Aseguramiento de la Calidad de Construcción de Software en NTE

Joan Bosch, Sara Martin

NTE, S.A

Introducción

- Enfoque tradicional de los departamentos de calidad: pruebas funcionales.
- Las pruebas funcionales no garantizan la robustez, la mantenibilidad y la flexibilidad del producto.
- Estas características dependen exclusivamente de la meticulosidad de los desarrolladores.
- Hace falta extender las actividades de calidad a todas las fases del desarrollo de software.

2

Los dos equipos de calidad de NTE (I)

Un equipo de pruebas funcionales que además asegura la calidad de los requerimientos, realiza el análisis de riesgos y programa herramientas de automatización.

Miembros: ingenieros de software enfocados al análisis funcional.

3

Los dos equipos de calidad de NTE (II)

Un equipo de ACCS (Aseguramiento de la Calidad de Construcción de Software) que trabaja en paralelo a las etapas de arquitectura, diseño, codificación,...

Miembros: profesionales con experiencia en el desarrollo de software.

4

Esquema del proceso de desarrollo de NTE (I)

- **Fase de requerimientos**
  - Representante del cliente.
  - Casos de uso, diagramas conceptuales y diccionarios de datos.
- **Fase de arquitectura y diseño**
  - Arquitecturas flexibles ante cambios de requerimientos.
  - Mapeo estricto entre clases y conceptos de usuario.
  - Prototipos de interfaces: pruebas de usabilidad.

5

Esquema del proceso de desarrollo en NTE (II)

- **Fase de codificación**
  - Integración continua
  - Pruebas unitarias, automáticas, incrementales y regresivas.
  - Uso de herramientas de cobertura.
  - Prueba básica diaria.
- **Fase de pruebas funcionales**
  - Equipo independiente de pruebas funcionales.
  - Casos de prueba. Cuando se estabilizan, se automatizan.

6

Responsabilidades de ACCS (I)

Asegurar

- la aplicación de los **estándares de desarrollo** de NTE.
- la **consistencia** y **trazabilidad** entre los requerimientos, la arquitectura, el diseño y el código.

- Seleccionar la **estrategia** de Caja Blanca.

- Proporcionar soporte a las **herramientas** de pruebas unitarias, de cobertura, etc.

- Ser el **repositorio** de los conocimientos técnicos de la compañía.

7

Responsabilidades de ACCS (II)

Revisar

- la **arquitectura**.

- el **diseño** de bajo nivel.

- el **código**.

- la **documentación técnica**.

8

Flujo de trabajo de ACCS

- Introducción **paso a paso** de la metodología.
- Proyecto **piloto**.
- **Paralelo** al desarrollo.
- Cuando costumbre de trabajo desarrollo – ACCS, **extensión actividades**.

9

Resistencias detectadas en la implantación de ACCS

- **A la visibilidad.** Los desarrolladores no querían "ser vistos" o sentirse "controlados".
- **A la transparencia.** Dificultades de comunicación de las personas técnicas y mantenimiento de la documentación.
- **Otras.** Adopción de estándares como veto a la creatividad y cambios en viejos hábitos adquiridos.

10

Problemas técnicos más frecuentes

- **Diseños pobres.** Rehacer continuamente versus crear un diseño estable al principio.
- **Documentación pobre.** O no se documenta o no se refleja el código.
- **Desconocimiento** de técnicas de **pruebas unitarias** y de integración. Ausente en los planes de estudios y falta de interés de los desarrolladores.

11

Estrategia de introducción de ACCS

- Miembros de ACCS: **personas técnicas**.

- Mismo acceso **a la información y a las herramientas** que los desarrolladores. Precisión y rigor.

- **Integración continua**. Supervisión.

- **Revisiones** de diseño informales preceden a las revisiones formales.

- **Ejemplos reales** como argumento.

12

Resultados conseguidos (I)

La calidad, algo **más** que pruebas funcionales:

- Encuentra **soluciones técnicas y metodológicas.**

- Se **comprende** al desarrollador.

- Calidad del **diseño**.

13

Resultados conseguidos (II)

- **Comunicación** entre equipos: se comparten técnicas y herramientas. Más **homogeneidad**.

- Cultura de empresa: metodologías compartidas y solicitadas a ACCS espontáneamente. **Repositorio**.

14

Conclusiones (I)

- **Mejor interacción** entre desarrolladores y miembros del departamento de calidad ✍ **mejor calidad global** del producto.
- A **más codificación de pruebas unitarias** ✍ **menos** tiempo invertido en **fijar errores.**
- **Menor tiempo de desarrollo global**, pese al impacto inicial.

15

Conclusiones (II)

- **Mayor estabilidad** del producto.
- Invertir en **buscar e implantar técnicas de calidad** en la primeras fases de un proyecto ✍ **beneficios** a medio plazo para toda la empresa.

16