Patrones Software
Tema 1-2: Introducción

Introducción a los Patrones de Diseño

- Los analistas/diseñadores con gran experiencia aplican, de forma intuitiva y automática, criterios precisos que, en general, solucionan de forma elegante y efectiva los problemas de modelado software de sistemas reales.

- Usualmente estos diseñadores utilizan métodos, estructuras y subsistemas que son, a la vez, herramientas de diseño y partes de la solución final, de una manera que difícilmente puede transmitirse, en un sentido formal, a especialistas menos expertos.

Introducción a los Patrones de Diseño

- Los "ingenieros de software" se enfrentan cada día a multitud de problemas con distinto grado de dificultad.
- La "efectividad" de un "ingeniero" se mide por su rapidez y acierto en la diagnosis, identificación y resolución de tales problemas.
- El mejor "ingeniero" es el que más reutiliza la misma solución -matizada- para resolver problemas similares.

Introducción a los Patrones de Diseño

- Los diseñadores experimentados poseen un sentido especial que "detecta" la completitud, en un sentido eminentemente arquitectónico, de un determinado diseño, con independencia de las posibles métricas y paradigmas utilizados.
- Naturalmente lo ideal sería extraer la "esencia" de estos afortunados diseños para formular una "poción" que pudieran ingerir los diseñadores noveles.

Orígenes de los Patrones de Diseño

- Los trabajos de Christopher Alexander intentan identificar y resolver, en un marco descriptivo formal, problemas esenciales en el dominio de la arquitectura.
- El traslado de muchas de sus ideas al desarrollo de software ha parecido el más adecuado a los diseñadores.
- Alexander ha servido, en realidad, de catalizador de ciertas tendencias "constructivas" utilizadas en el diseño de sistemas software.

Orígenes de los Patrones de Diseño

- ¿Existe en verdad una parte común en los buenos diseños, a veces tan dispares entre sí? Christopher Alexander así lo afirma y da la siguiente definición de patrón:

  "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma".

Orígenes de los Patrones de Diseño

- En 1987 Ward Cunningham y Kent Beck utilizan las ideas de Alexander para desarrollar un lenguaje de patrones como guía para los programadores de Smalltalk, dando lugar al libro "Using Pattern Languajes for Object-Oriented Programs".
- En 1991 Jim Coplien publica el libro "Advanced C++ Programming Styles and Idioms", donde realiza un catalogo de "idioms" (especie de patrones).
- Entre 1990 y 1994, Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (conocidos como GoF (Gang of Four) “la banda de los cuatro”) realizan el primer catálogo de patrones de diseño, que publican en el libro **"Design Patterns: Elements of Reusable Object-Oriented Software"**

Patrones de Software y OO

- Los **patrones** para el desarrollo de software son uno de los últimos avances de la Tecnología Orientada a Objetos.
- Los patrones se convierten en una parte muy importante en las Tecnologías Orientadas a Objetos para poder conseguir la **reutilización**. Entre los beneficios que se consiguen con la reutilización están:
  1. Reducción de tiempos.
  2. Disminución del esfuerzo de desarrollo y mantenimiento.
  3. Eficiencia.
  4. Consistencia.
  5. Fiabilidad.
- Las **metodologías** Orientadas a Objetos tienen como principio *“no reinventar la rueda”* para la resolución de diferentes problemas.

Patrones de Software + IS

- La **ingeniería del software** se enfrenta a problemas variados que hay que identificar para poder utilizar la misma solución (aunque matizada) con problemas similares.
- Los patrones son una forma documentada para resolver problemas de ingeniería del software.
- Se necesita algún esquema de documentación que permita la comunicación de resultados obtenidos por distintos ingenieros. No sirve solo con la documentación de líneas de código, hay que documentar análisis, diseños, arquitecturas, etc.
- El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores para comunicar experiencia sobre los problemas y sus soluciones.

Patrones de Software. Definición

- *Un patrón es una información que captura la estructura esencial y la perspicacia de una familia de soluciones probadas con éxito para un problema repetitivo que surge en un cierto contexto y sistema.*
- *Un patrón es una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto.*
- *Cada patrón es una regla de tres partes, la cual expresa una relación entre un cierto contexto, un conjunto de fuerzas que ocurren repetidamente en ese contexto y una cierta configuración software que permite a estas fuerzas resolverse por si mismas.*

Patrones de Software. Características

- **Solucionar un problema:** Los patrones capturan soluciones, no sólo principios o estrategias abstractas.
- **Ser un concepto probado**: Los patrones capturan soluciones demostradas, no teorías o especulaciones.
- **La solución no es obvia:** Muchas técnicas de solución de problemas tratan de hallar soluciones por medio de principios básicos. Los mejores patrones generan una solución a un problema de forma indirecta.
- **Describe participantes y relaciones entre ellos:** Los patrones no sólo describen módulos sino estructuras del sistema y mecanismos más complejos.

Patrones de Software. Clasificación

- **Patrones de arquitectura:** Expresa una organización o esquema estructural fundamental para sistemas software.
- **Patrones de diseño:** Proporciona un esquema para refinar los subsistemas o componentes de un sistema software, o las relaciones entre ellos.
- **Patrones de programación (Idioms patterns):** Un idioma es un patrón de bajo nivel de un lenguaje de programación específico.
- **Patrones de análisis:** Describen un conjunto de prácticas que aseguran la obtención de un buen modelo de un problema y su solución.
- **Patrones organizacionales:** Describen la estructura y prácticas de las organizaciones humanas, especialmente en las que producen, usan o administran software.

Patrones de Software. Clasificación

- También se puede hablar de otros tipos de patrones software, como pueden ser:

1. Patrones de programación concurrente.
2. Patrones de interfaz gráfica.
3. Patrones de organización del código.
4. Patrones de optimización de código.
5. Patrones de robustez de código.
6. Patrones de la fase de prueba.

Patrones de Software. Ventajas

- Ventajas de la utilización de patrones:

1. Facilitan la comunicación interna.
2. Ahorran tiempo y experimentos.
3. Mejoran la calidad del diseño y la implementación.
4. Son como “normas de productividad”.
5. En Java facilitan su aprendizaje y comprender cómo esta diseñado el propio lenguaje.

Patrones de Software. Notas

- ¿Patrones como moda?
  - Mejor como Intención.
- ¿Patrones como solución?
  - Mejor como núcleo de soluciones.
- ¿Patrones como normas?
  - Mejor como sugerencias.