1. Refactorización

El problema es que, con cada cambio que le hacemos al software, éste se degrada en cuanto a calidad del código.

Definición 1.1. (Entropía) Entropía del software: una degradación de la calidad inexorablemente continua y creciente.

1.1. Refactorización al rescate

Definición 1.2. (refactorización como técnica) Refactorización es una técnica que busca mejorar la calidad del código con vistas a facilitar su mantenimiento, pero sin alterar el comportamiento observable del mismo.

Definición 1.3. (refactorización como un cambio) Una refactorización es un cambio realizado sobre el código de un sistema de software de modo tal que no se afecte su comportamiento observable y con el sólo fin de mejorar la legibilidad, la facilidad de comprensión, la extensibilidad u otros atributos de calidad interna con vistas al mantenimiento.

Asegurar la preservacipon preservación del comportamiento luego de una refactorización Los métodos analíticos, y en especial los de análisis estático, son los que utilizan las herramientas de refactorización automática.

Pero no toda refactorización es posible de resolver por métodos analíticos ni automatizable. Por eso se suelen usar pruebas para asegurar la preservación del comportamiento. En definitiva, para permitir refactorizaciones más seguras y confiables, una buena táctica es trabajar con pruebas unitarias automatizadas, escritas antes de refactorizar, y correrlas después para asegurarnos que nuestro programa sigue funcionando tal como lo hacía antes del cambio.

1.2. A qué llamamos observable

El conjunto de entradas y el conjunto de salidas debería ser el mismo antes y después de una refactorización.

Lo que busca una refactorización es que se preserve lo que un usuario espera del sistema, que es lo que habitualmente llamamos requerimientos.

Definición 1.4. (refactorización (basada en requerimientos)) Una refactorización es un cambio realizado sobre el código de un sistema de software, con el sólo fin de mejorar la calidad interna con vistas al mantenimiento, de modo tal que, luego de esa transformación, se sigan cumpliendo los requerimientos de la aplicación

1.3. Refactorización como parte de TD

La práctica de refactorización permanente nos ayuda a que el diseño vaya evolucionando conforme los cambios van surgiendo. ver libro fontenla pag 135.