

Arquitectura de Proyectos de IT

Apunte:
Comunicación de Arquitectura de Software



Autores:
Ing. Gustavo A. Brey (gbrey@sistemas.frba.utn.edu.ar)
Santiago Blanco (santiago.blanco@gmail.com)

Versión: 0.8.20081106

Comunicación de Arquitectura de Software

gbrey@sistemas.frba.utn.edu.ar	1
Comunicación de Arquitectura de Software	2
1. Introducción.....	3
2. Concepto de Comunicación y Entendimiento de Arquitectura	3
2.1. ¿Por qué comunicamos?.....	3
2.2. Stakeholders y Necesidades	3
2.3. Elementos y relaciones.....	4
2.3.1. Stakeholders.....	5
2.3.2. Concerns (Necesidades).....	5
2.3.3. Viewpoints (Perspectivas)	5
2.3.4. View (Vista)	5
2.3.5. Models (Modelos)	5
2.3.6. Architectural description	5
3. Frameworks de Arquitectura de Software.....	5
3.1. The 4+1 View Model of Architecture.....	5
3.1.1. Logical View	6
3.1.2. Development View.....	6
3.1.3. Process View	6
3.1.4. Physical View	6
3.1.5. Scenarios View.....	6
4. Guía/Metodología de Comunicación.....	6
4.1. Diagrama de Contexto y Overview Diagram	7
4.2. Identificar Stakeholders y Concerns	7
4.3. Definir los ViewPoint necesarios.....	7
4.4. Documentar y Comunicar las Vistas.....	8
4.5. Documentar y Comunicar los Escenarios.....	8
4.6. Documentar las decisiones (Constantemente).....	8
4.7. Armado de SAD (Wiki, Documento de Texto, etc)	8

1. **Introducción**

2. **Concepto de Comunicación y Entendimiento de Arquitectura**

2.1. **¿Por qué comunicamos?**

La comunicación de la arquitectura tiene los siguientes usos.

- **La Arquitectura como elemento principal para la comunicación entre stakeholders.**

El tipo de comunicación, es decir, lo que se desea transmitir, variara de acuerdo a cada tipo de stakeholder involucrado. Por ejemplo, a un tester, se le especificara la caja negra de las piezas de software que deben ser testeadas, mientras que a los desarrolladores se les estaría comunicando no solo el alcance del módulo o sistemas, sino también como construir la aplicación, sus restricciones, principios, estilos arquitectónicos y entorno de desarrollo a utilizar (ej, IDE, DB, como compartir el código, etc).

Pero sin duda que una de las personas más interesadas en la documentación de la arquitectura es el arquitecto. En el futuro, el arquitecto puede ser la misma u otra persona, y en este ultimo caso, estará mas interesado aun en contar con documentación que le permita observar las decisiones que se han tomado y el porque de las misma.

- **La Arquitectura sirve como medio de educación.**

El término de medio de educación proviene del hecho de que la documentación de arquitectura es usada para introducir a nuevos trabajadores en el entendimiento del sistema. Estas personas bien pueden ser nuevos empleados, analistas externos o un nuevo arquitecto.

Uno de los desafíos que nos hemos encontrado a lo largo de nuestra experiencia trabajando en proyectos es la de mantener a todo el equipo correctamente comunicado sobre la organización, tanto de los equipos como de los componentes de la aplicación, sus responsabilidades y dependencias, y creemos que tener un correcto plan de comunicación de la arquitectura es clave para poder lograrlo, y la arquitectura contiene dichas decisiones significativas que permiten la educación constante de los integrantes para los cambios durante la evolución y avance del proyecto. Muchas veces mantener un único documento actualizado con las últimas decisiones no siempre es lo más feliz y/o aconsejable, a lo largo del apunte veremos otros medios y que comunicar.

- **La Arquitectura sirve como base para el análisis del sistema.**

Esta documentación debe tener la información necesaria para poder evaluar una variedad de atributos tales como seguridad, performance, usabilidad, disponibilidad y modificabilidad. Uno de los pre requisitos para la evaluación de arquitecturas tiene que ver con tener la documentación de la arquitectura actualizada, de esa manera pudiendo evaluar si una arquitectura va a poder cumplir con los requerimientos, antes de implementar un sistema, es más que valorable.

2.2. **Stakeholders y Necesidades**

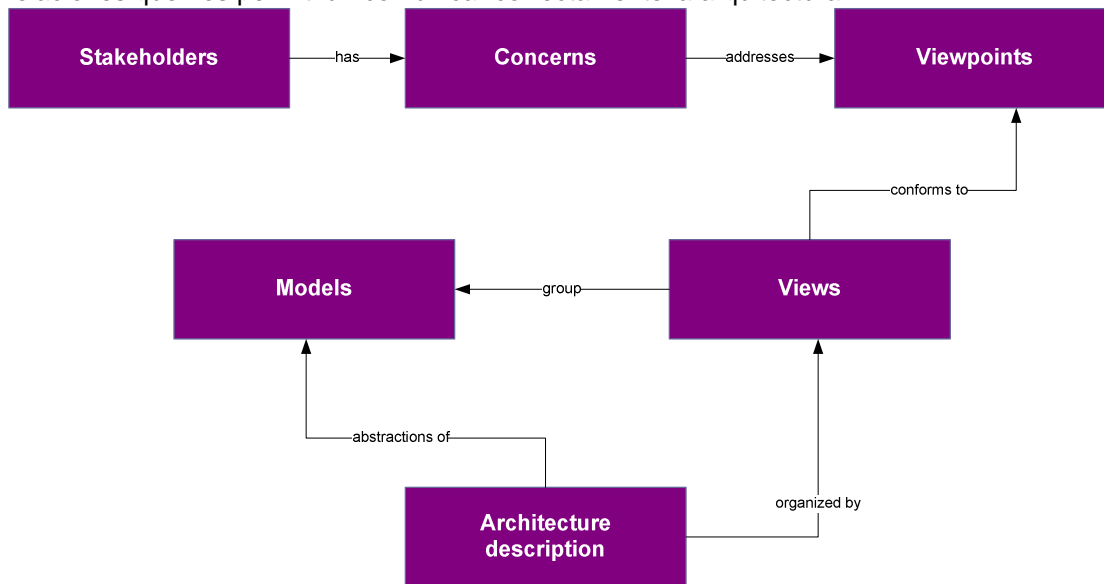
Es importante que la arquitectura necesita ser entendida y consensuada por los stakeholders. Lo que simplemente dijimos en una línea, no es algo sencillo de lograr, con lo cual vamos a comenzar a mencionar a los stakeholders más importantes como para saber que necesitan de la arquitectura.

Stakeholder	Necesidades
Cliente	Necesita saber sobre las solución va a contemplar con los requerimien-

Stakeholder	Necesidades
	tos del negocio que necesita el día a día para cumplir con sus objetivos. El tiempo y el costo asociado siempre es clave para este stakeholder.
Project Manager	Es el principal responsable de que el proyecto cumpla con lo pedido (alcance), en el tiempo que el negocio lo necesita y dentro del costo estipulado. La arquitectura es su principal herramienta de negociación frente al cliente, como así también para la organización de los equipos de trabajo, asignación de tareas y seguimiento. Los planes de staffing, compra de hardware y costos asociados son también importantes para el Project Manager.
Infraestructura	Necesitan entender como la aplicación será distribuida y donde debe funcionar para poder tener la infraestructura (Hardware, Red, Sistemas Operativos, Middleware, Bases de Datos) disponible para construir, testear y correr la aplicación en producción.
Analistas y Testers	No solo necesitan saber que es lo que el sistema proveerá al negocio, sino también la manera (solución=<=arquitectura) en la cual será resuelto. La arquitectura de un marco su trabajo.
Programadores y Diseñadores	Necesitan saber las responsabilidades del modulo que están construyendo, si alcance, sus dependencias y maneras de comunicación (protocolos). Los principios y estilos arquitectónicos que fijan las libertades y restricciones de constructibilidad.

2.3. Elementos y relaciones

Entendiendo lo importante que es la comunicación de la arquitectura y por que muchos roles tienen que entenderla, comenzaremos a explicar un modelo que permite resolver esta problemática. Pudimos ver que cada rol tiene determinadas necesidades y cada uno de ellos tienen diferentes experiencias y conocimientos, con lo cual es sencillo entender que es imposible comunicar una arquitectura de una única manera y en un único lenguaje. Teniendo clara esta base de la comunicación de arquitecturas, el siguiente modelo nos permite visualizar los elementos y sus relaciones que nos permitirán comunicar correctamente la arquitectura:



2.3.1. Stakeholders

Son todas las personas involucradas directa o indirectamente con el desarrollo de la aplicación en cuestión, no necesariamente tienen que ser parte del equipo, pueden ser usuarios, usuarios finales y/o usuarios indirectos.

2.3.2. Concerns (Necesidades)

Cuando hablamos de necesidades, estamos hablando de las incumbencias que tiene cada stakeholder con un sistema, dimos ejemplos en la sección [Stakeholders y Necesidades](#) sobre las necesidades de los stakeholders más comunes, y pudimos ver que cada una de ellas son muy variadas pasando de unos a otros lo cual requiere identificarlas, analizarlas y buscar la mejor alternativa de comunicación para satisfacerla. Podemos ver que los stakeholders pueden tener más de una necesidad y veremos que cada una de esas necesidades serán complacidas con los Viewpoints o Perspectivas.

2.3.3. Viewpoints (Perspectivas)

Un viewpoint determina los lenguajes (anotaciones, modelos, etc) que se usaran para describir una vista en particular intentando satisfacer las necesidades de los stakeholders. Generalmente en cada empresa existe un repositorio de viewpoints que permite la reutilización de modelos y muchos de estos vienen definidos en los Frameworks de Arquitectura que veremos más adelante en el apunte.

2.3.4. View (Vista)

Es la representación de una arquitectura con respecto a un viewpoint particular y se constituyen de View Packages y Modelos. La representación de una arquitectura se debe realizarse a través de varias vistas que permitan satisfacer las necesidades de los stakeholders. La relación entre Viewpoint y View puede explicarse con la analogía de una Clase y un Objeto, o una Tabla con un registro. Las vistas es algo específico del sistema a comunicar.

2.3.5. Models (Modelos)

Son los diagramas (elementos y sus relaciones) que utilizamos para comunicar una vista particular, podemos organizarla que mejor convenga para la comunicación de la vista. Es importante mantener trazabilidad y no repetición, y cada diagrama debe contener una clara descripción de los elementos, estereotipos y relaciones que la compone.

2.3.6. Architectural description

Es la documentación y/o comunicación de la arquitectura propiamente dicha, que está organizada en una serie de Vistas me mejor permitan describir el sistema.

3. Frameworks de Arquitectura de Software

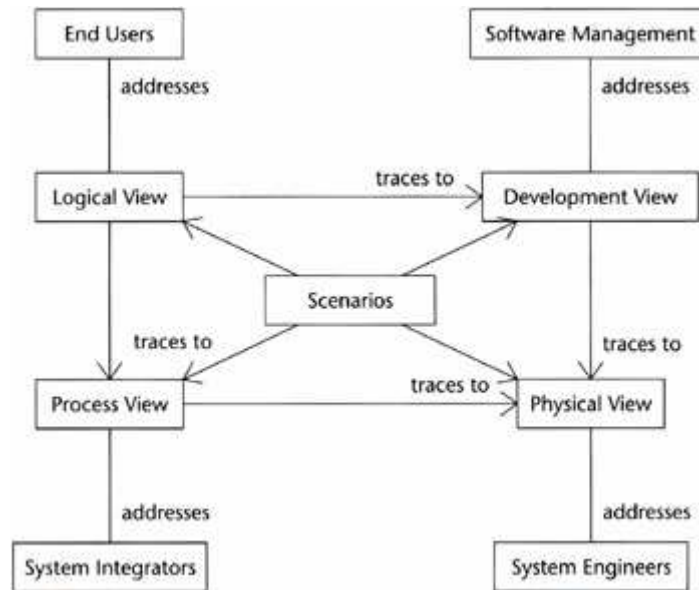
Los frameworks de arquitectura de software especifican las perspectivas (viewpoints) y relaciones entre estas necesarias para crear una arquitectura de software para sistemas específicos. No todos los Frameworks sirven para todos los tipos de software, para las aplicaciones enterprise y orientadas a objetos, vamos a desarrollar el "The 4+1 View Model of Architecture" creado por Kruchten en 1995 para Rational.

3.1. The 4+1 View Model of Architecture

El framework consiste en cuatro vistas (+ una):

- Lógica (logical)
- Proceso (process)
- Física (physical)
- Desarrollo (development)

Y también provee una quinta y redundante vista que permite juntar las diferentes vistas en escenarios. Cada una de estas vistas está dirigida a determinados stakeholders del sistema.



3.1.1. Logical View

Es un viewpoint que representa los requerimientos funcionales. Es independiente de plataforma, por lo general representan el concepto del dominio del problema, o sea los objetos del negocio. Esta vista debería mapear los requerimientos con las clases y sus relaciones. No solo al usuario final es dirigida este viewpoint, sino también a los desarrolladores y operadores.

3.1.2. Development View

Es un viewpoint que representa la organización de los subsistemas y cantidad layers, interfases entre estas y dependencias, permite visualizar y alocar diferentes equipos de trabajo para permitir la paralelización y entender el alcance y las interacciones necesarias. .

3.1.3. Process View

Es un viewpoint que representa el modelo de procesamiento del sistema, permite observar los diferentes unidades de procesamiento y los procesos que correrán en ella y sus diferentes interacciones. Tiene en cuenta los atributos no funcionales como performance y availability.

3.1.4. Physical View

Es un viewpoint que representa el mapeo entre el software y el hardware, como así también su distribución (build & deploy). Tiene en cuenta los atributos no funcionales como, availability, reliability, performance y scalability.

3.1.5. Scenarios View

Es un viewpoint integra los viewpoint (vistas) juntas en los casos de uso y escenarios de estos. Especificando los requerimiento funcionales.

4. Guía/Metodología de Comunicación

El objetivo de las guías es brindar un marco y orden para la comunicación de la arquitectura, teniendo en cuenta que comunicar, de que manera y como.

El orden propuesto es simplemente un gusto de los autores, puede variar.

1. Diagrama de Contexto y Overview Diagram
2. Identificar Stakeholders y Concerns
3. Definir los ViewPoint necesarios

4. Documentar y Comunicar las Vistas
5. Documentar y Comunicar los Escenarios
6. Documentar las decisiones (Constantemente)
7. Armado de SAD (Wiki, Documento de Texto, etc)

4.1. Diagrama de Contexto y Overview Diagram

No es la intención de este apunte explicar que es un Diagrama de Contexto, lo que si es relevante destacar la importancia de este, principalmente para entender los tipos de usuarios y sistemas con los que va a interactuar nuestro sistema, como así también dejar claro cual es el alcance y los límites del sistema, si este no existe, realmente recomiendo al arquitecto hacerlo, realmente sirve (confíen).

Por otro lado tenemos el diagrama de Solución (Overview Diagram), básicamente es un diagrama esquemático que representa las ideas preestablecidas y los módulos/componentes candidatos de un sistema o arquitectura. Provee un resumen de los elementos conceptuales y sus relaciones en una arquitectura que permiten observar la solución, o sea, cual es la solución que se va a construir para atacar la necesidad de negocio. Dicho diagrama, debería mostrar los siguientes elementos:

- Actores o Roles Principales
- Los módulos/componentes principales
- Los Nodos principales
- Repositorios de Datos
- Como fluye la información
- Las zonas de red

4.2. Identificar Stakeholders y Concerns

El objetivo aquí es identificar los stakeholders y priorizarlos de acuerdo a la importancia. Con ellos podremos entender sus necesidades, y de dichas necesidades podremos tener bien claro cuales de ellos tendremos que poner más énfasis en la comunicación. Esta es una sección que tiene que evolucionar más en las próximas versiones del apunte.

4.3. Definir los ViewPoint necesarios

De acuerdo a las necesidades de los stakeholders obtenidas del paso anterior se definen o seleccionan los ViewPoints que podrían ayudar a la comunicación. Es importante que por más que un stakeholder tenga mucho poder de decisión y necesitemos consensuar o discutir la arquitectura, no siempre necesita mucho detalle, con lo cual es importante categorizar las necesidades y stakeholder dentro de las siguientes categorías:

- Información detallada
- Algunos detalles
- Resumen
- Nada

No todas las organizaciones poseen un catálogo de Viewpoints disponibles para que el arquitecto pueda buscar y utilizar, con lo cual es necesario buscar en los Frameworks de Arquitectura disponibles.

A la hora de seleccionar los Viewpoints, no solo necesitamos identificar las necesidades de los stakeholders sino también es importante entender el lenguaje y la manera (canal) de cómo comunicar la arquitectura, hay que entender que un documento no es la única manera, tenemos muchos otros medios que nos beneficiarán y harán una recepción satisfactoria, tales como:

- Videos tipo Webcast
- Podcast
- Reuniones con pizarrón
- Maquetas
- Comunidades Virtuales como Secondlife
- etc

Como pueden ver en el área de comunicación, todo es posible, no nos cerremos busquemos la mejor la manera que nos permita comunicar exitosamente.

4.4. Documentar y Comunicar las Vistas

4.5. Documentar y Comunicar los Escenarios

4.6. Documentar las decisiones (Constantemente)

4.7. Armado de SAD (Wiki, Documento de Texto, etc)

-