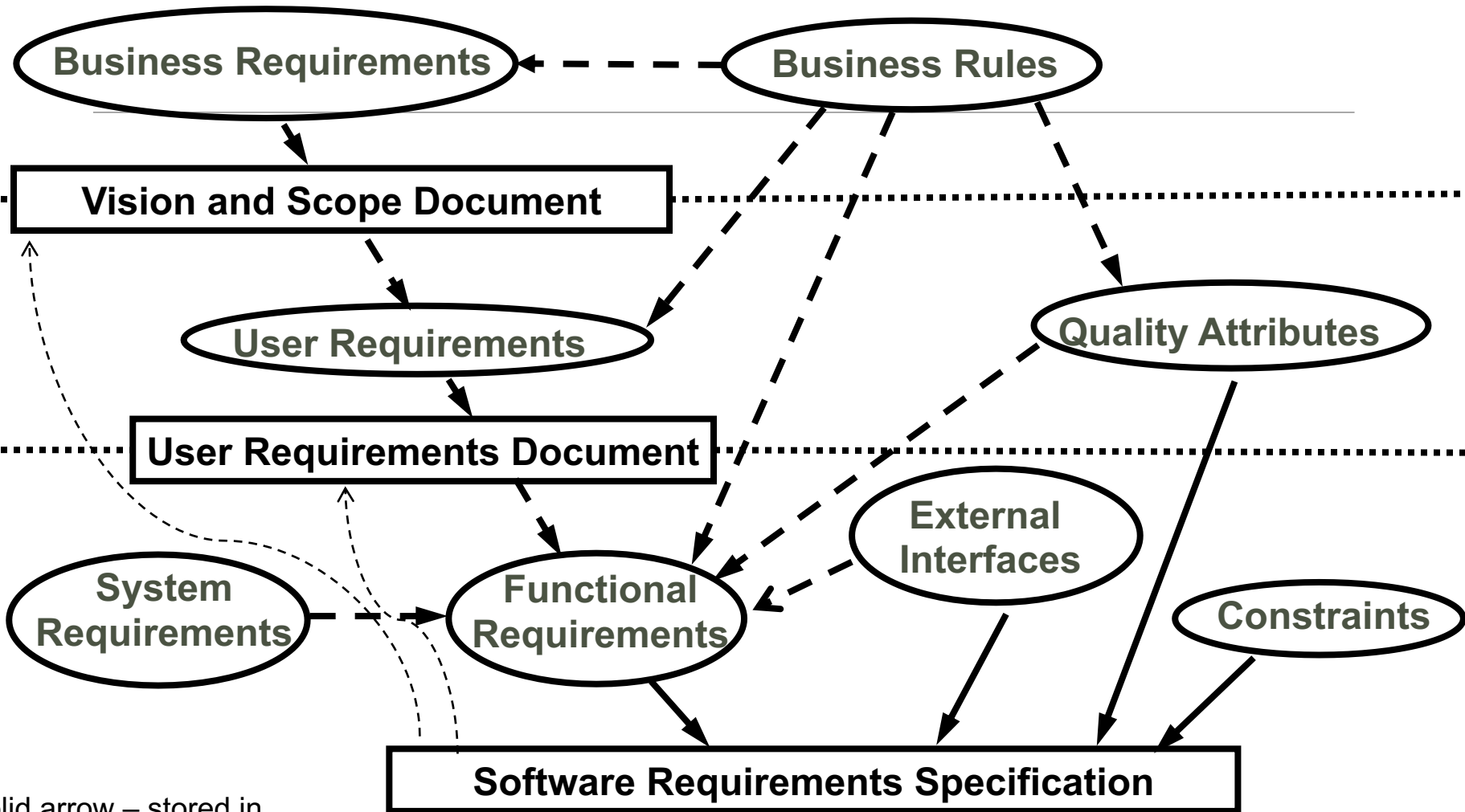
# Functional, Non- Functional Requirements

---

QUALITY ATTRIBUTES AND OTHER NF ITEMS

A solid orange horizontal bar at the bottom of the slide.

# Types of Requirements



Solid arrow – stored in  
Dotted arrow – origin of or influence

# Types of Requirements

---

- **Business Requirement:** A high-level business objective of the organization that builds a product or of a customer who procures it
- **Business Rule:** A policy, guideline, standard, or regulation that defines or constrains some aspect of the business
- **Constraint:** A restriction that is imposed on the choices available for the design and construction of a product
- **External Interface Requirement:** A description of a connection between a software system and a user, another software system or a hardware device
- **Functional Requirement:** A description of a behavior that a system will exhibit under specific conditions
- **Nonfunctional Requirement:** A description of a property or characteristic that a system must exhibit or a constraint that it must respect
- **Quality Attribute:** A nonfunctional requirement that describes a service or performance characteristic of a product
- **System Requirement:** A top-level requirement for a product that contains multiple subsystems
- **User Requirement:** A goal or task that specific classes of users must be able to perform with a system, or a desired product attribute

# Functional requirements

---

Functional requirements can have implications on your architecture

It is the architect's job to extract the ASRs from the functional requirements.

Examples of functional requirements

- Use cases
- Workflows
- Feature Descriptions

Sometimes the functional requirements can conflict with other architectural needs. Again, the architect must find and resolve these conflicts.

# Nonfunctional Requirements

---

## **System requirements**

- Associated hardware, inputs and outputs, major software components
- System architecture is not necessarily the software architecture

## **External interfaces**

- Interfaces to devices and systems across the system boundary
- E.g, legacy systems, services, peripheral devices

## **Constraints – limiting factors**

- E.g., technology choices, standards, business rules, laws, etc.

## **Quality**

Items affecting the overall ‘goodness’ of the system

# Definition & Advice

---

## Definition:

“A Quality attribute is a **measurable** or **testable** property of a system that is used to indicate **how well the system satisfies the needs** of its stakeholders.” (SAiP p.63)

## Advice:

How do you know something is an important Quality attribute?

Ask yourself this question – ‘would people still use my application if we have a terrible score on this attribute?’

Some material in these slides is adapted from Software Architecture in Practice, 3rd edition by Bass, Clements and Kazman.

# Quality Attributes – Master List

- **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

# Descriptions

## Usability

Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, efficiency of interactions, and accessibility. The usability requirements specified here will help the user interface designer create the optimum user experience.

## Performance

State specific performance requirements for various system operations. If different functional requirements or features have different performance requirements, it's appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.

## Security

Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.

## Safety

Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.

## Others

Availability

Efficiency

Installability

Integrity

Interoperability

Modifiability

Portability

Reliability

Reusability

Robustness

Scalability

verifiability

# Scenarios

---

Web application for banking

Describe a:

- Performance Attribute
- Security Attribute
- Integrity Attribute
- Availability Attribute

Control system for driverless car

Describe a:

- Performance Attribute
- Security Attribute
- Safety Attribute
- Reliability Attribute

What would be a good example of a Portability Attribute?

# Requirements Assumptions and Dependencies

---

## **Business and technology**

**Assumptions** – **stuff you expect** to happen (versus known facts)

- Project impacts if incorrect or they change

**Dependencies** – **stuff** that is **needed** from **external sources**

Examples of “stuff” – schedules, budget, contracts, intellectual property, people resources, equipment resources, licenses, standards, regulations, open source software, external APIs, cloud services, development tools, ...

# Examples

---

If you are building a desktop application

- Connects to online brokerage system

List some assumptions

List some dependencies