UPT

Tarea 1: Historia de la Arquitectura de Software y diferencia entre diseño de Software y Arquitectura.

Arquitectura de Software

José Luis Jiménez Landa

**Historia de la Arquitectura de Software**

Las arquitecturas de software se volvieron muy importantes en la segunda mitad de los noventas ya que el origen de las ideas se remonta treinta años atrás cuando surgió la ingeniería de software en 1968, con ello vino la aparición de el primer ciclo de interés activo en componentes y su reutilización en tiempo real a inicios de 1980.

Desde el comienzo de la ingeniería de software, las arquitecturas de software siempre han estado con nosotros. Retornando a la década de los 60´s cuando los sistemas eran relativamente pequeños, se dibujaban diagramas abstractos para entender diseños y comunicarlos a los demás. Posteriormente a ello los sistemas crecieron en tamaño y ambiciones, entonces las descripciones estructurales se volvieron esenciales y fueron usadas naturalmente derivando similares prácticas en otras ramas de la ingeniería. En algunos proyectos militares fueron mandatorios varios niveles de descripción, por ejemplo en UK a través del standard militar JSP188 que pertenecía a hardware, el diseño de software MASCOT fue desarrollado para asistir las 4 capas de abstracción que demandaba el proceso.

Los niveles más altos de diseño es lo que conocemos como Arquitectura de Software. Aunque no se expresa todo el tiempo, incluso a inicios de las practicas de desarrollo de software, usábamos algunos niveles de descripciones de arquitecturas, a veces solo como diseño con descripciones pero después esas descripciones se volvieron esenciales para propósitos de mantenimiento y tuvieron que preservarse y ser mantenidos así.

En 1976, De Remer and Kron publicaron su seminario en papel “programando en grande” contra programando en pequeño. Hablaban de sistemas en dos niveles, el nivel de lenguaje de control y el nivel de programas en secuencia que eran controlados por el primero. Esto ha sido la línea de investigación sobre los lenguajes de interconexión de módulos durante los pasados 20 años.

Shaw y Garlan fueron los pioneros en introducir la arquitectura de software como concepto de estudio, aunque ya vimos que siempre ha estado implícita en el desarrollo de software, está idea la pudimos ver marcada en el workshop de DARPA en 1990.

La idea esencial de la arquitectura de software es que el software a un alto nivel de abstracción puede ser descrito como un número de elementos distintos y subsistemas que poseen interconexión e interacciones entre ellos. Otra parte importante que acompaña al nivel de abstracción de diseño en largos sistemas es la reutilización efectiva de los componentes de software.

El 1968 en la primera celebración de la conferencia de ingeniería de software se volvió obvio que esta cosa llamada software se estaba volviendo muy difícil y engorrosa de mantener y que se necesitaban diferentes formas de enfrentar esta situación. Posteriormente con una publicación de Doug Mcllroy quien hablo sobre los componentes de software, aunque solo fuera como librerías de utilidad o librerías de rutinas matemáticas, es decir no lo que conocemos actualmente como componentes de software, dio a conocer que eran fragmentos de un sistema de software que se podían ensamblar con otros fragmentos para conformar grandes piezas o aplicaciones completas. De cualquier forma su visión a sido probada como correcta.

En 1970 surgieron los métodos estructurados, principalmente en US, mientras que en Europa emergía el diseño centralizado. Al final de los 70´s con un proyecto llamado DRACO en California se implemento el primer proyecto de software de reutilización por Peter Freeman. En dicho proyecto las ideas de componentes, niveles de abstracción y análisis de dominios fueron incluidos. El modulo de interconexión de lenguajes comenzó gracias a De Remer y Kron y fueron usados en DRACO, desarrollos como estos fueron puestos juntos en un excelente escrito por Ruben Prieto Diaz y Jim Vecino en 1986.

El siguiente boom de actividad concerniente a los componentes de software fue en la mitad de los 80´s cuando la gento comenzó a hablar sobre la organización industrial del desarrollo de software.

En 1984, Peter Wegner describió al desarrollo de software como una industrial de capital intensivo que gastaba grandes cantidades de dinero al inicio, invirtiendo en software para correr software (Sistemas operativos y middleware) y el software que ayudara a desarrollar software más efectivo.

En 1986 Brad Cox escribió un libro de programación orientada a objetos en donde introdujo la idea de circuitos integrados de software, la cual fue posicionada desde entonces como la solución a los problemas de desarrollo. Estos circuitos integrados de software son piezas de software que tu puedes conectar y desconectar o reconfigurar, vender en el mercado abierto y así. El desarrollo un pequeño lenguaje llamado Objective-C el cual podía ser embebido en C y preprocesado por C, pero que perdió cuando compitió con C++.

**Diferencia entre Arquitectura y Diseño de Software**

La palabra diseño puede ser utilizada como verbo o sustantivo. En la forma de verbo, diseño es el proceso de tener en consideración el diseño y en la forma de sustantivo el diseño es el resultado del proceso de diseñar algo.

El diccionario de Merriam-Webster provee de múltiples significados para la palabra diseño. Los más apropiados para nuestro fin son por ejemplo “concebir y planear en la mente” e “idear una función específica” así como “realizar un maquetado, pintura o patrón”. La función de un diseñador entonces podría ser definida como la combinación de arte, ciencia y tecnología para crear nuevos productos que la gente pueda utilizar.

El diseño de software es un proceso creativo que asegura que un conjunto de funcionalidades es entregado de manera eficiente, económica, estética, segura y a salvo enfocándose en todos los aspectos, esto para facilitar el entendimiento de dicho diseño del producto a aquellos que lo desarrollan y asegurando que el producto resultante sea fácil de utilizar, rentable, fácil, seguro, flexible, mantenible y vendible. Todo lo anterior será basándose en la ciencia y la ingeniería junto con la ciencia de lo estético y con el conocimiento de la completa organización.

La arquitectura de software acomoda los componentes de un sistema en forma tal que dicho sistema sea fácil de mantener, que facilite la funcionalidad del producto de manera eficiente escondiendo los componentes que necesitan ser ocultados facilitando así el mantenimiento del mismo y manteniéndolo tan pequeño como sea posible.

Las tres funciones principales al diseñar un producto son el diseño de componentes, el diseño de la integración de dichos componentes y organizar los componentes. En el diseño de la arquitectura de un producto nos enfocamos en la organización de los componentes del producto de software. La IEEE standard 610 en su glosario de términos de Ingeniería de Software define la arquitectura de diseño como el proceso de definir una colección de componentes de hardware y software y sus interfaces para establecer un framework para el desarrollo de un sistema de computación.

Referencias:

Software Architectures: Advances and applications, Leonor Barroca, Jon Hall, 2008

Software Design: A comprehensive Guide to Software Development Projects, 2010