



RECUPERACIÓN DEL SOFTWARE LEGACY

Carlos Fontela



Frase del día



Dado que en muchas de las situaciones en que se deben mantener sistemas heredados el problema es la reconstrucción del conocimiento, se nos ocurrió abordarlo mediante la práctica de especificar con ejemplos, con su carga de actividad colaborativa y de descubrimiento de comportamiento.

(Carlos Fontela, “Planteo metodológico para la recuperación del software heredado”, 2015)

Recuperación del software heredado

Una propuesta desde
FIUBA

Del LIMET
(Laboratorio de
Métodos de Desarrollo y
Mantenimiento de
Software)

Fragilidad de las soluciones (ingenuas) propuestas

Tirar y reescribir

Costo y tiempo impracticables en general

Clientes no dispuestos a esperar el desarrollo de un nuevo sistema sin seguir pidiendo cambios al anterior => blanco móvil

Enormes emprendimientos difíciles de estimar

=> fechas de puestas en producción poco fiables

Enfoque documental (manuales de usuario y documentación de las APIs)

Documentación desactualizada

El sistema funciona por lo que está representado en el código

Fragilidad del (exitoso) planteo de Feathers

Feathers tiene razón: las pruebas son necesarias

Pero no alcanzan las pruebas unitarias y de integración técnicas

Ni sabemos si cada porción de código en particular funcionan tan bien como el sistema a mantener

Necesitamos especificaciones de características

Que evidencien el comportamiento desde la perspectiva de interacciones reales de usuarios

Descripción de la arquitectura: pruebas de integración

Propuesta LIMET:

SBE para recuperar conocimiento

Ir acotando el sistema existente por funcionalidades, las cuales se deberán especificar con ejemplos que sirvan como casos de aceptación



Intermedio: ¿qué es SBE?

Técnica de especificación de software

Ejemplos de uso del sistema

Documentación viva

Casos de prueba de aceptación

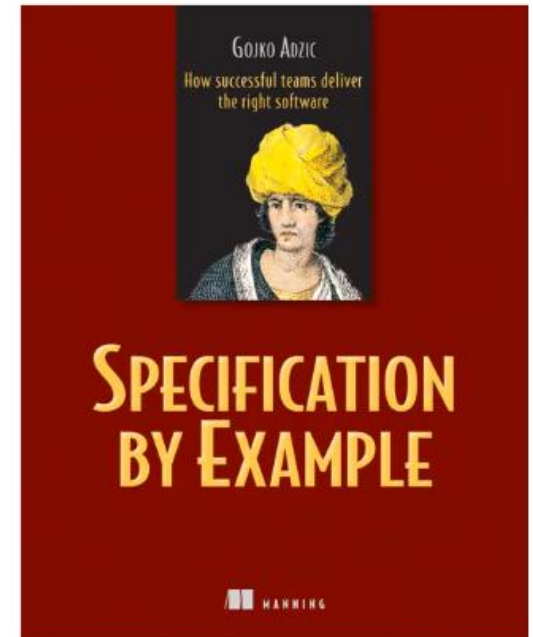
Especificaciones ejecutables

Talleres multidisciplinares para construcción de ejemplos

Obtención de un lenguaje en común

Relacionado con el lenguaje ubicuo de Domain Driven Design

Volumen



SBE en la recuperación de software heredado

Ejemplos para descubrir funcionalidades

Hipótesis en formato de ejemplos

Ver cómo reacciona el sistema

Construidos en talleres de interesados multidisciplinarios

Luego, refactorizar o cambiar el sistema

Los mismos ejemplos servirán para probar el sistema cambiado



Orden de los cambios

¿Trabajamos con el sistema congelado?

Recordar: todo cambio tiene que tener valor para el negocio

Aplicar el método por demanda

Pedido de cambios

Taller de especificaciones y construcción de ejemplos

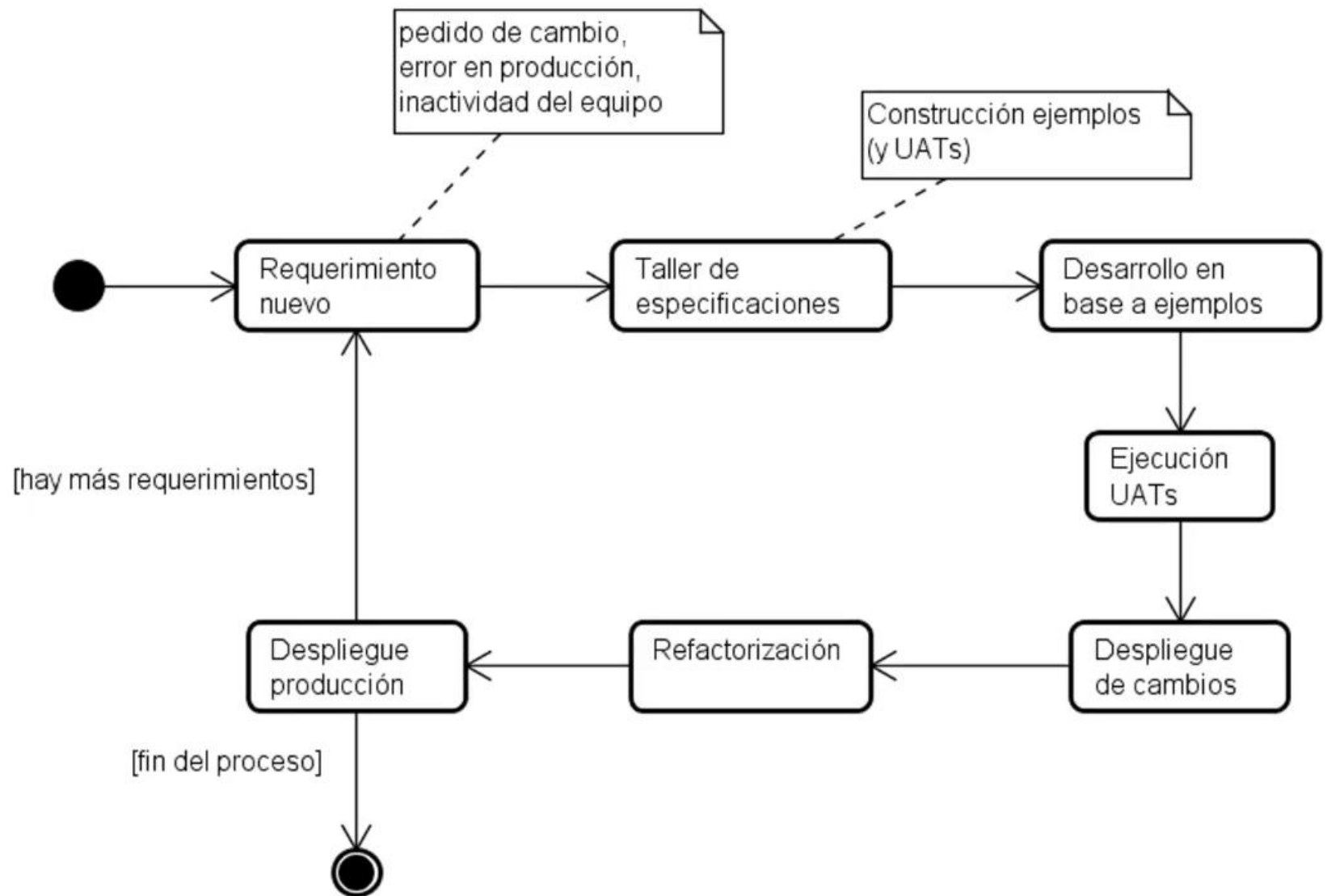
Implementación de los cambios

Reemplazo por partes, de manera incremental

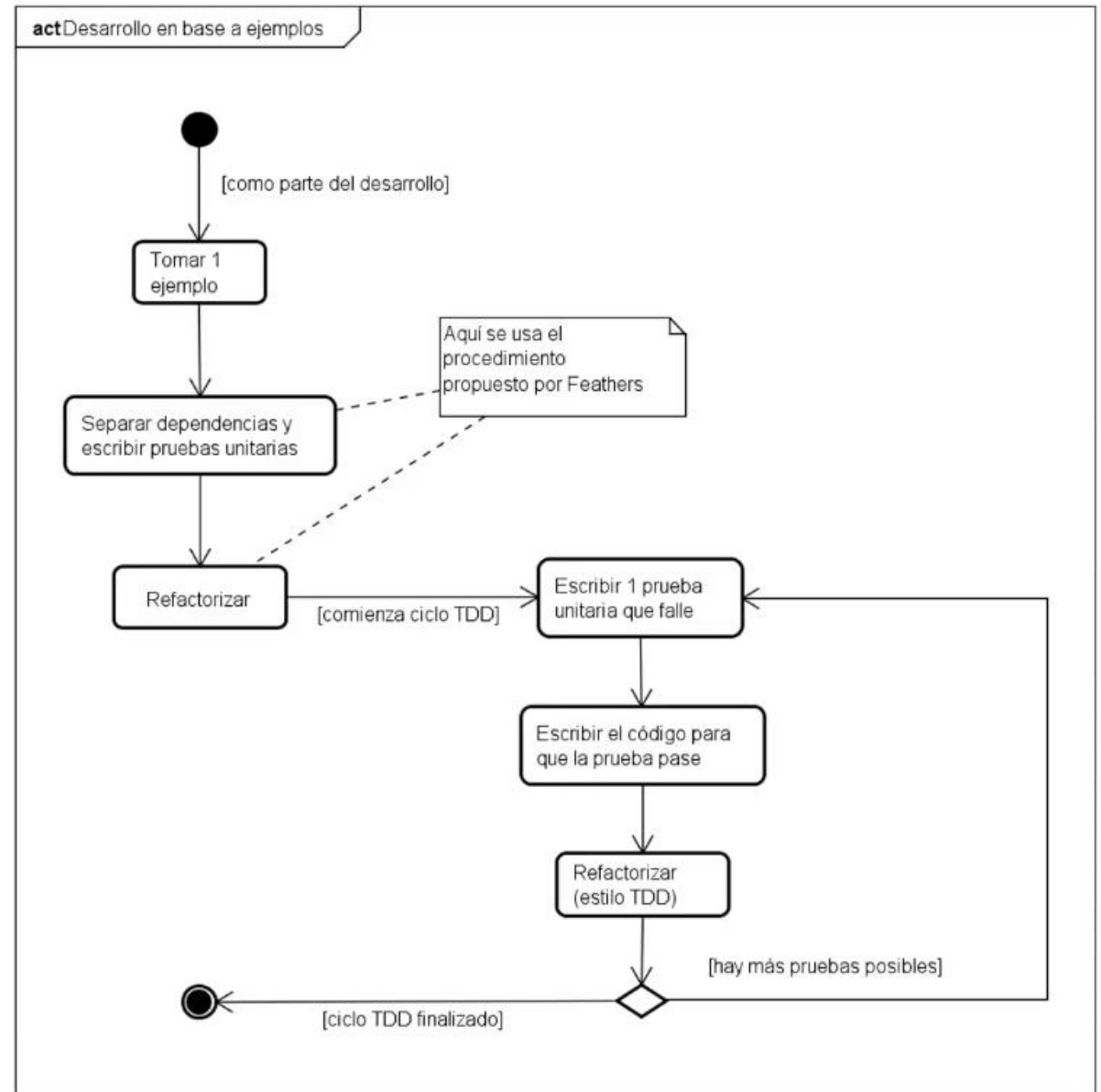
El resto queda igual

y sólo se cambian partes que interactúan con el sistema viejo a través de alguna API

Procedimiento



Parte “Desarrollo en base a ejemplos”



¿Y la arquitectura?

¿Microservicios?

El uso del lenguaje ubicuo y contextos encerrados parece sugerirlo

Las APIs y comunicación entre módulos también

Pero cuidado con los costos de sistemas distribuidos

Lo que importa no es la implementación de la arquitectura

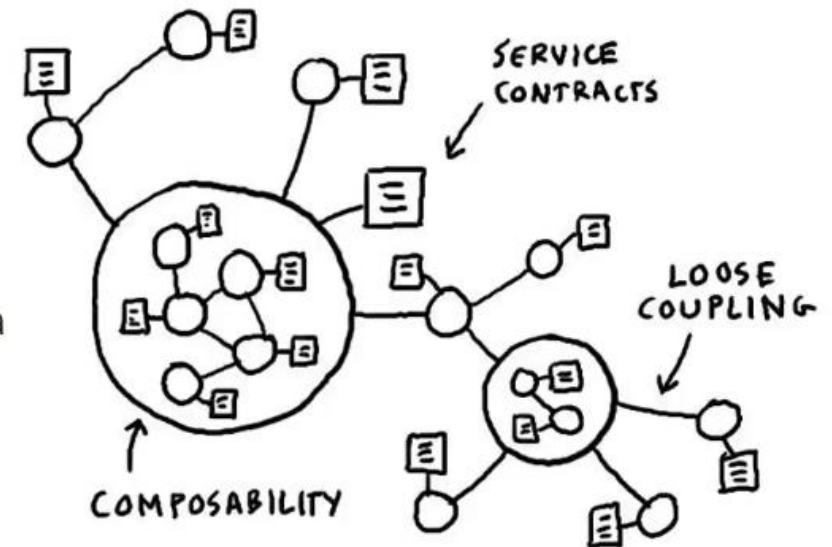
Volver a los principios de diseño

Modularidad

Bajo acoplamiento entre módulos

Etc.

SERVICE ORIENTATION



Casos de estudio en FIUBA

Trabajos Profesionales dirigidos desde el LIMET

Martín Margonari, 2017-2020, “Especificaciones con Ejemplos para la recuperación del conocimiento y evolución de la aplicación Recorridas de la Fundación SI”.

Nadia Galli y Martín Zaragoza, 2018-2019, “La evolución de Open Navent Jobs (Creación de una API abierta para terceros y una interna, ambas basadas en microservicios, mediante el uso de métodos de especificación con ejemplos)”.

Relacionados:

Nicolás Dascanio, 2013-2014 (tesis), “Análisis de impacto de cambios realizados sobre sistemas con pruebas automatizadas de distinta granularidad”

Lionel Raymundi, 2012- 2015. Reingeniería de una aplicación Java legacy en funcionamiento

Recapitulación

Hemos postulado la posibilidad de usar SBE para recuperar software legacy

E hicimos algunas pruebas

Falta probar en escala industrial

