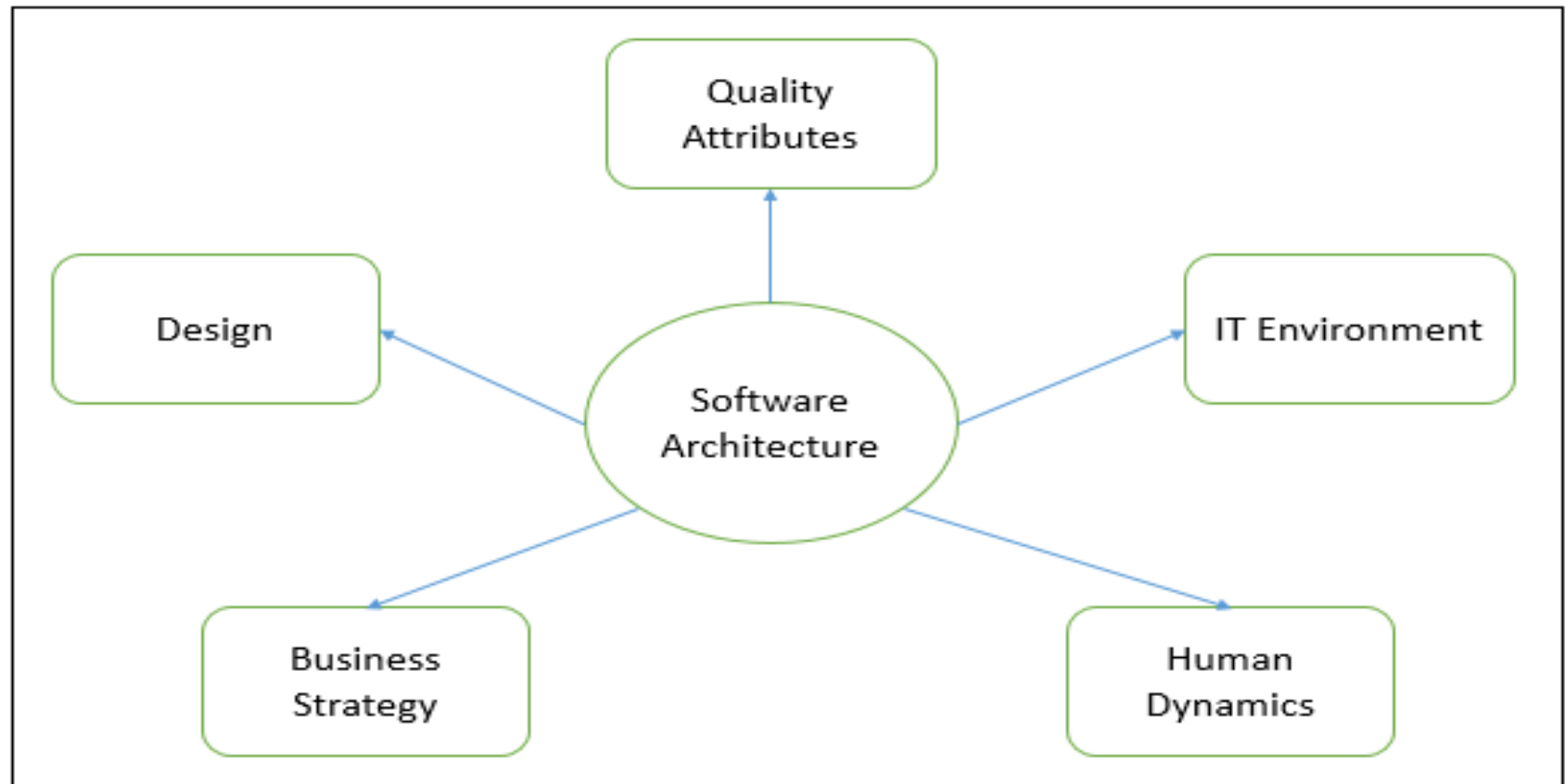


Software architecture

Software Architecture & Design Introduction

The architecture of a system describes its major components, their relationships (structures), and how they interact with each other. Software architecture and design includes several contributory factors such as Business strategy, quality attributes, human dynamics, design, and IT environment.



We can segregate Software Architecture and Design into two distinct phases: Software Architecture and Software Design. In **Architecture**, nonfunctional decisions are cast and separated by the functional requirements. In Design, functional requirements are accomplished.

An example of a **functional requirement** would be:

- A system must send an email whenever a certain condition is met (e.g. an order is placed, a customer signs up, etc).

A related **non-functional requirement** for the system may be:

- Emails should be sent with a latency of no greater than 12 hours from such an activity.

The functional requirement is **describing the behavior of the system** as it relates to the system's functionality. The non-functional requirement **elaborates a performance characteristic** of the system.

Typically non-functional requirements fall into areas such as:

- Accessibility
- Capacity, current and forecast
- Compliance
- Documentation
- Disaster recovery
- Efficiency
- Effectiveness
- Extensibility
- Fault tolerance

Software Architecture

Architecture serves as a **blueprint for a system**. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components.

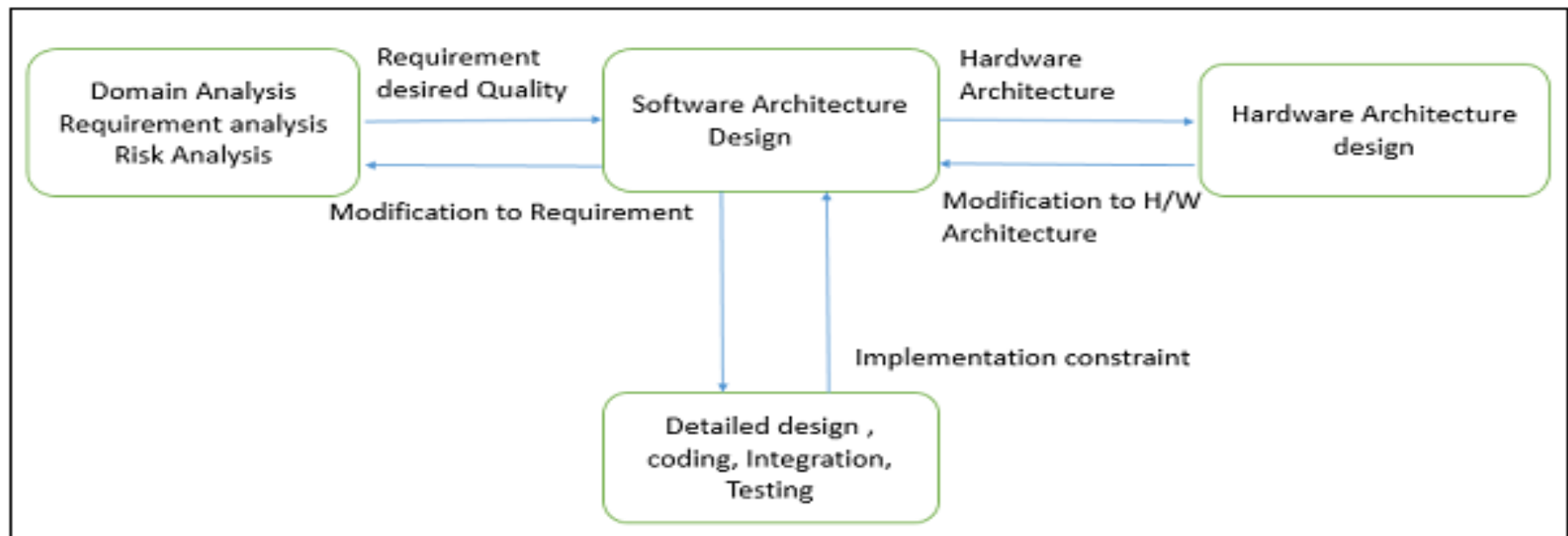
- ▣ It defines a **structured solution** to meet all the technical and operational requirements, while optimizing the common quality attributes like performance and security.
- ▣ Further, it involves a set of significant decisions about the organization related to software development and each of these decisions can have a considerable impact on quality, maintainability, performance, and the overall success of the final product. These decisions comprise of –
 - ▣ Selection of structural elements and their interfaces by which the system is composed.
 - ▣ Behavior as specified in collaborations among those elements.
 - ▣ Composition of these structural and behavioral elements into large subsystem.
 - ▣ Architectural decisions align with business objectives.
 - ▣ Architectural styles guide the organization.

Software design

Software design provides a **design plan** that describes the elements of a system, how they fit, and work together to fulfill the requirement of the system. The objectives of having a design plan are as follows –

- To negotiate system requirements, and to set expectations with customers, marketing, and management personnel.
- Act as a blueprint during the development process.
- Guide the implementation tasks, including detailed design, coding, integration, and testing.

It comes before the detailed design, coding, integration, and testing and after the domain analysis, requirements analysis, and risk analysis.



Goals of Architecture

The primary goal of the architecture is to identify requirements that affect the structure of the application. A well-laid architecture reduces the business risks associated with building a technical solution and builds a bridge between business and technical requirements.

Some of the other goals are as follows –

- ▣ Expose the structure of the system, but hide its implementation details.
- ▣ Realize all the use-cases and scenarios.
- ▣ Try to address the requirements of various stakeholders.
- ▣ Handle both functional and quality requirements.
- ▣ Reduce the goal of ownership and improve the organization's market position.
- ▣ Improve quality and functionality offered by the system.
- ▣ Improve external confidence in either the organization or system.

Limitations

Software architecture is still an emerging discipline within software engineering. It has the following limitations –

- ▣ Lack of tools and standardized ways to represent architecture.
- ▣ Lack of analysis methods to predict whether architecture will result in an implementation that meets the requirements.
- ▣ Lack of awareness of the importance of architectural design to software development.
- ▣ Lack of understanding of the role of software architect and poor communication among stakeholders.
- ▣ Lack of understanding of the design process, design experience and evaluation of design.