CLASE 4 – TEORICO 24-08-21

DEFINICION DE ARQUITECTURA

* Es el conjunto de estructuras necesarias para **razonar** (*decidir e incidir en el contexto que estamos diseñando para favorecer los atributos de calidad*) sobre el sistema, que comprenden elementos de software, la relación entre ellos, y las propiedades de ambos, elementos y sus relaciones 🡪 Definición de Kazmann 3ed
* La funcionalidad no define las estructuras
* La funcionalidad no define la arquitectura
* Los requerimientos funcionales no determinan la arquitectura.
* Como va a estar construido? No te la determina la funcionalidad.
* La arquitectura potencialmente va a cambiar
* ***Los requerimientos no funcionales son los que definen nuestras estructuras, es decir nuestra arquitectura.***
* **Estructura:** es un conjunto de elementos vinculados por algún tipo de relación
* Estructuras de alto nivel
  + **Modulo:** describe las unidades de implementación que proveen un conjunto cohesivo de responsabilidades. Unidades de implementación, con mi código
  + **Componentes y conectores:** describe los elementos que tienen presencia en tiempo de ejecución y su forma de interacción. Ej.: assemblies, un archivo de ejecución, las cosas q yo resuelvo en tiempo de ejecución. Como se comporta nuestro sistema una vez que esta corriendo. Mi código en tiempo de ejecución.
  + **Asignación:** describe el mapeo entre las estructuras anteriores y la organización del desarrollo, la instalación y los ambientes de ejecución. La otra parte, la infraestructura. La asignación es el :Donde ,el entorno, el contexto

ESTRUCTURAS Y VISTAS

* Como resolvemos los atributos de calidad? Tácticas, patrones para resolverlos. Las tecnologías normalmente implementan esos mecanismos. Por eso es importante tener en cuenta las tecnologías que utilizamos, es parte de la decisión de un arquitecto. A las tácticas las vamos a razonar en ESTRUCTURAS.
* Como arquitecto voy a razonar pero muchas veces tengo que comunicar. Para eso basado en las estructuras, utilizo las VISTAS.
* Modelo de vistas: “Views and beyond”. Separa bien las cosas, no indica una notación (para representar arquitectura en general), pero vamos a usar UML para especificar las distintas vistas. La vista de módulos, la vista de componentes y conectores y la vista de asignación.
* VISTA: representación de un conjunto coherente de elemento s y sus relaciones desde la perspectiva de quienes la escriben o leen. Conjunto de estructuras que nos permiten razonar para tomar una decisión basado en una perspectiva en particular.
* Modelo especifico “Views and beyond” esta basado en las diferentes estructuras antes descriptas.
* ***VISTA DE MÓDULOS:*** QUE QUEREMOS COMUNICAR A PARTIR DE ESA VISTA?
  + **Descomposición :** como se organiza el código, organizar al equipo de desarrollo para ver todo mas claro. Nos ayuda a organizar al código.
  + **Uso:** que atributos de calidad podemos observar? Mantenibilidad, me sirve para ver el impacto del cambio.
  + **Clases:** organización de las clases, ayuda a pensar en colecciones de cosas que tienen el mismo uso. Si quiero reusar un determinado modulo, se que es lo que tengo que reutilizar. Permite saber donde tengo que agregar que código y en donde. Se usa cuando se quiere diseñar a nivel de código. Pensar en el reúso, para ir modificando en forma incremental ese código.
  + **Layers:** agrupaciones de responsabilidades, donde se agrupan los módulos, y que tiene cada módulo. Se puede ver el acoplamiento entre los layers. Es una división lógica. Puede llegar a servir para distribuir el trabajo en los equipos, para tener claro la dependencia entre los diferentes equipos.
* ***VISTA DE COMPONENTES Y CONECTORES:***
  + Representamos componentes (cosas que se manifiestan en tiempo de ejecución), puertos (la forma en que se comunican con el exterior con distintos componentes), interfaces (contratos, como me comunico con este componente) requeridas y provistas, conectores (forma de comunicar esos distintos componentes), vista interna de componentes (detalle de implementación).
  + UML: vamos a utilizar muchos estereotipos para ayudarnos a comunicar la arquitectura
* ***VISTA DE ASIGNACION:***
  + **Nodos**: elementos físicos que existe en tiempo de ejecución y representa un recurso computacional que tiene por lo menos algo de memoria, y algunas veces, capacidad de procesamiento. Se pueden especificar atributos: velocidad, memoria, etc.
  + **Artefactos**: especifican un elemento físico de información utilizado o producido por un proceso de desarrollo , o por el despliegue u operación de un sistema. Ejemplo: archivos, scripts, ejecutables. Se representan con el símbolo de clase con estereotipos.

ATRIBUTOS DE CALIDAD:

* característica de tu sistema
* tiene relación con los requerimientos no funcionales.
* Atributo medible de mi sistema.
* Los atributos de calidad los vamos a ir favoreciendo o no dependiendo de las tácticas que utilicemos.
* Patrones: conjunto de tácticas.

TACTICAS DE ARQUITECTURA

* Táctica individual que yo hago para favorecer determinado atributo de calidad.