



Anexo

Recomendaciones para el Diseño

Objetivo del documento

Este documento presenta un conjunto de recomendaciones generales para la elaboración del diseño.

Está basado en los problemas más recurrentes detectados en las revisiones técnicas de proyectos con características similares.

Aspectos de presentación y organización del entregable

Debe incluir una portada identificando claramente el entregable, equipo responsable, proyecto, fecha.

Debe incluir una contraportada identificando claramente responsables, control de cambios y cualquier otro aspecto de interés para el seguimiento del trabajo.

Deben incluirse índice de contenidos, figuras/diagramas, tablas... que faciliten la localización de los mismos en el documento.

Debe incluirse un apartado que describa la organización del documento y facilite su manejo.

Debe incluirse un apartado que describa los aspectos generales de diseño y que sirva de introducción para facilitar el correcto entendimiento del mismo.

La presentación del diseño debe comenzar introduciendo el sistema completo general. Desde el más alto nivel de abstracción, presentando subsistemas, componentes, atribuciones de los mismos, **interfaces internas y con componentes externos**, organización en paquetes... y con una breve explicación que resuma de forma clara estos aspectos generales.

Las dependencias de sistemas externos son críticas, no deben obviarse y por tanto deben manifestarse claramente. Si es necesario pueden indicarse los riesgos derivados de estas dependencias y las medidas "paliativas".

El resto del diseño (más concreto, detallado y refinado) debe ser coherente con este diseño general y debería organizarse por subsistemas o componentes, de forma que durante toda la descripción se tenga siempre clara la referencia a este diseño general.

Se recomienda que la asignación de tareas a los miembros del equipo de proyecto se realice por subsistemas, y que en las reuniones de coordinación se asegure el correcto entendimiento de las interfaces.

Si el sistema no se ha diseñado por completo debe aclararse qué subsistemas o componentes se acometen y cuáles quedan fuera del ámbito del modelo presentado. **Esta decisión debe ser coherente con el plan de proyecto.**

A la hora de explicar con más detalle el modelo de un subsistema o componente es conveniente comenzar con una introducción al mismo que profundice en sus funciones/atribuciones, dependencias, interfaces usadas e interfaces ofrecidas y describa brevemente su estructura.

Presentar los casos de uso en los que interviene un subsistema o componente y los diagramas (de cualquier tipo) que describen el comportamiento en dichos casos de uso, puede facilitar su explicación.

Debe incluirse un **apartado con las referencias** de interés que incluya cualquier; estándar, documento, solución, aproximación, trabajo previo... en el que se haya apoyado el trabajo y que facilite la localización de las mismas para su consulta.

Aspectos generales relacionados con el modelo

Todas las entidades del modelo (paquetes, clases, objetos, componentes...) deben tener una justificación, es necesario mantener el principio KISS.

Los nombres de todas las entidades deben facilitar la identificación y comprensión de las mismas y por tanto deben ser claros (no ambiguos), significativos, coherentes, constantes (no pueden cambiar a lo largo del diseño) y en ningún caso deben dar lugar a confusión.

De forma sistemática es recomendable incluir trazas desde las entidades hacia la ERS, esto puede facilitar la comprensión de las mismas.

Se asume como ciertas las reglas:

1. Si dos entidades se llaman igual es porque son la misma.
2. Si dos entidades se llaman de forma diferente es porque no son la misma.
3. Si se diera alguna excepción a las reglas 1 ó 2 es necesario indicarlo claramente y de forma explícita.

La funcionalidad y capacidades atribuidas a cada entidad del modelo deben estar correctamente definidas, de forma clara y coherente. No deben cambiar a lo largo del modelo.

La granularidad de las entidades debe estar correctamente controlada pensando en la implementación del sistema.

El modelo no debe contener:

- Contradicciones: las capacidades, funcionalidad, interfaz... de una entidad deben ser siempre las mismas y no cambiar a lo largo del modelo.
- Duplicados: no se le debe asignar las mismas capacidades o funcionalidades a dos entidades distintas sin una justificación adecuada y clara. La misma entidad debe llamarse igual a lo largo de todo el modelo.
- Entidades sin atribuciones
- Ambigüedades

El diseño debe especificar claramente las interfaces provistas y usadas. Cuando se declara una interfaz es necesario especificar sus métodos detallando la firma de los mismos.

Si esta interfaz es de un sistema externo la especificación y uso de la misma debe ser coherente con la publicación detallada por el equipo responsable. Si hubiera algún problema con este asunto debe indicarse de forma explícita la causa del mismo.

Si un sistema externo tiene unas capacidades atribuidas éstas no pueden duplicarse en el sistema y es imprescindible recurrir a él a través de las interfaces publicadas.

Si se decide introducir un elemento, eliminarlo, modificarlo, se hace un refinamiento, etc... todo el modelo debe ser consistente, y debe revisarse para verificar que los cambios introducidos no provocan inconsistencias “hacia atrás”.

Aspectos sobre el uso de patrones

Los patrones deben solucionar problemas existentes que se hayan identificado. No deben aparecer patrones que no aportan nada y que únicamente complican el diseño del sistema.

Es necesario justificar el uso de cada patrón identificando el problema y cómo lo soluciona.

Es necesario mostrar, **a lo largo de todo el diseño**, cómo se aplican los patrones. De modo que en cualquier diagrama en el que los componentes afectados por algún patrón aparezcan deben estar presentes los artefactos necesarios para la aplicación del mismo.

El patrón Modelo/Vista/Controlador debe aplicarse en todos los proyectos.

Aspectos generales relacionados con los diagramas

Los nombres de los diagramas deben ser suficientemente explicativos, no ambiguos, y aparecer siempre en el documento.

Debe incluirse una breve explicación sobre el diagrama que incluya un resumen de su contenido y cualquier otro tipo de información de interés sobre el mismo. El uso de la facilidad para mostrar la información de un diagrama en Magic Draw (**show diagram info**) puede resultar de gran ayuda para aplicar esta recomendación de forma sistemática.

Es recomendable incluir trazas desde los diagramas hacia la ERS, esto puede facilitar la comprensión de los mismos.

Los diagramas del documento deben poder verse, el tamaño de la letra debe permitir su lectura si se imprime en papel (sin necesidad de usar el zoom) o si se incluye en una presentación. Si un diagrama completo fuera demasiado grande puede fraccionarse pero atendiendo siempre la siguiente recomendación.

Si hay diagramas relacionados (porque uno detalla al otro, se refiere a él, lo engloba, lo completa...) estas relaciones deben aparecer de forma explícita en los mismos. Se deberían utilizar comentarios y enlaces que permitan la rápida “navegación” entre diagramas.

Incluir cualquier tipo de comentario en los diagramas que pueda facilitar la comprensión de los mismos.

Sobre los diagramas de clases

Utilizar los estereotipos **boundary**, **entity** y **control** para matizar el papel de las clases según el patrón Modelo/Vista/Controlador cuando sea posible.

Priorizar el uso de relaciones frente a la inclusión de atributos para especificar las relaciones entre clases. Esto permite una mayor flexibilidad a la hora de la implementación.

Aclarar siempre la multiplicidad de las relaciones.

Deben quedar claramente reflejadas las clases que implementan las interfaces ofrecidas por el sistema.

Deben quedar claramente reflejadas las clases que usan las interfaces requeridas por el sistema.

La granularidad de las clases debe facilitar la reutilización, evolución y sustitución de las mismas.

Sobre los diagramas de actividad

Debe aclararse a qué entidad/es se refieren y el nivel de abstracción de los mismos.

Debería ser fácil enlazar desde una entidad determinada del modelo con el/los diagrama/s de actividad en los que interviene (en los que hay acciones de las que es responsable).

Si en una acción se usa, crea, modifica... alguna entidad (objeto) esto debe quedar reflejado en el diagrama.

Si un diagrama de actividad incluye acciones ejecutadas por diferentes entidades debe recurrirse al uso de “**swimlanes**”. Debería ser fácil enlazar desde el diagrama de actividad con los diagramas que detallan tanto las entidades responsables de las acciones como las involucradas en la ejecución de las mismas.

Sobre los diagramas de máquina de estados

Debe aclararse a qué entidad, y si fuera necesario en qué contexto, se refieren los estados presentados.

Los estados no deben representar acciones. Para especificar acciones a la entrada/salida o durante el estado se debe recurrir a las capacidades UML (a través de la especificación o propiedades del estado).

Las transiciones deben reflejar claramente los disparadores, así como los efectos y guardas (cuando sean relevantes).

Si las acciones involucradas están detalladas/refinadas en algún otro diagrama debería poder enlazarse a él de forma rápida/fácil.

Sobre los diagramas de secuencia

Deben quedar claras las entidades participantes y los mensajes intercambiados deben ser coherentes con las interfaces de las mismas, definidas en el modelo. Debe poder resultar fácil enlazar desde este diagrama con aquellos en los que se especifica con más detalla las entidades participantes.

Debe quedar reflejada claramente la firma de cada mensaje/método que aparece en el diagrama, especificando incluso valores de los parámetros cuando sea necesario para una correcta comprensión del mismo.

Si hay representaciones de control del tipo bucle, condición, opciones... debe quedar claro el comportamiento y ser coherente con otros diagramas que representen estos comportamientos (actividades, estados...).