**Software Architecture**

Software Architecture defines fundamental organization of a system and more simply defines a structured solution. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team.

Software architecture defines a list of things which results in making many things easier in the software development process.

* A software architecture defines structure of a system.
* A software architecture defines behavior of a system.
* A software architecture defines component relationship.
* A software architecture defines communication structure.
* A software architecture balances stakeholder’s needs.
* A software architecture influences team structure.
* A software architecture focuses on significant elements.
* A software architecture captures early design decisions.

**Arquitectura de software**

La arquitectura de software define la organización fundamental de un sistema y, más sencillamente, define una solución estructurada. Define cómo se ensamblan los componentes de un sistema de software, su relación y la comunicación entre ellos. Sirve como un plan para la aplicación de software y la base de desarrollo para el equipo de desarrolladores.

La arquitectura de software define una lista de cosas que facilitan muchas cosas en el proceso de desarrollo de software.

- Una arquitectura de software define la estructura de un sistema.

- Una arquitectura de software define el comportamiento de un sistema.

- Una arquitectura de software define la relación de los componentes.

- Una arquitectura de software define la estructura de comunicación.

- Una arquitectura de software equilibra las necesidades de las partes interesadas.

- Una arquitectura de software influye en la estructura del equipo.

- Una arquitectura de software se centra en elementos significativos.

- Una arquitectura de software captura las primeras decisiones de diseño.

Source: https://www.geeksforgeeks.org/fundamentals-of-software-architecture/

**Software architecture** refers to the fundamental structures of a [software system](https://en.wikipedia.org/wiki/Software_system) and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations. The *architecture* of a software system is a metaphor, analogous to the [architecture](https://en.wikipedia.org/wiki/Architecture) of a building. It functions as a blueprint for the system and the developing project, laying out the tasks necessary to be executed by the design teams.

Software architecture is about making fundamental structural choices that are costly to change once implemented. Software architecture choices include specific structural options from possibilities in [the design of the software](https://en.wikipedia.org/wiki/Software_design). For example, the systems that controlled the [Space Shuttle](https://en.wikipedia.org/wiki/Space_Shuttle) launch vehicle had the requirement of being very fast and very reliable. Therefore, an appropriate [real-time computing](https://en.wikipedia.org/wiki/Real-time_computing) language would need to be chosen. Additionally, to satisfy the need for reliability the choice could be made to have multiple redundant and independently produced copies of the program, and to run these copies on independent hardware while cross-checking results.

La **arquitectura de software** se refiere a las estructuras fundamentales de un sistema de software y a la disciplina de crear dichas estructuras y sistemas. Cada estructura comprende elementos de software, relaciones entre ellos y propiedades tanto de los elementos como de las relaciones. La arquitectura de un sistema de software es una metáfora, análoga a la arquitectura de un edificio. Funciona como un plano para el sistema y el proyecto en desarrollo, estableciendo las tareas necesarias que deben ejecutar los equipos de diseño.

La arquitectura del software consiste en tomar decisiones estructurales fundamentales que son costosas de cambiar una vez implementadas. Las elecciones de arquitectura de software incluyen opciones estructurales específicas a partir de las posibilidades en el diseño del software. Por ejemplo, los sistemas que controlaban el vehículo de lanzamiento del transbordador espacial tenían el requisito de ser muy rápidos y muy fiables. Por lo tanto, habría que elegir un lenguaje de computación en tiempo real adecuado. Además, para satisfacer la necesidad de fiabilidad se podría optar por tener múltiples copias redundantes e independientes del programa, y ejecutar estas copias en hardware independiente mientras se verifican los resultados.

Source: https://en.wikipedia.org/wiki/Software\_architecture

**Enterprise architects:** Enterprise architects look at the high-level strategy of the company. They may analyze the most efficient technology for a company to use or determine the best end-to-end tech strategies to use to meet the business’s goals. Enterprise architects are concerned with how all the company's solutions interact and how they can be improved. EAs are commonly involved in the solution life cycle. They’re concerned with the technology mesh across the entire enterprise and they delegate their recommended solutions to others to implement. Enterprise architects are concerned primarily with innovation, security, and optimization.

**Software architects:** Software architects have a much more micro view than EAs. They define best practice standards and provide technical guidance and leadership for dev teams. They’re responsible for the design and strategic planning of new products, including software planning, design code methodology, and hardware planning. Software architects are concerned primarily with deployment, compatibility, tech strategy, and support.

When enterprise architects and software architects work together, they can develop an accurate view of the entire IT landscape and how it fits into the future needs of the enterprise. Then they can work together to begin minimizing waste and optimizing resources to meet the priorities of the business.

**Arquitectos de empresa**: Los arquitectos de empresa examinan la estrategia de alto nivel de la empresa. Pueden analizar la tecnología más eficiente que puede utilizar una empresa o determinar las mejores estrategias tecnológicas de extremo a extremo para alcanzar los objetivos de la empresa. Los arquitectos empresariales se ocupan de cómo interactúan todas las soluciones de la empresa y de cómo pueden mejorarse. Los EA suelen participar en el ciclo de vida de las soluciones. Se ocupan de la malla tecnológica de toda la empresa y delegan sus soluciones recomendadas a otros para que las implementen. Los arquitectos de empresa se ocupan principalmente de la innovación, la seguridad y la optimización.

**Arquitectos de software**: Los arquitectos de software tienen una visión mucho más micro que los EA. Definen los estándares de las mejores prácticas y proporcionan orientación técnica y liderazgo a los equipos de desarrollo. Son responsables del diseño y la planificación estratégica de nuevos productos, incluida la planificación del software, la metodología del código de diseño y la planificación del hardware. Los arquitectos de software se ocupan principalmente de la implantación, la compatibilidad, la estrategia tecnológica y el soporte.

Cuando los arquitectos empresariales y los arquitectos de software trabajan juntos, pueden desarrollar una visión precisa de todo el panorama de TI y de cómo encaja en las necesidades futuras de la empresa. A continuación, pueden trabajar juntos para empezar a minimizar el despilfarro y optimizar los recursos para satisfacer las prioridades de la empresa.

Source: <https://www.lucidchart.com/blog/enterprise-architect-vs-software-architect>

**Enterprise architecture** (**EA**) is an analytical discipline that provides methods to comprehensively define, organize, standardize, and document an organization’s structure and interrelationships in terms of certain critical business domains (physical, organizational, technical, etc.) characterizing the entity under analysis. The goal of EA is to create an effective representation of the business enterprise that may be used at all levels of stewardship to guide, optimize, and transform the business as it responds to real-world conditions. EA serves to capture the relationships and interactions between domain elements as described by their processes, functions, applications, events, data, and employed technologies.

**La arquitectura empresarial (EA)** es una disciplina analítica que proporciona métodos para definir, organizar, estandarizar y documentar de forma exhaustiva la estructura y las interrelaciones de una organización en términos de ciertos dominios empresariales críticos (físicos, organizativos, técnicos, etc.) que caracterizan a la entidad analizada. El objetivo de la EA es crear una representación eficaz de la empresa que pueda utilizarse en todos los niveles de dirección para guiar, optimizar y transformar la empresa a medida que responde a las condiciones del mundo real. La EA sirve para capturar las relaciones e interacciones entre los elementos del dominio descritos por sus procesos, funciones, aplicaciones, eventos, datos y tecnologías empleadas.

Source: <https://en.wikipedia.org/wiki/Enterprise_architecture>

**Relation between Software Architecture and Enterprise Architecture:**

From what I can tell, Enterprise Architecture is the one that dictates the “roadmap” for the enterprise business to reach it’s goals, give that, Software Architects prepare the blueprints for their team based on said “roadmap”.

**Relación entre la Arquitectura de Software y la Arquitectura Empresarial:**

Por lo que veo, la Arquitectura Empresarial es la que dicta el " roadmap" para que el negocio empresarial alcance sus objetivos, dado que, los Arquitectos de Software preparan los planos para su equipo basados en dicho " roadmap ".