Contenido

[Introducción 2](#_Toc397417901)

[Usos y audiencias para la documentación de la Arquitectura 2](#_Toc397417902)

[Notaciones de la Documentación de Arquitectura 4](#_Toc397417903)

[Vistas 5](#_Toc397417904)

[Módulo Vistas 6](#_Toc397417905)

[Resumen del módulo de vistas 6](#_Toc397417906)

[Vista de componentes y conectores (C&C) 8](#_Toc397417907)

[Notaciones para la vista C&C 12](#_Toc397417908)

[Asignación de vistas 13](#_Toc397417909)

[Calidad de vistas 15](#_Toc397417910)

[La elección de las vistas 16](#_Toc397417911)

[Combinando Vistas 17](#_Toc397417912)

[Construyendo el paquete de documentación 19](#_Toc397417913)

[La Documentación de una vista 19](#_Toc397417914)

[La documentación de la información después de las Vistas 21](#_Toc397417915)

[Documentación en línea, hipertexto y Wikis 23](#_Toc397417916)

[Seguir una estrategia de lanzamiento 24](#_Toc397417917)

[La documentación de los patrones 24](#_Toc397417918)

[Comportamiento documentado 25](#_Toc397417919)

[Documentación de Arquitectura y Atributos de Calidad 28](#_Toc397417920)

[Arquitecturas Documentar que cambian rápidamente que usted puede documentarlos 29](#_Toc397417921)

[La documentación de la arquitectura en un proyecto de desarrollo ágil 30](#_Toc397417922)

[Resumen 31](#_Toc397417923)

# Introducción

Incluso la mejor arquitectura, es perfectamente adecuada más para el trabajo, será esencialmente inútil si las personas que necesitan utilizarlo no saben lo que es; no puede entender lo suficientemente bien como para usar, construir, o modificarlo; o (peor de todo) malinterpretan y se aplican de forma incorrecta. Y todo el esfuerzo, el análisis, el trabajo duro, y el diseño y detalles por parte del equipo de arquitectura se habrá desperdiciado.

La creación de una arquitectura no es suficiente. Tiene que ser comunicado en una manera de dejar a las partes interesadas adecuadamente para hacer su trabajo. Si usted va a la molestia de crear una arquitectura fuerte, una que usted espera resistirá el paso del tiempo, entonces usted debe de describirlo con suficiente detalle, sin ambigüedad, y organizarlo para que otros puedan encontrar de forma rápida y actualización la información necesaria.

La documentación habla por el arquitecto. Se habla hoy al arquitecto, cuando el arquitecto debería estar haciendo otras cosas además de responder a un centenar de preguntas acerca de la arquitectura. Y habla por el arquitecto el día de mañana, cuando él o ella ha dejado el proyecto y ahora otra persona está a cargo de su evolución y mantenimiento.

La triste verdad es que la documentación de arquitectura hoy en día, si se hace en absoluto, se trata a menudo como algo secundario, algo que las personas hacen porque tienen que hacerlo. Tal vez un contrato así lo requiere. Tal vez un cliente lo demanda. O tal proceso estándar de una empresa lo requiere. De hecho, todas estas razones son legítimas. Pero ninguno de ellos es lo suficientemente convincente como para producir documentación de alta calidad. ¿Por qué el arquitecto gasta valioso tiempo y energía sólo para que un gerente pueda marcar un entregable?

Los mejores arquitectos producen buena documentación no porque es "necesario", sino porque ven que es esencial a la materia en un producto de alta. Ven a sus actores inmediatos como las personas más íntimamente involucrados en esta empresa: desarrolladores, implementadores, evaluadores y analistas.

Pero los arquitectos también ven la documentación como la entrega de valor a sí mismos. La documentación sirve como el recipiente para contener los resultados de las principales decisiones de diseño que se confirmen. Un sistema de documentación bien pensado puede hacer que el proceso de diseño vaya mucho más suavemente y de forma sistemática.

# Usos y audiencias para la documentación de la Arquitectura

La documentación de arquitectura debe servir para varios fines. Debe ser lo suficientemente **transparente y accesible** para ser entendido rápidamente por los nuevos empleados. Debe ser lo **suficientemente concreta** para servir como modelo para la construcción. Debe tener **suficiente información** para servir como una base para el análisis.

La documentación de arquitectura es **prescriptiva y descriptiva**. Para algunas personas, se prescribe lo que debe ser verdad, imponer las restricciones sobre las decisiones que aún no se ha hecho. Para otros públicos, describe lo que es verdadero, relatando las decisiones ya tomadas sobre el diseño de un sistema.

Para la mejor documentación de arquitectura, por ejemplo, el análisis de rendimiento puede ser muy diferente de la mejor documentación de arquitectura que desearíamos entregar a un implementador. Y ambas cosas serán diferentes de lo que ponemos en "Bienvenido a bordo" paquete de una nueva contratación o una sesión informativa que armamos para un ejecutivo. Al planificar y revisar la documentación, es necesario asegurar el apoyo a todas las necesidades pertinentes.

Podemos ver que muchos tipos diferentes de personas van a tener un interés personal en un documento de arquitectura. Esperan y esperan que el documento de arquitectura les ayude a hacer sus respectivos trabajos. La comprensión de sus usos de la documentación de arquitectura es esencial, ya que esos usos determinan la información importante a la captura.

Fundamentalmente, **la documentación arquitectura tiene tres usos**:

**1.** La documentación arquitectura sirve como un medio de educación. El uso educativo consiste en introducir a las personas al sistema. Las personas pueden ser nuevos miembros del equipo, los analistas externos, o incluso un nuevo arquitecto. En muchos casos, la "nueva" persona es el cliente al que usted está mostrando su solución por primera vez, una presentación que espera se traducirá en la financiación o el visto bueno de aprobación.

**2.** La documentación arquitectura sirve como vehículo principal para la comunicación entre las partes interesadas. El uso preciso de una arquitectura como vehículo de comunicación dependerá de lo que las partes interesadas hagan en la comunicación.

Tal vez uno de los más ávidos consumidores de documentación arquitectura no es otro que el arquitecto en el futuro del proyecto. El arquitecto futuro puede ser la misma persona o puede ser un reemplazo, pero en cualquier caso él o ella están garantizando tener un enorme interés en la documentación. Los nuevos arquitectos están interesados ​​en aprender cómo sus predecesores abordaron las cuestiones difíciles del sistema y por qué se tomaron ciertas decisiones. Incluso si el futuro arquitecto es la misma persona, él o ella utilizarán la documentación como un repositorio de pensamiento, un depósito de las decisiones de diseño muy numeroso e irremediablemente entrelazado, incapaz de ser reproducible en la memoria.

**3.** La documentación de arquitectura sirve como base para el análisis del sistema y la construcción. La arquitectura dice los implementadores que implementan. Cada módulo tiene interfaces que deben ser proporcionadas y utiliza interfaces de otros módulos. Esto no sólo proporciona instrucciones sobre las interfaces proporcionadas y usados, pero también determina con qué otros equipos el equipo de desarrollo deben comunicarse para el módulo.

Durante el desarrollo, una arquitectura puede ser muy compleja, con muchos temas que quedaron por resolver. La documentación puede servir como recipiente para el registro y la comunicación de estos temas que de otro modo podrían pasarse por alto.

Para aquellos interesados ​​en la capacidad del diseño para satisfacer los objetivos de calidad del sistema, la documentación de la arquitectura sirve de forraje para su evaluación. Debe contener la información necesaria para evaluar una variedad de atributos, como la seguridad, el rendimiento, la facilidad de uso, la disponibilidad y la modificabilidad.

Para los constructores de sistemas que utilizan herramientas automáticas de generación de código, la documentación puede incorporar los modelos utilizados para la generación. Estos modelos proporcionan una guía para aquellos que deseen comprender el comportamiento del módulo con más detalle que normalmente está documentado, pero con menos detalle que proporcionaría examinar el código.

# Notaciones de la Documentación de Arquitectura

Las Anotaciones para documentar las opiniones, difieren considerablemente en su grado de formalidad. En términos generales, hay tres categorías principales de la notación:

• **Notaciones informales**. Se representan con vistas (a menudo gráficamente) usando diagramas de uso general y herramientas de edición, y convenciones visuales elegidos para el sistema en cuestión. La semántica de la descripción se caracteriza por ser en lenguaje natural, y que no pueden ser analizados formalmente. En nuestra experiencia, la herramienta más común para notaciones informales es **PowerPoint**.

• **Notaciones semiformales**. Las vistas se expresan en una notación estándar que establece los elementos gráficos y las normas de construcción, pero no proporciona un tratamiento semántico completo del significado de esos elementos. El análisis elemental se puede aplicar para determinar si una descripción satisface propiedades sintácticas. **UML** es una notación semiformal en este sentido.

• **Notaciones formales**. Las vistas se describen en una anotación que tiene una precisión semántica (por lo general de base matemática). El análisis formal de las sintaxis y la semántica es posible. Hay una variedad de notaciones formales para la arquitectura de software disponible. Generalmente se conoce como descripción de la arquitectura idiomas (**ADL**), que suelen ofrecer tanto un vocabulario gráfico y una semántica subyacente para la representación de la arquitectura. En algunos casos, estas notaciones son especializadas para determinadas vistas arquitectónicas. En otros, permiten muchos puntos de vista, o incluso ofrecen la posibilidad de definir formalmente los nuevos puntos de vista. La utilidad de las **ADL** radica en su capacidad para apoyar la automatización a través de herramientas asociadas: la automatización para proporcionar un análisis útil de la arquitectura o ayudar en la generación de código. En la práctica, el uso de tales notaciones es raro.

Determinar qué forma de notación utilizar consiste en hacer varias concesiones. Típicamente, las **notaciones más formales** toman más tiempo y esfuerzo para crear y entender, pero devuelven este esfuerzo en reducción de la ambigüedad y más oportunidades para el análisis. Por el contrario, las **notaciones más informales** son más fáciles de crear, pero dan menos garantías.

Independientemente del nivel de formalidad, siempre recuerde que las diferentes notaciones son mejores (o peores) para expresar diferentes tipos de información. La formalidad de lado, sin diagrama de clases UML le ayudará a razonar sobre planificabilidad, ni un cuadro de secuencias le dirán mucho acerca de la probabilidad de que el sistema sea entregado a tiempo. Usted debe elegir sus notaciones y lenguajes de representación siempre teniendo en cuenta las cuestiones importantes que usted necesita capturar.

# Vistas

Tal vez el concepto más importante asociado con la documentación arquitectura de software es el de la vista. **Una arquitectura de software es una entidad compleja que no se puede describir en una manera simple de una sola dimensión**. **Una vista es una representación de un conjunto de elementos del sistema y las relaciones entre ellas-no todos los elementos del sistema, pero los de un tipo particular**. Por ejemplo, una vista en capas de un sistema mostraría elementos de tipo "capa", es decir, sería una muestra de la descomposición del sistema en capas-y las relaciones entre esas capas. Una vista en capas pura no sería, sin embargo, mostrar los servicios del sistema, o clientes y servidores, o modelo de datos, o cualquier otro tipo de elemento.

Por lo tanto, las vistas nos permiten dividir la entidad multidimensional en un número (esperamos) de representaciones interesantes y manejables del sistema. El concepto de puntos de vista nos da nuestro principio más fundamental de la documentación de la arquitectura:

La documentación de una arquitectura es una cuestión de la documentación de los puntos de vista relevantes, y luego añadir la documentación que se aplica a más de una vista.

Esta máxima da nuestro enfoque a la documentación de su nombre: Vistas y más allá.

¿Cuáles son los puntos de vista relevantes? Esto depende totalmente de sus metas. Como vimos anteriormente, la documentación de arquitectura puede servir para muchos propósitos: el enunciado de una misión para los implementadores, una base para el análisis, la especificación para la generación automática de código, el punto de partida para la comprensión del sistema y la recuperación de activos, o el modelo para la planificación de proyectos.

Diferentes puntos de vista también exponen diferentes atributos de calidad en diferentes grados. Por lo tanto, los atributos de calidad que son de mayor interés para usted y las otras partes interesadas en el desarrollo del sistema, afectarán la elección de qué vistas usa para el documento. Por ejemplo, una vista en capas le permitirá decidir acerca de la portabilidad de su sistema, una vista de despliegue le permitirá decidir sobre el rendimiento y la fiabilidad de su sistema, y así sucesivamente.

Diferentes vistas tienen diferentes objetivos y usos. Es por esto que no abogamos por una visión particular o una colección de puntos de vista. Los puntos de vista que debe documentar dependen de los usos que se usted espera para hacer de la documentación. Diferentes puntos de vista destacarán diferentes elementos del sistema y las relaciones. ¿Cuántos puntos de vista diferentes se puede usar para representar el resultado de una decisión de costo / beneficio?. Cada vista tiene un costo y un beneficio, y usted debe asegurarse de que los beneficios de mantener una visión particular son mayores que sus costos.

Las vistas pueden ser impulsadas por la necesidad de documentar un patrón particular en su diseño. Algunos patrones están compuestos por módulos, otros de los componentes y conectores, y otros tienen consideraciones de implementación. Vistas del módulo, vistas de componentes-y-conector (C & C), y puntos de vista de asignación son los mecanismos adecuados para la representación de estas consideraciones.

## Módulo Vistas

**Un módulo es una unidad de ejecución que proporciona un conjunto coherente de responsabilidades**. Un módulo puede tomar la forma de una clase, una colección de clases, una capa, un aspecto, o cualquier descomposición de la unidad de ejecución. Los módulos de las son un ejemplo de descomposición, son usos y capas. **Cada módulo tiene un conjunto de propiedades que se le asignan. Estas propiedades están destinadas a expresar la información importante asociada con el módulo, así como limitaciones en el módulo**. Algunas **propiedades** son las **responsabilidades**, **información**, **visibilidad** y el **historial de revisiones**. Las **relaciones** que los módulos tienen entre sí incluyen: una parte de, depende de y es una.

La manera en que el software de un sistema se descompone en unidades manejables sigue siendo una de las formas importantes de la estructura del sistema. Como mínimo, esto determina cómo se descompone el código fuente de un sistema en unidades, qué tipo de supuestos cada unidad puede hacer sobre los servicios prestados por otras unidades, y cómo esas unidades se agregan en conjuntos más grandes. También incluye estructuras de datos globales que impactan y son impactados por múltiples unidades. Las estructuras de módulos suelen determinar cómo los cambios en una parte de un sistema pueden afectar a otras partes y por lo tanto la capacidad de un sistema de apoyo a la modificabilidad, portabilidad y reutilización.

Es poco probable que la documentación de cualquier arquitectura de software puede ser completa sin al menos una vista del módulo.

### Resumen del módulo de vistas

|  |  |
| --- | --- |
| Elementos | módulos, que son implementaciones de unidades de software que proporcionan un conjunto coherente de responsabilidades |
| Relaciones | Es parte de: que define un todo / parte relación entre el submódulo. La parte, el módulo agregado o el conjunto. Depende de: que define una relación de dependencia entre dos módulos. Las vistas específicas de módulos elaboran lo que se entiende por dependencia. Es una: que define una relación de generalización/especificación entre un módulo y uno más específico (el hijo, un módulo más general, el padre) |
| Restricciones | Los puntos de vista diferentes de los módulos pueden imponer restricciones topológicas específicas, tales como limitaciones en la visibilidad entre los módulos. |
| Uso | Anteproyecto para la construcción del código.  Análisis de cambio de impacto.  Planificación de desarrollo incremental.  Análisis de los requisitos de trazabilidad.  Comunicar la funcionalidad de un sistema y la estructura de su base de código.  Soportar la definición de asignaciones de trabajo, calendarios de ejecución y la información presupuestaria.  Que muestra la estructura de información que el sistema necesita para administrar |

Las propiedades de los módulos que ayudan a orientar la aplicación o se introducen en el análisis deben registrarse como parte de la documentación de apoyo para una vista del módulo. La lista de propiedades puede variar, pero es probable que incluya lo siguiente:

• **Nombre**. El nombre de un módulo es, por supuesto, el principal medio para referirse al mismo. El nombre de un módulo menudo sugiere algo acerca de su papel en el sistema. Además, el nombre de un módulo puede reflejar su posición en una jerarquía de descomposición; el nombre ABC, por ejemplo, se refiere a un módulo de C que es un submódulo de un módulo B, en sí un submódulo de A.

• **Responsabilidades**. La propiedad responsabilidad de un módulo es una manera general de identificar su papel en el sistema y establece una identidad más allá del nombre. Mientras que el nombre de un módulo puede sugerir su papel, una declaración de responsabilidad se establece con mucha más certeza. Las responsabilidades deben ser descritos con suficiente detalle para dejar claro al lector de lo que hace cada módulo.

• **Visibilidad de interfaz** (s). Cuando un módulo tiene submódulos, algunas interfaces de los submódulos son públicos y algunos pueden ser privadas; es decir, las interfaces se utilizan sólo por los submódulos de la envolvente módulo padre. Estas interfaces privadas no son visibles fuera de ese contexto.

• **Información de Implementación**. Los módulos son unidades de ejecución. Por tanto, es útil registrar la información relacionada con su aplicación desde el punto de vista de la gestión de su desarrollo y la construcción del sistema que los contiene. Esto podría incluir lo siguiente:

* **Asignación de unidades de código fuente**. Esto identifica los archivos que constituyen la aplicación de un módulo. Por ejemplo, una cuenta de módulo, si se implementa en Java, puede tener varios archivos que constituyen su aplicación: IAccount.java (una interfaz), AccountImpl.java (la implementación de la funcionalidad de la cuenta), AccountBean.java (una clase para mantener el estado en la memoria), AccountOrmMapping.xml (un archivo que define la correlación entre AccountBean y un mapeo objeto-relacional tabla de base de datos), y tal vez incluso un AccountTest.java de prueba unitaria.
* **Información de la prueba**. El plan del módulo de prueba, casos de prueba, andamios prueba, y los datos de prueba son importantes para documentar. Esta información puede ser simplemente un indicador a la ubicación de estos artefactos.
* **Gestión de la información**. Un administrador puede necesitar información sobre el horario y el presupuesto previsto del módulo. Esta información puede ser simplemente un indicador a la ubicación de estos artefactos.
* **Las limitaciones de implementación**. En muchos casos, el arquitecto tendrá una estrategia de implementación en cuenta para un módulo pero debe saber de las limitaciones que la aplicación puede tener.
* **Historial de revisiones**. Conocer la historia de un módulo que incluye autores y cambios concretos puede ayudar al realizar las actividades de mantenimiento.

Debido a que los módulos particionan el sistema, debería ser posible determinar cómo los requisitos funcionales de un sistema están soportados por las responsabilidades del módulo. Las vistas del módulo muestran las dependencias entre módulos o capas (que son grupos de módulos que tienen un patrón específico de uso permitido) proporcionan una buena base para el análisis del cambio-impacto. Los módulos son típicamente modificados como resultado de informes de incidencias o peticiones de cambio. Los análisis de impacto requieren un cierto grado de integridad de diseño y la integridad de la descripción del módulo. En particular, la información de dependencia tiene que estar disponible y correcta para ser capaz de crear resultados útiles.

Una vista del módulo puede ser utilizada para explicar la funcionalidad del sistema a alguien no familiarizado con él. Los diversos niveles de granularidad del módulo de descomposición proporcionan una presentación de arriba hacia abajo de las responsabilidades del sistema y por lo tanto puede guiar el proceso de aprendizaje. Para un sistema cuya ejecución ya está en su lugar, las vistas del módulo, si se mantiene hasta la fecha, son útiles, ya que explican la estructura de la base de código para un nuevo desarrollador en el equipo. Por lo tanto, hasta la fecha, los módulos de visitas pueden simplificar y regularizar el mantenimiento del sistema.

Por otro lado, es difícil de usar las vistas del módulo para hacer inferencias sobre el comportamiento en tiempo de ejecución, porque estos puntos de vista son sólo una partición estática de las funciones del software. Por lo tanto, una visión del módulo no se suele utilizar para el análisis de rendimiento, fiabilidad, y muchas otras cualidades de tiempo de ejecución. Para ello, confiamos en vistas de componentes-y-conectores y asignación.

Las vistas del módulo son comúnmente asignados a las vistas de componentes-y-conectores. Las unidades de ejecución que se muestran en las vistas de módulos tienen un mapeo de los componentes que se ejecutan en tiempo de ejecución. A veces, el mapeado es bastante sencillo, incluso uno-a-uno para aplicaciones pequeñas y sencillas. Más a menudo, un único módulo será replicado como parte de muchos componentes en tiempo de ejecución, y un componente dado se puede asignar a varios módulos. Las vistas del módulo también proporcionan los elementos de software que se asignan a los diversos elementos que no son de software, del entorno del sistema en los distintos puntos de vista de asignación.

## Vista de componentes y conectores (C&C)

Las vistas de componentes-y-conectores **muestran los elementos que tienen alguna participación en tiempo de ejecución, tales como procesos, objetos, clientes, servidores y almacenamiento de datos**. Estos elementos son los llamados **componentes**. Además, las vistas componente-y-conectores incluyen como elementos de los caminos de interacción, tales como enlaces de comunicación y protocolos, los flujos de información, y el acceso al almacenamiento compartido. Estas interacciones se representan como conectores en vistas de C & C. Las vistas de ejemplo de C & C son la arquitectura orientada a servicios (SOA), cliente-servidor, o comunicar puntos de vista del proceso.

**Los componentes tienen interfaces llamados** **puertos**. **Un puerto define un punto de potencial de interacción de un componente con su entorno**. Un puerto generalmente tiene un tipo explícito, que define el tipo de comportamiento que puede tener lugar en ese punto de interacción. Un componente puede tener muchos puertos del mismo tipo, cada uno formando un canal de entrada o de salida diferente en tiempo de ejecución. En este sentido difieren de las interfaces de puertos de módulos, cuyas interfaces no se replican. **La notación de un puerto puede ser con un número o una serie de números que indican la replicación; por ejemplo, "1..4" puede significar que una interfaz podría ser replicado hasta cuatro veces**. Los puertos de un componente se deben documentar de forma explícita, mostrándoles en el diagrama y definirlos en la documentación de apoyo de la figura.

Un componente en una vista en C y C puede representar un subsistema complejo, que en sí mismo puede ser descrito como una sub arquitectura C & C. Esta sub arquitectura se puede representar gráficamente in situ cuando la subestructura no es demasiado compleja, mostrando como se va anidado dentro del componente que se refina. A menudo, sin embargo, está documentado por separado. La Sub arquitectura de un componente puede emplear un patrón diferente de aquel en el que aparece el componente.

Los **conectores** son otro tipo de elemento en una vista de C & C. Ejemplos sencillos de **conectores son la invocación de servicio**; colas de mensajes asíncronos; multicast evento apoyando publicación-suscripción interacciones; y tuberías que representan flujos de datos asíncronos, el orden de preservación. Los conectores a menudo representan una forma mucho más complejas de interacción, como un canal de comunicación orientado a transacciones entre un servidor de base de datos y un cliente, o un bus de servicios empresariales que mide las interacciones entre las colecciones de los usuarios de servicios y proveedores.

Los conectores tienen **roles**, **que son sus interfaces, que definen las formas en que el conector puede ser utilizado por los componentes para llevar a cabo la interacción**. Por ejemplo, un conector de **cliente-servidor** podría invocar los servicios y proporcionar servicios-roles. Una tubería podría tener papeles escritor y el lector. Al igual que los puertos de componentes, los roles de conectores son diferentes de las interfaces de módulos en que se pueden replicar, indicando el número de componentes que pueden estar involucrados en su interacción. Un conector de **publicación-suscripción** puede tener muchas instancias de las funciones de editor y suscriptor.

Como componentes, los conectores complejos pueden a su vez ser descompuestos en colecciones de componentes y conectores que describen la subestructura arquitectónica de los conectores. Los conectores no necesitan ser binario. Es decir, que no necesitan tener exactamente dos roles. Por ejemplo, un conector de publicación-suscripción puede tener un número arbitrario de funciones publicador y suscriptor. Incluso si el conector se implementa en última instancia, utilizando conectores binarios, como una llamada de procedimiento, puede ser útil adoptar representaciones conectores n-arias en una vista en C y C. Los conectores incorporan un protocolo de interacción. Cuando dos o más componentes interactúan, tienen que obedecer las convenciones sobre el orden de las interacciones, locus de control, y la manipulación de las condiciones de error y los tiempos de espera. El protocolo de interacción debe ser documentado.

La relación primaria dentro de una vista C & C es el apego. Los adjuntos indican que los conectores están unidos a qué componentes, definiendo de este modo un sistema como un gráfico de los componentes y conectores. En concreto, un archivo adjunto se denota mediante la asociación (fijación) de puerto de un componente para el papel de un conector. Un accesorio válido es uno en el que los puertos y funciones son compatibles entre sí, bajo las restricciones semánticas definidas por la vista. La compatibilidad con frecuencia se define en términos de tipo de información y protocolo. Por ejemplo, en una arquitectura llamada-retorno, usted debe comprobar para asegurarse de que todos los "llamados" puertos se conectan a algún conector llamada-retorno. En un nivel más profundo de semántica, usted debe comprobar para asegurarse de que el protocolo de un puerto es consistente con el comportamiento esperado por la función a la que está unido.

Un elemento (componente o conector) de una vista C & C tendrá diferentes propiedades asociadas. Cada elemento debe tener un nombre y tipo. Las propiedades adicionales dependen del tipo de componente o conector. Definir los valores de las unidades que apoyan los análisis previstos para la visión particular de C & C. Por ejemplo, si la vista se utilizará para análisis de rendimiento, latencias, las capacidades de la cola, y las prioridades de rosca puede ser necesario. Los siguientes son ejemplos de algunas propiedades típicas y sus usos:

• Confiabilidad. ¿Cuál es la probabilidad de fallo de un componente o conector dado? Esta propiedad podría ser utilizada para ayudar a determinar la disponibilidad general del sistema.

• Rendimiento. ¿Qué tipos de tiempo de respuesta del componente proporcionará bajo qué cargas? ¿Qué tipo de ancho de banda, latencia, jitter, el volumen de transacciones, o el rendimiento se puede esperar de un conector dado? Esta propiedad se puede utilizar con otros para determinar las propiedades de todo el sistema, tales como los tiempos de respuesta, rendimiento y necesidades de tamponamiento.

• Recursos necesarios. ¿Cuáles son las necesidades de procesamiento y almacenamiento de un componente o un conector? Esta propiedad se puede utilizar para determinar si una configuración de hardware propuesto será adecuada.

• Funcionalidad. ¿Qué funciones realiza un elemento? Esta propiedad se puede utilizar para razonar acerca de cómputo global realizado por un sistema.

• Seguridad. ¿Tiene un componente o un conector de hacer cumplir o proporcionan características de seguridad, como la encriptación, pistas de auditoría, o la autenticación? Esta propiedad se puede utilizar para determinar las vulnerabilidades de seguridad del sistema.

• Concurrencia. ¿Este componente ejecuta como un proceso independiente o hilo? Esta propiedad puede ayudar a analizar o simular el rendimiento de los componentes concurrentes e identificar posibles puntos muertos.

• Modificabilidad. ¿Apoya la estructura de mensajería una estructura para atender a la evolución de los intercambios de datos? ¿Pueden los componentes pueden adaptar para procesar esos mensajes nuevos? Esta propiedad se puede definir para ampliar la funcionalidad de un componente.

• Nivel. Para una topología de niveles, en qué nivel tiene el componente reside en? Esta propiedad ayuda a definir los procedimientos de construcción y despliegue, así como requisitos de la plataforma para cada nivel.

Vistas de C & C se utilizan comúnmente para mostrar a los desarrolladores y otros interesados ​​cómo funciona un sistema puede "animar" o rastrear a través de una vista de C & C, que muestra un hilo de actividad de extremo a extremo. Vistas de C & C también se utilizan para razonar acerca de los atributos de calidad del sistema en tiempo de ejecución, como el rendimiento y la disponibilidad. En particular, una visión bien documentada permite a los arquitectos para predecir las propiedades generales del sistema como la latencia o la fiabilidad, estimaciones dadas o mediciones de las propiedades de los elementos individuales y sus interacciones.

|  |  |
| --- | --- |
| Elementos | Componentes: Son las principales unidades de procesamiento y almacenes de datos. Un componente tiene un conjunto de puertos a través del cual interactúa con otros componentes.  Conectores: caminos de interacción entre los componentes. Conectores tienen un conjunto de funciones (interfaces) que indican cómo los componentes pueden utilizar un conector en las interacciones. |
| Relaciones | Adjuntos: Puertos de componentes que están asociados con funciones de conector para producir un gráfico de componentes y conectores.  Delegación de interfaz: En algunas situaciones los puertos de componentes están asociados con uno o más puertos en un sub arquitectura "interna". El caso es similar para los funciones de un conector |
| Restricciones | Los componentes sólo se pueden conectar a los conectores, no directamente a otros componentes.  Los conectores sólo se pueden unir a los componentes, no directamente a otros conectores.  Los archivos adjuntos sólo pueden hacerse entre los puertos y las funciones compatibles.  La interfaz de delegación sólo puede ser definida entre dos puertos compatibles (o dos funciones compatibles.)  Los conectores no pueden aparecer en la insolación; un conector debe estar conectado a un componente. |
| Uso | Mostrar cómo funciona el sistema.  Guía de desarrollo de la estructura y comportamiento especificando los elementos en tiempo de ejecución.  Ayuda sobre los atributos de calidad del sistema en tiempo de ejecución, como el rendimiento y la disponibilidad |

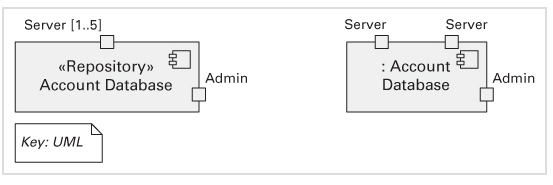
Resume los elementos, relaciones y propiedades que pueden aparecer en las vistas de C & C. Esta tabla es seguida por una discusión más detallada de estos conceptos, junto con las directrices relativas a su documentación.

## Notaciones para la vista C&C

Como siempre, los dibujos de caja y de línea están disponibles para representar las opiniones de C & C. Aunque las notaciones informales son limitadas en la semántica que puede ser transportado, siguiendo algunas pautas simples pueden prestar rigor y profundidad a las descripciones. La directriz principal es simple: asignar a cada tipo de componente y cada conector una forma visual independiente (símbolo), y la lista de cada uno de los tipos en una clave.

Los **componentes UML** son un buen partido de semántica de los componentes de C & C, ya que permiten la documentación intuitiva de información importante, como las interfaces, propiedades y descripciones de comportamiento. Los componentes UML también distinguen entre tipos de componentes y las instancias de componentes, que es útil en la definición específica de vista de los tipos de componentes.

Los **puertos UML** son un buen partido de semántica a los puertos de C & C. Un puerto de UML se puede adornar con una multiplicidad, como se muestra en la parte izquierda de la figura 18.1, aunque esto suele ser sólo hace en tipos de componentes. El número de puertos en instancias de componentes, como se muestra en la parte derecha de la Figura 18.1, esta típicamente unido a un número específico. Los componentes se crean dinámicamente y gestionar un conjunto de puertos deberían conservar un descriptor de multiplicidad de descripciones de instancia.

  
Una representación UML de los puertos de un tipo de componente C & C (izquierda) y la instancia de componente (a la derecha). El tipo de componente de base de datos cuenta con dos tipos de puertos, servidores y administración (señalado por las cajas en el borde del componente). El puerto del servidor se define con una multiplicidad, lo que significa que varias instancias del puerto se permiten en cualquier instancia de componente correspondiente.

Mientras los conectores C & C son semánticamente ricos como componentes de C & C, pero lo mismo no puede decirse de los conectores de UML. Los conectores UML no pueden tener la subestructura, atributos, o descripciones de comportamiento. Esto hace que la elección de la forma de representar C & C conectores sea más difícil, como conectores UML no siempre son lo suficientemente ricos.

Usted debe representar un "simple" conector C & C usando un conector de una línea de UML. Muchos conectores C & C de uso común han sido bien conocido, la semántica y las implementaciones independientes de las aplicaciones, tales como llamadas a funciones o datos de operaciones de lectura. Si la única información que necesita para suministrar es el tipo de conector, a continuación, un conector UML es adecuada. Call-conectores de retorno pueden ser representados por un conector del conjunto de UML, que une interfaz requerida de un componente (socket) a la interfaz proporcionada el otro componente (paleta). Puede utilizar un estereotipo para indicar el tipo de conector. Si todos los conectores en una presentación primaria son del mismo tipo, usted puede observar esto una vez en un comentario más que explícitamente en cada conector para reducir el desorden visual. Adjunto se muestra mediante la conexión de los extremos del conector a los puertos de componentes. Papeles conectores no pueden ser representados de manera explícita con un conector UML porque el elemento conector UML no permite la inclusión de interfaces de (a diferencia del puerto de UML, que sí permite interfaces). La mejor aproximación es etiquetar los extremos del conector y utilizar estas etiquetas para identificar las descripciones de funciones que deben ser documentados en otros lugares.

Usted debe representar una "rica" conector C & C usando un componente UML, o anotando un conector de la línea UML con una etiqueta u otra documentación auxiliar que se explica el significado del conector complejo.

## Vistas de asignación

Vistas de asignación describen el mapeo de unidades de software para elementos de un entorno en el que el software se desarrolla o en el que se ejecuta. El ambiente podría ser el hardware, el entorno operativo en que se ejecuta el software, los sistemas de archivos de apoyo al desarrollo o despliegue o la organización (s) de desarrollo.

La Tabla 18.3 resume las características de vistas de asignación. Vistas Asignación consisten en elementos de software y elementos ambientales. Ejemplos de elementos del medio ambiente son un procesador, una granja de disco, un archivo o carpeta, o un grupo de desarrolladores. Los elementos de software provienen de un módulo o C & C Vista.

|  |  |
| --- | --- |
| Elementos | Elementos de software. Un elemento de software tiene propiedades que se requieren el medio ambiente.  Elemento para el Medio Ambiente. Un elemento del medio ambiente tiene propiedades que se proporcionan con el software. |
| Relaciones | Distribuidos elemento de software estrictamente asignada (asignado a) un elemento del medio ambiente. Propiedades comieron depende del punto de vista particular. |
| Restricciones | Varía según el punto de vista |
| Uso | Para razonar sobre el rendimiento, la disponibilidad, la seguridad, y la seguridad.  Para razonar sobre el desarrollo distribuido y asignación de los trabajos a los equipos.  Para razonar sobre el acceso simultáneo a las versiones de software.  Para razonar sobre los forma y los mecanismos de instalación del sistema. |

La relación en una vista de la asignación se destina a. Solemos hablar de puntos de vista de asignación en términos de una asignación de elementos de software a elementos del medio ambiente, aunque el mapeo inverso también puede ser relevante e interesante. Un único elemento de software se puede asignar a varios elementos del medio ambiente, y varios elementos de software se puede asignar a un solo elemento del medio ambiente. Si estas asignaciones cambian con el tiempo, ya sea durante el desarrollo o ejecución del sistema, a continuación, la arquitectura se dice que es dinámica con respecto a que la asignación. Por ejemplo, los procesos pueden migrar de un procesador o máquina virtual a otra. Del mismo modo módulos pueden migrar de un equipo de desarrollo a otro.

Elementos de software y elementos ambientales tienen propiedades en las vistas de asignación. El objetivo habitual de asignación es una vista para comparar las propiedades requeridas por el elemento de software con las propiedades proporcionadas por los elementos ambientales para determinar si la asignación será exitosa o no. Por ejemplo, para asegurar el tiempo de respuesta requerido de un componente, que tiene que ejecutar en (ser asignado a) un procesador que proporciona potencia de procesamiento suficientemente rápido. Para otro ejemplo, una plataforma de computación podría no permitir que una tarea de utilizar más de 10 kilobytes de memoria virtual. Un modelo de ejecución del elemento de software en cuestión se puede utilizar para determinar el uso de la memoria virtual deseado. Del mismo modo, si está migrando de un módulo de un equipo a otro, es posible que desee asegurarse de que el nuevo equipo tiene las habilidades adecuadas y los conocimientos de fondo.

Vistas de asignación pueden representar vistas estáticas o dinámicas. Una vista estática representa una asignación fija de recursos en un entorno. Una visión dinámica representa las condiciones y los factores desencadenantes de los que la asignación de recursos de acuerdo a los cambios de carga. Algunos sistemas de reclutar y utilizar nuevos recursos a medida que aumenta la carga. Un ejemplo es un sistema de balanceo de carga en el que los nuevos procesos o hilos se crean en otra máquina. En este punto de vista, las condiciones en que han de estar documentados los cambios de asignación, la asignación de software runtime, y el mecanismo de asignación dinámica. (Recuerde del capítulo 1 que una de las estructuras de asignación es la estructura de asignación de trabajo, que asigna los módulos a los equipos de desarrollo. Esa relación, también, se puede asignar de forma dinámica, en función de "carga", en este caso, la carga en los equipos de desarrollo.)

## Calidad de vistas

Módulo, C & C, y vistas de asignación son todos los puntos de vista estructurales: Se muestran principalmente las estructuras que el arquitecto ha diseñado en la arquitectura para satisfacer los requisitos de atributos funcionales y de calidad.

Estos puntos de vista son excelentes para guiar y limitar los desarrolladores aguas abajo, cuyo trabajo principal es implementar esas estructuras. Sin embargo, en los sistemas en los que ciertos atributos de calidad (o, para el caso, cualquier otro tipo de preocupaciones de los interesados) son particularmente importantes y penetrante, vistas estructurales puede no ser la mejor manera de presentar la solución arquitectónica a esas necesidades. La razón es que la solución se puede propagar a través de múltiples estructuras que son inconvenientes para combinar (por ejemplo, porque los tipos de elementos que se muestran en cada uno son diferentes).

Otro tipo de vista, lo que llamamos una vista de la calidad, se puede adaptar para que los interesados ​​específicos o para responder a las preocupaciones específicas. Estos puntos de vista de calidad se forman mediante la extracción de las piezas pertinentes de vista estructural y empaquetarlos juntos. He aquí cinco ejemplos:

• **Una vista de seguridad puede mostrar todas las medidas arquitectónicas adoptadas para garantizar la seguridad**. Sería mostrar los componentes que tienen alguna función de seguridad o la responsabilidad, como estos componentes se comunican, los repositorios de datos para la información de seguridad y los depósitos que son de interés de la seguridad. Información de contexto de la vista mostraría otras medidas de seguridad (por ejemplo, seguridad física) en el entorno del sistema. La parte el comportamiento de una vista de la seguridad sería mostrar el funcionamiento de los protocolos de seguridad y dónde y cómo los seres humanos interactúan con los elementos de seguridad. También sería captar cómo el sistema podría responder a las amenazas y vulnerabilidades específicas.

• **La vista de las comunicaciones** podría ser especialmente útil para sistemas que están globalmente dispersos y heterogéneos. Este punto de vista mostraría todos los canales-componente a componente, los diversos canales de la red, los valores de los parámetros de calidad de servicio y áreas de concurrencia. Este punto de vista se puede utilizar para analizar ciertas clases de rendimiento y la fiabilidad (tales como detección de estancamiento o condición de carrera). La parte el comportamiento de este punto de vista podría mostrar (por ejemplo) cómo se asigna dinámicamente el ancho de banda de red.

• **Una excepción o vista de control** de errores podrían ayudar a iluminar y llamar la atención a error mecanismos de información y resolución. Tal punto de vista sería mostrar cómo los componentes de detectar, informar y resolver las fallas o errores. Sería de gran ayuda a identificar las fuentes de errores y acciones correctivas apropiadas para cada uno. Análisis de causa raíz en esos casos podría facilitarse mediante un punto de vista.

• **Una vista de la fiabilidad** sería aquella en la que se modelan los mecanismos de confiabilidad tales como la replicación y conmutación. También representaría problemas de tiempo y la integridad de la transacción.

• **Una vista del rendimiento** incluiría aquellos aspectos de la arquitectura útiles para inferir el rendimiento del sistema. Este punto de vista podría mostrar modelos de tráfico de red, latencias máximas para las operaciones, y así sucesivamente.

# La elección de las vistas

Documentar las decisiones durante el proceso de diseño (algo muy recomendable) produce vistas, que son el corazón de un documento de la arquitectura. Es más probable que estas opiniones son bocetos más que productos terminados listos para su publicación; esto le dará la libertad de copia de seguridad y repensar las decisiones de diseño que resultan ser problemáticos sin tener tiempo perdido en una amplia difusión y pulido cosmético. Están documentados puramente como su propia ayuda para la memoria.

En el momento en que esté listo para liberar un documento de la arquitectura, es muy probable que tenga una colección bastante bien trabajado fuera de puntos de vista de la arquitectura. En algún momento tendrá que decidir cuál tomar hasta su finalización, con el nivel de detalle, y que se debe incluir en un comunicado dado. También tendrá que decidir qué vistas se pueden combinar de manera útil con otros, a fin de reducir el número total de puntos de vista en el documento y revelar las relaciones importantes entre los puntos de vista.

Puede determinar qué puntos de vista se requiere, cuando la creación de ellos, y la cantidad de detalles a incluir si usted sabe lo siguiente:

• Lo que la gente y con qué habilidades, están disponibles   
• ¿Qué normas hay que cumplir?  
• ¿Cuál es el presupuesto en la mano?  
• Lo que el programa es   
• Cuáles son las necesidades de información de los grupos de interés importantes  
• Cuáles son los requisitos de atributos de calidad de conducción son   
• Lo que el tamaño del sistema es

Como mínimo, espera tener por lo menos un módulo de vista, al menos una vista C & C, y para sistemas más grandes, al menos una vista de asignación en el documento de la arquitectura. Más allá de eso regla básica, sin embargo, hay un método de tres pasos para la elección de los puntos de vista:

• **Paso 1: Construir un cuadro de actores / vista. Enumerar los partes interesadas para la documentación de la arquitectura de software de su proyecto por las filas**. Sea lo más completa que pueda. Para los columnas, enumerar los puntos de vista que se aplican a su sistema. (Utilice los estructuras discutidas en el Capítulo 1, los puntos de vista tratados en este capítulo, y los puntos de vista que su trabajo de diseño en el ADD ha sugerido como una lista inicial de candidatos.) Algunos puntos de vista (tales como la descomposición, utiliza y asignación de trabajo) se aplican a todos los sistemas, mientras que (vistas diferentes de C & C, la vista en capas) que otros sólo se aplican a algunos sistemas. Para los columnas, asegúrese de incluir los puntos de vista o visión bocetos que ya tiene como resultado de su trabajo de diseño hasta ahora.

Una vez que tenga las filas y las columnas definidas, llene en cada celda para describir la cantidad de información el interesado requiere de la vista: ninguno, sólo información general, detalle moderado o alto detalle. La lista de candidatos de la vista de entrar en el paso 2 se compone actualmente de esos puntos de vista para la que alguna de las partes interesadas tiene un interés personal.

• **Paso 2 Combine visitas.** La lista de candidatos de la vista desde el paso 1 es probable que produzca un impracticablemente gran número de visitas. Este paso aventar la lista a un tamaño manejable. Busque vistas marginales en la mesa: los que requieren sólo una visión general, o que servir muy pocos interesados​​. Combine cada vista marginal con otro punto de vista que tiene un electorado más fuerte.

• **Paso 3: Priorizar y escenario**. Después del paso 2 usted debe tener el conjunto mínimo de puntos de vista necesarios para servir a su comunidad de interesados​​. En este punto, usted necesita decidir qué hacer primero. Lo que se hace primero depende de su proyecto, pero aquí hay algunas cosas a tener en cuenta:

• La vista de descomposición (una de las vistas del módulo) es un punto de vista particularmente útil para liberar temprano. De alto nivel (es decir, ancho y poco profundo) descomposiciones son a menudo fáciles de diseñar, y con esta información el director del proyecto puede comenzar a los equipos de desarrollo de personal, poner la formación en el lugar, determinar qué partes de externalizar, y empezar a producir presupuestos y horarios.

• Tenga en cuenta que usted no tiene que satisfacer todas las necesidades de información de todos los interesados​en la mayor medida. Proporcionar el 80 por ciento de la información va un largo camino, y esto podría ser lo suficientemente bueno como para que los interesados ​​puedan hacer su trabajo. Consulte con el interesado para ver si un subconjunto de la información sería suficiente. Por lo general prefieren un producto que es entregado a tiempo y dentro del presupuesto sobre conseguir la documentación perfecta.

• Usted no tiene que completar una vista antes de iniciar otra. Las personas pueden progresar con información a nivel de información general, por lo que un enfoque en amplitud es a menudo la mejor.

# Combinando Vistas

El principio básico de la documentación de una arquitectura como un conjunto de puntos de vista distintos trae una ventaja de divide y vencerás a la tarea de documentación, pero si las vistas eran irrevocablemente diferente, sin asociación con otros, nadie sería capaz de entender el sistema como un todo.

Debido a que todos los puntos de vista en una arquitectura son parte de esa misma arquitectura y existen para lograr un propósito común, muchos de ellos tienen fuertes asociaciones entre sí. Gestión del modo de estructuras arquitectónicas asociadas es una parte importante del trabajo del arquitecto, independiente de si existe alguna documentación de esas estructuras.

A veces la manera más conveniente para mostrar una fuerte asociación entre dos puntos de vista es colapsar en una sola vista combinada, según lo dictado por el paso 2 del método de tres pasos que acabamos de presentar para elegir los puntos de vista. Una vista combinada es una vista que contiene elementos y relaciones que surgen a partir de dos o más puntos de vista. Visitas combinadas pueden ser muy útiles, siempre y cuando no se trate de los sobrecargue con demasiadas asignaciones.

La forma más fácil de combinar visitas es crear una plantilla que combina la información que de otro modo habría sido en dos vistas separadas. Esto funciona bien si el acoplamiento entre los dos puntos de vista es escaso; es decir, hay una estrecha relación entre los elementos en una vista y elementos en la otra vista. Si ese es el caso, la estructura descrita por la vista combinada será más fácil de entender que los dos puntos de vista se ven por separado. Para un ejemplo, ver la superposición de la descomposición y utiliza bocetos mostrados en la figura 18.2. En una plantilla, los elementos y los relaciones mantienen los tipos como se define en sus puntos de vista constituyen.

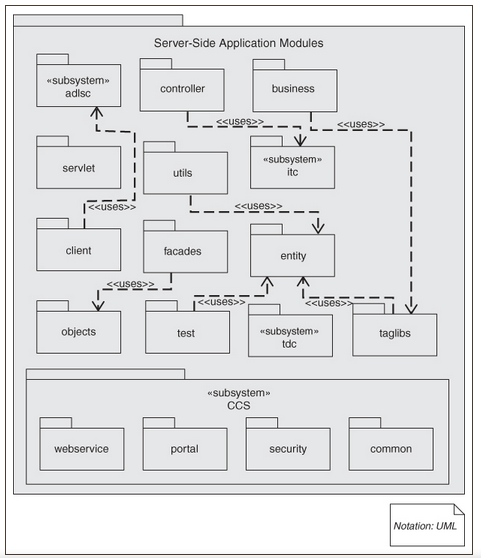


Figura 18.2. Una vista descomposición sobrepuesto con "utiliza" la información, para crear una descomposición / utiliza superposición.

Los puntos de vista siguientes menudo se combinan de forma natural:

• Diversas vistas de C & C. Debido a que C & C Vistas todas las relaciones espectáculo de tiempo de ejecución entre componentes y conectores de diferentes tipos, tienden a combinar bien. Los diferentes puntos de vista (por separado) C & C tienden a mostrar diferentes partes del sistema, o tienden a mostrar refinamientos de descomposición de componentes en otros puntos de vista. El resultado suele ser un conjunto de puntos de vista que se pueden combinar fácilmente.

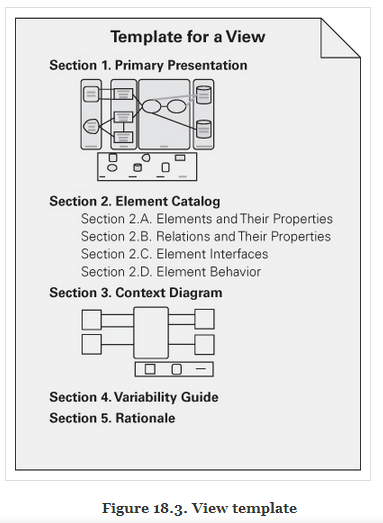
• La vista de implementación, ya sea con SOA o comunicantes-procesos vistas. Una vista SOA muestra los servicios, y una vista comunicantes-procesos muestra procesos. En ambos casos, estos son componentes que se implementan en los procesadores. Así, hay una fuerte asociación entre los elementos en estos puntos de vista.

• Vista de descomposición y cualquiera de asignación de trabajo, implementación, usos, o puntos de vista en capas. Los módulos descompuestos forman las unidades de trabajo, desarrollo y usos. Además, estos módulos pueblan capas.

# Construyendo el paquete de documentación

Recuerda el principio de la documentación de la arquitectura, con la que comenzamos este capítulo. Este principio nos dice que nuestra tarea consiste en documentar las opiniones pertinentes y para documentar la información que se aplica a más de una vista.

## La Documentación de una vista



La documentación para una vista se puede colocar en una organización estándar que consta de las siguientes partes:

• **Sección 1**: La Presentación de primaria. La presentación principal muestra los elementos y las relaciones de la vista. La presentación principal debe contener la información que desea transmitir sobre el sistema-en el vocabulario de esta opinión. Sin duda, debe incluir los elementos y las relaciones primarias, pero en algunas circunstancias podría no incluir a todos ellos. Por ejemplo, puede que desee para mostrar los elementos y relaciones que entran en juego durante el funcionamiento normal, pero relegar el control de errores o el procesamiento de excepciones a la documentación de apoyo.

La presentación primaria es más a menudo gráfica. Puede ser que sea un diagrama que ha dibujado en una notación informal utilizando una herramienta de dibujo simple, o podría ser un diagrama en una notación semiformal o formal importado de una herramienta de diseño o modelado que está utilizando. Si su presentación primaria es gráfica, asegúrese de incluir una clave que explica la notación. La falta de una clave es el error más común que vemos en la documentación en la práctica.

De vez en cuando la presentación principal será textual, tal como una tabla o una lista. Si ese texto se presenta de acuerdo a ciertas reglas de estilo, estas normas deben expresarse o incorporados por referencia, como el análogo a la tecla gráfica notación. Independientemente de si la presentación principal es textual en lugar de gráfica, su papel es el de presentar un resumen conciso de la información más importante en la vista.

• **Sección 2**: El Catálogo de Elementos. Los detalles del catálogo elemento al menos aquellos elementos representados en la presentación primaria. Por ejemplo, si un diagrama muestra los elementos A, B, y C, entonces el elemento de catálogo necesita explicar lo que A, B, y C son. Además, si los elementos o relaciones relevantes a esta vista se omitieron en la presentación primaria, deben ser introducidos y explicados en el catálogo. Partes específicas del catálogo se incluyen los siguientes:

• Los elementos y sus propiedades. Esta sección nombra cada elemento en la vista y enumera las propiedades de ese elemento. Cada vista presentado en el Capítulo 1 aparece un conjunto de propiedades sugeridas asociados con esa opinión. Por ejemplo, los elementos en una vista de descomposición pueden tener la propiedad de la "responsabilidad" explicación -an del papel de cada módulo en el sistema-y los elementos en una vista comunicantes-procesos podrían haber parámetros, entre otras cosas temporales, como propiedades. Si las propiedades son de carácter genérico a la vista elegida o el arquitecto ha introducido otros nuevos, aquí es donde se documentan y los valores dados.

• Relaciones y sus propiedades. Cada vista tiene tipos de relaciones específicas que representa a uno de los elementos en esa vista. Mayormente, estas relaciones se muestran en la presentación primaria. Sin embargo, si la presentación primaria no muestra todas las relaciones o si hay excepciones a lo que se muestra en la presentación primaria, este es el lugar para registrar esa información.

• Elementos de interface. Esta sección interfaces de documentos elemento.

• El comportamiento del elemento. Este elemento comportamiento documentos sección que no es obvio a partir de la presentación principal.

• **Sección 3**: Diagrama de contexto. Un diagrama de contexto muestra cómo el sistema o parte del sistema representado en esta vista relaciona con su entorno. El propósito de un diagrama de contexto es describir el alcance de la vista. Aquí "contexto" significa un entorno con el que la parte del sistema interactúa. Entidades en el medio ambiente pueden ser seres humanos, otros sistemas informáticos o los objetos físicos, tales como sensores o dispositivos controlados.

• **Sección 4**: Guía de variabilidad. Una guía de la variabilidad muestra cómo ejercer cualquiera de los puntos de variación que son parte de la arquitectura que se muestra en esta vista.

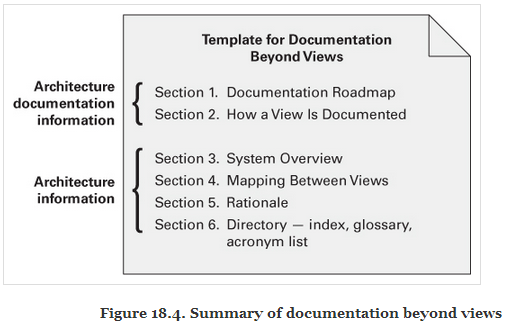
• **Sección 5**: Fundamentos. Fundamento explica por qué el diseño se refleja en la vista llegó a ser. El objetivo de esta sección es explicar por qué el diseño es como es, y para proporcionar un argumento convincente de que es sonido. La elección de un patrón en este punto de vista debe ser justificada aquí describiendo el problema arquitectónico que el patrón elegido resuelve y la justificación de la elección que sobre otra.

## La documentación de la información después de las Vistas

Como se muestra en la Figura 18.4, la documentación más allá de puntos de vista se puede dividir en dos partes:

**1. Visión general de la documentación arquitectura**. Esto indica cómo la documentación se presenta y organiza de manera que una de las partes interesadas de la arquitectura se puede encontrar la información que necesita de manera eficiente y fiable.

**2. Información acerca de la arquitectura**. Aquí, la información que aún no se ha capturado más allá de los puntos de vista a sí mismos es una visión general del sistema cortocircuito a tierra a cualquier lector en cuanto a la finalidad del sistema y la forma en que los puntos de vista están relacionados entre sí, una visión general de y razón de ser de diseño de todo el sistema enfoques, una lista de elementos y en el que aparecen, y un glosario y una lista de siglas para toda la arquitectura.



La Figura 18.4 resume nuestra plantilla para documentación más allá de puntos de vista. Documentación más allá de puntos de vista se compone de las siguientes secciones:

• información de control de documentos. Listar la organización emisora​​, los número de versión actual, fecha de emisión y el estado, un historial de cambios, y el procedimiento para la presentación de solicitudes de cambio al documento. Por lo general, esto se refleja en el texto preliminar. Herramientas de control de cambios pueden proporcionar mucha de esta información.

• **Sección 1: Guía básica de documentación**. La hoja de ruta de la documentación le dice al lector qué información se encuentra en la documentación y dónde encontrarlo. Un mapa de la documentación se compone de cuatro secciones:

• Alcance y resumen. Explicar el propósito del documento y un breve resumen de lo que está cubierto y (si usted piensa que va a ayudar) lo que no está cubierto. Explicar la relación con otros documentos (como los documentos de diseño descendente o documentos de ingeniería de sistemas de aguas arriba).

• ¿Cómo está organizada la documentación. Para cada sección de la documentación, dar una breve sinopsis de la información que se puede encontrar allí. Una alternativa a esto es utilizar un índice anotado. Esta es una mesa que no sólo enumera los títulos de secciones y números de página, sino que también da una sinopsis de cada entrada. Proporciona una ventanilla única para un lector de intentar buscar un tipo particular de información.

• Ver resumen. La mayor parte del mapa se describen los puntos de vista que el arquitecto ha incluido en el paquete. Para cada punto de vista, el mapa ofrece la siguiente información:

• El nombre de la vista y lo que el patrón se crea una instancia, si los hubiere.

• Una descripción de los tipos de la vista de elementos, tipos de relación, y los tipos de propiedad. Esto permite que un lector entiende el tipo de información que se presenta en la vista.

• Una descripción del lenguaje, técnicas de modelado, o los métodos de análisis utilizados en la construcción de la vista.

• ¿Cómo los interesados ​​pueden utilizar la documentación. El mapa sigue con una sección que describe lo que los interesados ​​y las preocupaciones se refiera cada vista; este hecho se recoge convenientemente en forma de tabla. En esta sección se muestra cómo las diversas partes interesadas pueden utilizar la documentación para ayudar a resolver sus inquietudes. Incluya escenarios cortos, como "Un mantenedor desea saber los unidades de software que pueden ser cambiados por una modificación propuesta. El mantenedor consulta la vista descomposición de entender las responsabilidades de cada módulo con el fin de identificar los módulos susceptibles de cambiar. El mantenedor entonces consulta la view1 usos para ver qué módulos utilizar los módulos afectados (y por tanto también podría tener que cambiar). "Para cumplir con la norma ISO / IEC 42010 hasta 2.007, debe tener en cuenta las preocupaciones de los usuarios, por lo menos, los compradores, desarrolladores y mantenedores.

• **Sección 2: ¿Cómo se documenta una vista. Aquí es donde se explica la organización de normalización que está utilizando para documentar visitas**-ya sea el que se describe en este capítulo o una de su preferencia. Le dice a sus lectores cómo encontrar información en una vista. Si su organización ha estandarizado en una plantilla para una vista, como debe ser, entonces simplemente puede hacer referencia a ese estándar. Si se carece de dicha plantilla, a continuación de texto como el que da por encima de describir nuestra plantilla de vista debería aparecer en esta sección de la documentación de la arquitectura.

• **Sección 3: Descripción general del sistema.** Esta es una breve descripción en prosa de la función del sistema, de sus usuarios, y cualquier fondo o limitaciones importantes. Esta sección ofrece a sus lectores con un modelo mental coherente del sistema y su propósito. Esto podría ser simplemente un puntero a un documento-concepto de operaciones.

• Sección 4: Correlación entre Reproducciones. Debido a que todos los puntos de vista de una arquitectura describen el mismo sistema, es lógico pensar que cualquiera de los dos puntos de vista tienen mucho en común. Ayudar a un lector a comprender las asociaciones entre vistas ayudará ese lector obtener una poderosa visión de cómo funciona la arquitectura como un todo conceptual unificado.

Las asociaciones entre los elementos a través de puntos de vista en una arquitectura son, en general, de muchos a muchos. Por ejemplo, cada módulo puede asignar a varios elementos de tiempo de ejecución, y cada elemento de tiempo de ejecución puede asignar a varios módulos.

Asociaciones-View-a la vista se pueden capturar convenientemente en forma de tablas. Enumerar los elementos de la primera vista de alguna orden de búsqueda conveniente. La tabla en sí se anotarán o introdujeron con una explicación de la asociación que representa; es decir, lo que la correspondencia entre los elementos a través de los dos puntos de vista. Los ejemplos incluyen "se implementa por" para la asignación de una vista de componentes-y-conector a una vista del módulo, "instrumentos" para la asignación de una vista del módulo a una vista de componentes-y-conector ", incluido en" para el mapeo de una visión de descomposición a una vista en capas, y muchos otros.

• **Sección 5: Fundamentos**. Esta sección documenta los decisiones arquitectónicas que se aplican a más de una vista. Prime candidatos incluyen la documentación de fondo o de organización restricciones o requisitos principales que llevaron a las decisiones de importación de todo el sistema. Las decisiones sobre qué patrones fundamentales de la arquitectura a utilizar son a menudo descritos aquí.

• **Sección 6: Directorio**. El directorio es un conjunto de material de referencia que ayuda a los lectores a encontrar más información de forma rápida. Incluye un índice de términos, un glosario y una lista de siglas.

## Documentación en línea, hipertexto y Wikis

Un documento puede ser estructurado en forma de páginas web enlazadas. En comparación con los documentos escritos con una herramienta de edición de texto, documentos orientados a la web suelen consistir en páginas cortas (creados para que quepa en una sola pantalla) con una estructura más profunda. Una página por lo general proporciona cierta información general y tiene enlaces a información más detallada. Cuando se hace bien, un documento basado en la web es más fácil de usar para las personas que necesitan información general. Por otra parte, puede llegar a ser más difícil para personas que necesitan detalle. Búsqueda de información puede ser más difícil en varias páginas, documentos basados ​​en la web que en un solo archivo, documento basado en texto, a menos que un motor de búsqueda está disponible.

El uso de herramientas fácilmente disponibles, es posible crear un documento compartido que muchas partes interesadas pueden contribuir a. La organización anfitriona tiene que decidir lo que los permisos se quiere dar a las diversas partes interesadas; la herramienta utilizada tiene que apoyar los política de permisos. En el caso de la documentación de la arquitectura, nos gustaría que todas las partes interesadas a comentar y añadir información aclaratoria a la arquitectura, pero sólo querríamos arquitectos sean capaces de cambiar la arquitectura o al menos proporcionar arquitectos con un mecanismo de "aprobación final". Un tipo especial de documento compartido que es ideal para este propósito es un wiki.

## Seguir una estrategia de lanzamiento

Plan de desarrollo de su proyecto debe especificar el proceso para mantener la documentación importante, incluyendo la documentación de arquitectura, actual. El arquitecto debe planear para emitir comunicados de la documentación para apoyar los principales hitos del proyecto, que por lo general significa lo suficiente por delante del hito que dan tiempo a los desarrolladores para poner la arquitectura al trabajo. Por ejemplo, el final de cada iteración o Sprint o versión incremental podría estar asociado con la provisión de documentación revisada al equipo de desarrollo.

## La documentación de los patrones

Los arquitectos pueden, y normalmente lo hacen, utilizar patrones como punto de partida para su diseño, como hemos discutido en el capítulo 13 Estos patrones podrían ser publicados en los catálogos existentes o en el repositorio de propiedad de una organización de los diseños estándar, o creados específicamente para el problema en la mano del arquitecto. En cada uno de estos casos, proporcionan un genérico (es decir, incompleto) enfoque de solución que el arquitecto tendrá que perfeccionar y crear una instancia.

En primer lugar, registrar el hecho de que se utiliza el patrón dado. Luego dijo por qué este enfoque solución fue elegida-por qué es una buena opción para el problema en cuestión. Si el enfoque elegido proviene de un patrón, esto consistirá esencialmente de demostrar que el problema en cuestión se ajusta el problema y el contexto del patrón.

Usando un patrón significa tomar decisiones de diseño sucesivas que eventualmente resultan en una arquitectura. Estas decisiones de diseño se manifiestan como elementos y relaciones recién instanciadas entre ellos. El arquitecto puede documentar una instantánea de la arquitectura en cada etapa. ¿Cuántas etapas hay depende de muchas cosas, no menos importante de los cuales es la capacidad de los lectores a seguir el proceso de diseño en caso de que tengan que volver a visitarlo en el futuro.

# Comportamiento documentado

Documentar una arquitectura requiere documentación comportamiento que complementa vistas estructurales mediante la descripción de cómo elementos de la arquitectura interactúan entre sí. Razonar sobre características tales como el potencial de un sistema a un punto muerto, la capacidad de un sistema para completar una tarea en el tiempo deseado, o el consumo máximo de memoria requiere que la descripción de la arquitectura contiene información sobre ambos los características de los elementos individuales, así como los patrones de interacción entre ellos, es decir, cómo se comportan entre sí. En esta sección, proporcionamos orientación en cuanto a qué tipos de cosas que usted tendrá que documentar para cosechar estos beneficios. En nuestra plantilla de vista de la arquitectura, el comportamiento tiene su propia sección en el catálogo de elementos.

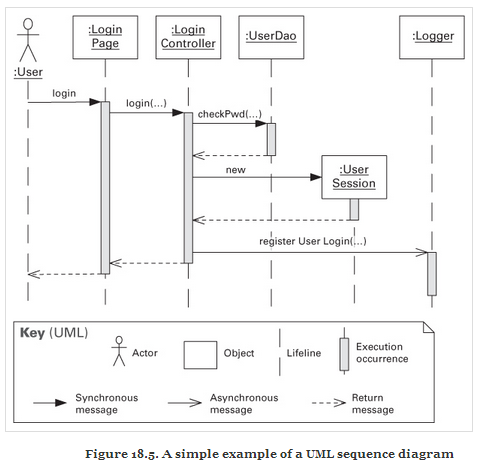
Hay dos tipos de notaciones disponibles para la documentación de comportamiento. El primer tipo de notación se llama lenguas traza-orientado; el segundo se llama lenguas integrales.

Las huellas son secuencias de actividades o interacciones que describen la respuesta del sistema a un estímulo específico cuando el sistema está en un estado específico. Una traza describe una secuencia de actividades o interacciones entre los elementos estructurales del sistema. Aunque es concebible para describir todos los rastros posibles para generar el equivalente de un modelo de comportamiento integral, no es la intención de la documentación traza orientada para hacerlo. A continuación se describen cuatro anotaciones en busca de rastros que documentan: casos de uso, diagramas de secuencia, diagramas de comunicación, y los diagramas de actividad. Aunque otras anotaciones están disponibles (por ejemplo, gráficos de secuencias de mensajes, diagramas de tiempo, y los Business Process Execution Language), hemos elegido estos cuatro como una muestra representativa de las lenguas traza-orientado.

• Los **casos de uso** describen cómo los actores pueden usar un sistema para lograr sus objetivos. Los casos de uso se utilizan con frecuencia para capturar los requisitos funcionales de un sistema. UML proporciona una notación gráfica para diagramas de casos de uso, pero no dice cómo se debe escribir el texto de un caso de uso. El diagrama de casos de uso UML puede ser utilizado eficazmente como una visión general de los actores y los comportamiento de un sistema. La descripción de casos de uso es textual y debe contener el nombre del caso de uso y una breve descripción, el actor o actores que inician el caso de uso (actores primarios), otros actores que participan en el caso de uso (actores secundarios), el flujo de los acontecimientos, los flujos alternativos y casos nonsuccess.

• Un **diagrama de secuencia** UML muestra una secuencia de interacciones entre las instancias de elementos extraídos de la documentación estructural. Muestra sólo las instancias que participan en el escenario están documentando. Un diagrama de secuencia tiene dos dimensiones: vertical, que representa el tiempo, y horizontales, que representan a las distintas instancias. Las interacciones están dispuestos en secuencia de tiempo de arriba a abajo. La figura 18.5 es un ejemplo de un diagrama de secuencia que ilustra la notación UML básica.

Objetos (es decir, instancias de elementos) tienen una línea de vida, dibujado como una línea vertical discontinua a lo largo del eje de tiempo. La secuencia se inicia normalmente por un actor en el extremo izquierdo. Los casos interactúan mediante el envío de mensajes, que se muestran como flechas horizontales. Un mensaje puede ser un método o una función llamada, un evento enviado a través de una cola, o algo más. El mensaje generalmente se asigna a un recurso (funcionamiento) en la interfaz de la instancia de receptor. Una punta de flecha lleno en una línea continua representa un mensaje sincrónico, mientras que la punta de flecha abierta representa un mensaje asíncrono. La flecha discontinua es un mensaje de retorno. Los bares de la ocurrencia de ejecución a lo largo de la línea de vida indican que la instancia está procesando o bloqueado a la espera de un retorno.

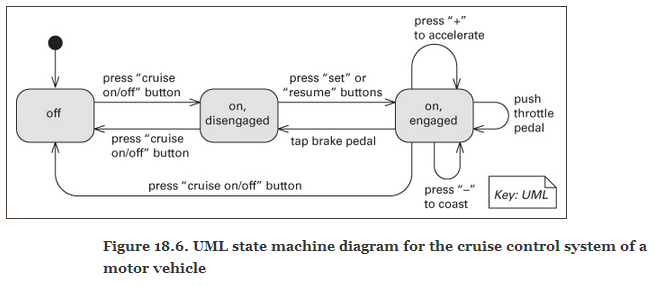


• **Un diagrama de comunicación** UML muestra un gráfico de los elementos que interactúan y anota cada interacción con un número que denota el fin. Del mismo modo para secuenciar los diagramas, los casos que se muestran en un diagrama de la comunicación son elementos que se describen en la documentación de acompañamiento estructural. Diagramas de comunicación son útiles cuando la tarea es verificar que la arquitectura puede cumplir con los requisitos funcionales. Los diagramas no son útiles si los comprensión de las acciones concurrentes es importante, ya que cuando se realiza un análisis de rendimiento.

• **diagramas de actividad** UML son similares a los diagramas de flujo. Ellos muestran un proceso de negocio como una secuencia de pasos (llamados acciones) e incluyen la notación para expresar bifurcación condicional y concurrencia, así como para mostrar el envío y recepción de eventos. Las flechas entre acciones indican el flujo de control. Opcionalmente, los diagramas de actividad pueden indicar el elemento de la arquitectura o el actor realizando las acciones. Los diagramas de actividades pueden expresar concurrencia. Un nodo tenedor (representado como una barra ortogonal de espesor a las flechas de flujo) divide el flujo en dos o más flujos simultáneos de acciones. Los flujos concurrentes pueden más tarde ser sincronizados en un único flujo a través de un nodo de unirse a (también representado como una barra ortogonal). La unión espera de nodo para todos los flujos entrantes para completar antes de continuar. A diferencia de la secuencia y de la comunicación diagramas, diagramas de actividad no muestran los operaciones reales que se llevan a cabo en objetos específicos. Los diagramas de actividades son útiles para describir en términos generales los pasos de un flujo de trabajo específico. Bifurcación condicional (rombo) permite que un solo diagrama para representar múltiples vestigios, aunque normalmente no es la intención de un diagrama de actividades para mostrar todos los rastros posibles o el comportamiento completo para el sistema o parte de él.

En contraste con rastrear notaciones, modelos integrales muestran el comportamiento completo de elementos estructurales. Ante este tipo de documentación, es posible deducir todos los caminos posibles de estado inicial a otro final. El formalismo de la máquina de estado representa el comportamiento de los elementos de la arquitectura, ya que cada estado es una abstracción de todas las historias posibles que podrían llevar a ese estado. **Lenguajes de máquina del Estado le permiten complementar una descripción estructural de los elementos del sistema con restricciones sobre las interacciones y reacciones programadas a los estímulos internos y ambientales.**

Notación UML diagrama de **máquina de estado** se basa en el formalismo gráfica statechart desarrollado por David Harel para el modelado de sistemas reactivos; que le permite rastrear el comportamiento de su sistema, dado insumos específicos. Un diagrama de máquina de estados UML muestra estados representados como cajas y las transiciones entre los estados representados como flechas. Los diagramas de máquina de estado ayudan a modelar elementos de la arquitectura y ayudar a ilustrar sus interacciones en tiempo de ejecución. Figura 18.6 es un ejemplo sencillo que muestra los estados de un sistema de control de crucero del vehículo.



Cada transición en un diagrama de máquina de estado se etiqueta con el evento que causa la transición. Por ejemplo, en la figura 18.6, los transiciones corresponden a los botones los conductor puede presionar o acciones motrices que afectan el sistema de control de crucero. Opcionalmente, la transición puede especificar una condición de guardia, que está encerrada entre paréntesis. Cuando se produce el evento correspondiente a la transición, la condición de guardia se evalúa y la transición sólo se activa si el protector es cierto en ese momento. Las transiciones también puede tener consecuencias, llamadas acciones o efectos, indicadas por una barra. Cuando se observa una acción, que indica que el comportamiento después de la barra se realizará cuando se produce la transición. Los estados también pueden especificar las acciones de entrada y salida.

Existen otras notaciones para describir el comportamiento integral. Por ejemplo, Arquitectura Análisis y Diseño Idioma (AADL) se pueden utilizar para razonar acerca del comportamiento de tiempo de ejecución. Las especificaciones y el lenguaje de descripción (SDL) se utiliza en la telefonía.

# Documentación de Arquitectura y Atributos de Calidad

Si la arquitectura es en gran parte sobre el logro de los atributos de calidad y si uno de los principales usos de la documentación arquitectura es la de servir como base para el análisis (para asegurarse de que la arquitectura alcanzará sus atributos de calidad requeridos), hacer que los atributos de calidad se muestran en los documentación? A falta de una visión de pleno derecho de la calidad, hay cinco formas principales:

1. Cualquier enfoque de diseño más importante (por ejemplo, un patrón de arquitectura) tendrá propiedades de los atributos de calidad asociados. Cliente-servidor es bueno para la escalabilidad, la estratificación es bueno para la portabilidad, una descomposición basada en la información escondite es bueno para modificabilidad, los servicios son buenos para la interoperabilidad, y así sucesivamente. Al explicar la elección del enfoque es probable que incluya una discusión sobre la satisfacción de la calidad del atributo requisitos y las desventajas incurridos. Busque el lugar en la documentación donde se produce tal explicación. En nuestro enfoque, a eso le llamamos razón.

2. elementos arquitectónicos individuales que proporcionan un servicio a menudo tienen límites de atributos de calidad que se les asignen. Los consumidores de los servicios necesitan saber qué tan rápido, seguro, confiable o esos servicios son. Estos límites de atributos de calidad se definen en los documentación de la interfaz de los elementos, a veces en los forma de un acuerdo de nivel de servicio. O simplemente pueden ser registrados como propiedades que los elementos de exposiciones.

3. Calidad atributos a menudo confieren a éste un "lenguaje" de las cosas que usted debe buscar. La seguridad implica niveles de seguridad, los usuarios autenticados, pistas de auditoría, cortafuegos, y similares. Performance trae a la mente las capacidades de amortiguación, los plazos, los plazos, las tasas de eventos y distribuciones, relojes y cronómetros, y así sucesivamente. Disponibilidad evoca el tiempo medio entre fallos, mecanismos de conmutación por error, la funcionalidad primaria y secundaria, los procesos críticos y no críticos, y los elementos redundantes. Alguien con fluidez en el "lenguaje" de un atributo de calidad puede buscar los tipos de elementos arquitectónicos (y las propiedades de esos elementos) que se pusieron en marcha precisamente para satisfacer ese requisito atributo de calidad.

4. documentación Arquitectura menudo contiene una asignación a los requisitos que muestra cómo se satisfacen las necesidades (incluyendo requisitos de atributos de calidad). Si su documento de requisitos establece un requisito de disponibilidad, por ejemplo, entonces usted debería ser capaz de mirar hacia arriba por nombre o referencia en su documento de arquitectura para ver los lugares donde ese requisito quedará satisfecho.

5. Cada requisito atributo de calidad tendrá un agrupamiento de partes interesadas que quieren saber que va a ser satisfecho. Para estos grupos de interés, el arquitecto debe proporcionar un lugar especial en la introducción de la documentación que, o bien responde a lo el actor está buscando, o dice el actor que en el documento para encontrarlo. Se diría algo como esto: "Si usted es un analista de rendimiento, se debe prestar atención a los procesos y subprocesos y sus propiedades (definidas [aquí]), y su despliegue en la plataforma de hardware subyacente (definido [aquí])." en nuestro enfoque documentación, ponemos este tope aquí tienes-que-estás-buscando información en una sección llamada los hoja de ruta de la documentación.

# Arquitecturas Documentar que cambian rápidamente que usted puede documentarlos

Cuando su navegador se encuentra con un tipo de archivo que nunca se ha visto antes, las probabilidades son que va a ir a Internet, buscar y descargar el plug-in adecuado para manejar el archivo, instalarlo, y reconfigurarse para usarlo. Sin siquiera tener que cerrar, y mucho menos pasar por el ciclo de desarrollo de código-integrar-test, el navegador es capaz de cambiar su propia arquitectura, añadiendo un nuevo componente.

Sistemas orientados a servicios que utilizan el descubrimiento de servicios dinámicos y vinculantes también exhiben estas propiedades. (Es decir, consciente de sí mismo) ya existen sistemas más desafiantes que son altamente dinámico, auto-organización, y reflexivo. En estos casos, la identidad de los componentes que interactúan entre sí no puede ser inmovilizados, por no hablar de sus interacciones, en cualquier documento de la arquitectura estática.

Otro tipo de dinamismo arquitectónico, igualmente difícil desde una perspectiva de la documentación, se encuentra en los sistemas que son reconstruidas y trasladados con gran rapidez. Algunos talleres de desarrollo, como los responsables de los sitios web comerciales, construir y "ir a vivir" con sus sistemas muchas veces todos los días.

Ya sea un cambio de arquitectura en tiempo de ejecución, o como resultado de un ciclo de lanzamiento y despliegue de alta frecuencia, los cambios ocurren mucho más rápido que el ciclo de la documentación. En cualquier caso, nadie se va a sostener las cosas hasta que se produce un nuevo documento de la arquitectura, la crítica, y puesto en libertad.

Pero conocer la arquitectura de estos sistemas es tan importante, y podría decirse que más, que para los sistemas en el mundo de los ciclos más tradicionales de vida. Esto es lo que puede hacer si usted es un arquitecto en un entorno altamente dinámico:

• Documentar lo que es cierto acerca de todas las versiones de su sistema. Su navegador web no se apaga y agarrar cualquier pieza de software cuando se necesita un nuevo plug-in; un plug-in debe tener propiedades específicas y una interfaz específica. Y esto no sólo tiene que enchufar en cualquier lugar, pero en un lugar predeterminado en la arquitectura. Anote esos invariantes como lo haría para cualquier arquitectura. Esto puede hacer que su arquitectura documentado más una descripción de las restricciones o directrices que cualquier versión compatible del sistema debe seguir. Eso está bien.

• Documento de las formas en que la arquitectura se le permite cambiar. En los ejemplos anteriores, lo cual suele significar la adición de nuevos componentes y la sustitución de componentes con nuevas implementaciones. En los Vistas y más allá enfoque, el lugar para hacer esto se llama la guía de la variabilidad (capturado en la Sección 4 de la plantilla de la vista).

# La documentación de la arquitectura en un proyecto de desarrollo ágil

"Ágil" se refiere a un enfoque de desarrollo de software que hace hincapié en el desarrollo rápido y flexible y de-enfatiza proyecto y la infraestructura de proceso para su propio bien. Aquí nos centramos sólo en la forma de documentar la arquitectura en un entorno Agile.

Los puntos de vista y más allá y ágiles filosofías totalmente de acuerdo en un punto central: Si no se necesita información, no documentarla. Toda la documentación debe tener un uso previsto y el público en mente, y se produjo de una manera que sirve a la vez. Uno de los principios fundamentales de la documentación técnica es "Escribir para el lector." Eso significa entender que va a leer la documentación y cómo lo utilizarán. Si no hay audiencia, no hay necesidad de producir la documentación.

Vista de la arquitectura de selección es un ejemplo de aplicación de este principio. Los puntos de vista y más allá enfoque prescribe que la producción de una vista si y sólo si aborda las preocupaciones de una comunidad identificada explícitamente las partes interesadas.

Otra idea central que debemos recordar es que la documentación no es una actividad monolítica que sostiene todos los demás progresos hasta que esté completa. El método de selección de vistas dada anteriormente prescribe la producción de la documentación en las etapas prioritarias para satisfacer las necesidades de los partes interesadas que lo necesitan ahora.

Cuando la producción de Vistas y la documentación arquitectura basada en Más allá utilizando los principios ágiles, mantenga en mente lo siguiente:

• Adoptar una plantilla u organización estándar para capturar sus decisiones de diseño.

• Plan de documentar una vista si (y sólo si) que tiene un electorado fuertemente identificado las partes interesadas.

• Rellene las secciones de la plantilla para una vista, y para obtener información más allá de puntos de vista, cuando (y en cualquier orden) la información está disponible. Pero sólo hacer esto si escribir esta información, será más fácil (o más barato o que el éxito es más probable) para alguien abajo haciendo su trabajo.

• No se preocupe por la creación de un documento de diseño arquitectónico y luego un documento de diseño más fino de grano. Producir información de diseño sólo lo suficiente para que pueda pasar a código. Captura la información de diseño en un formato que es fácil de usar y fácil de cambiar-un wiki, tal vez.

• No se sienta obligado a llenar todas las secciones de la plantilla, y ciertamente no todos a la vez. Todavía sugerimos que definir y utilizar plantillas de ricos, ya que pueden ser útiles en algunas situaciones. Pero siempre se puede escribir "N / A" para los secciones para las que no necesitas para registrar la información (tal vez porque usted transmitirlo por vía oral).

• Los equipos ágiles veces hacen modelos en breves deliberaciones de la pizarra. Al documentar una vista, la presentación principal puede consistir en una imagen digital de la pizarra. Más información acerca de los elementos (catálogo de elementos), la discusión lógica (la arquitectura de fondo), los mecanismos de variabilidad se utiliza (guía variabilidad), y todo lo demás puede ser comunicada verbalmente a-en los equipo menos por ahora. Más adelante, si se entera de que es útil para grabar una pieza de información acerca de un elemento, un diagrama de contexto, el fundamento para una determinada decisión de diseño, o alguna otra cosa, la plantilla tendrá el lugar correcto listo para recibirlo.

# Resumen

Escribiendo documentación arquitectónica es muy similar a otros tipos de escritura. Usted debe entender los usos a los que la escritura se va a poner y la audiencia para la escritura. Documentación arquitectónica sirve como medio de comunicación entre los distintos grupos de interés, no sólo en la cadena de gestión y hacia abajo a los desarrolladores, sino también a través de los compañeros.

Una arquitectura es un artefacto complicado, mejor expresado, centrándose en las perspectivas particulares en función del mensaje que se quiere comunicar. Estas perspectivas son llamados puntos de vista, y usted debe elegir los puntos de vista para documentar, debe elegir la notación para documentar estos puntos de vista, y deben elegir un conjunto de puntos de vista que es a la vez mínima y adecuada. Esto puede implicar la combinación de diversos puntos de vista que tienen un gran solapamiento. Usted debe documentar no sólo la estructura de la arquitectura, sino también el comportamiento.

Una vez que haya decidido sobre los puntos de vista, debe decidir cómo empaquetar la documentación. El embalaje dependerá de los medios utilizados para la expresión de la documentación. La impresión tiene características diferentes para la comprensión y la agrupación de varios medios de comunicación en línea. Diferentes medios de comunicación en línea también tendrán características diferentes.

El contexto del proyecto también afectará a la documentación. Algunos de los factores contextuales son los atributos de calidad importantes del sistema, la tasa de cambio del sistema, y la estrategia de gestión de proyectos.