

Plan de calidad del software en Fagor Automation

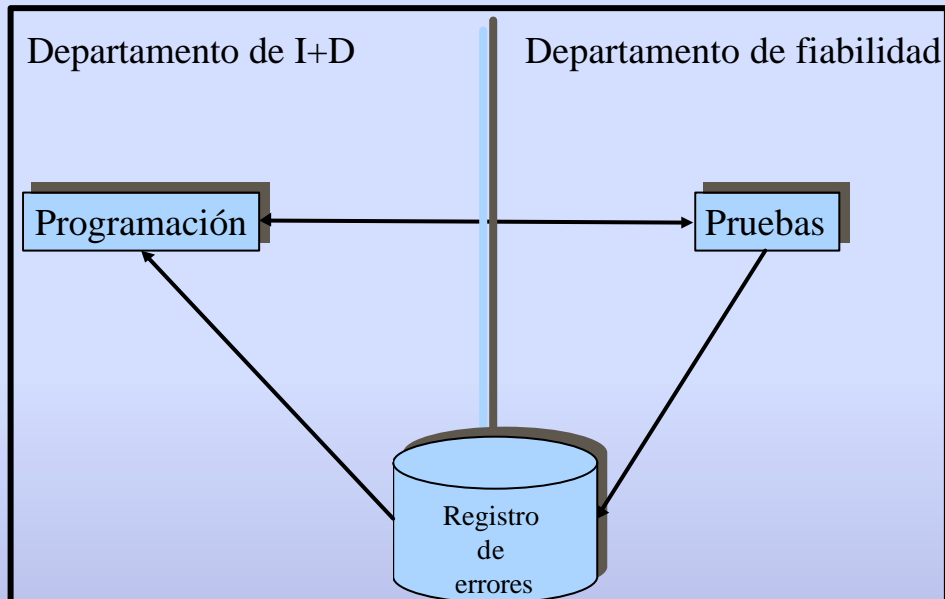
Goiuria Sagardui
Mondragon Unibertsitatea
Mondragon Goi Eskola Politeknikoa
Departamento de informática
gsagardui@eps.muni.es

Josu Onandia
Fagor Automotion S. Coop.
Departamento de I+D
jonandia@fagorautomation.es

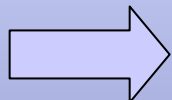
Introducción

- Empresa: Fagor Automation, Dept. I+D, Grupo Software
- Productos:
 - Automatización de Máquina Herramienta: Motores, Reguladores, Captadores de Posición, Visualizadores y Controles Numéricos.
- Personal:
 - 40 personas. Equipos de 1 a 14 personas
 - Alto nivel de formación, pero no en Informática
- Características del Software:
 - Muy amplio espectro: De plataforma hardware, de tipo de software, de esfuerzo de desarrollo.
 - Lenguajes C, C++ y ensamblador
 - Producto estándar
 - Larga duración del software

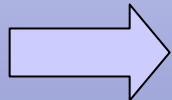
Diagnóstico



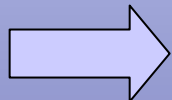
- Malas Especificaciones
- Diseños Frágiles
- Errores de Programación
- Falta de Testeo en I+D



Tardía detección de los errores



Formato y calidad del software dependiente de las personas

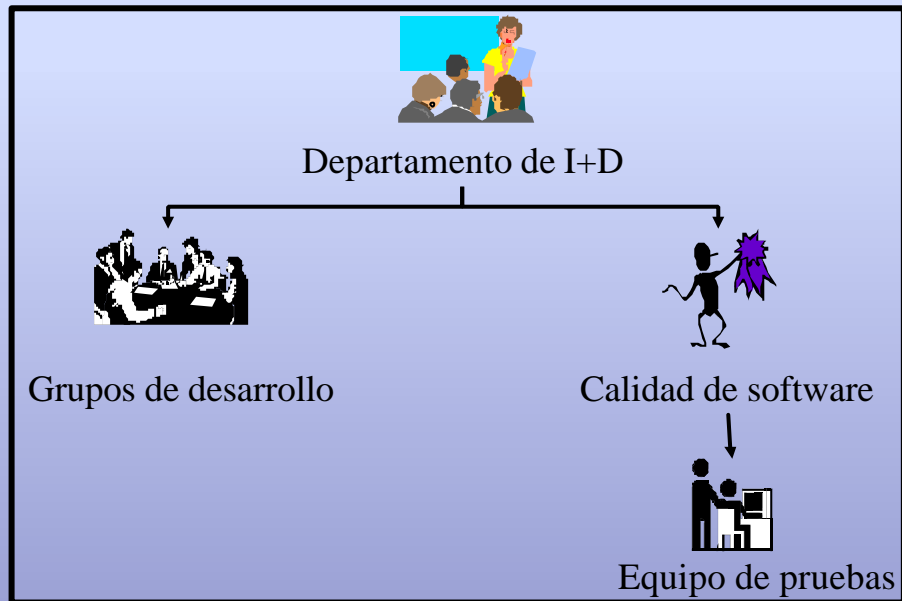


Dificultad en el mantenimiento

Iniciativa de Mejora

- Objetivos:

- Disminuir drásticamente el número de Errores
- Aumentar la Productividad



- Plan de actuación:

- Mejora del proceso de desarrollo
- Inclusión de actividades de calidad

- Enfoque:

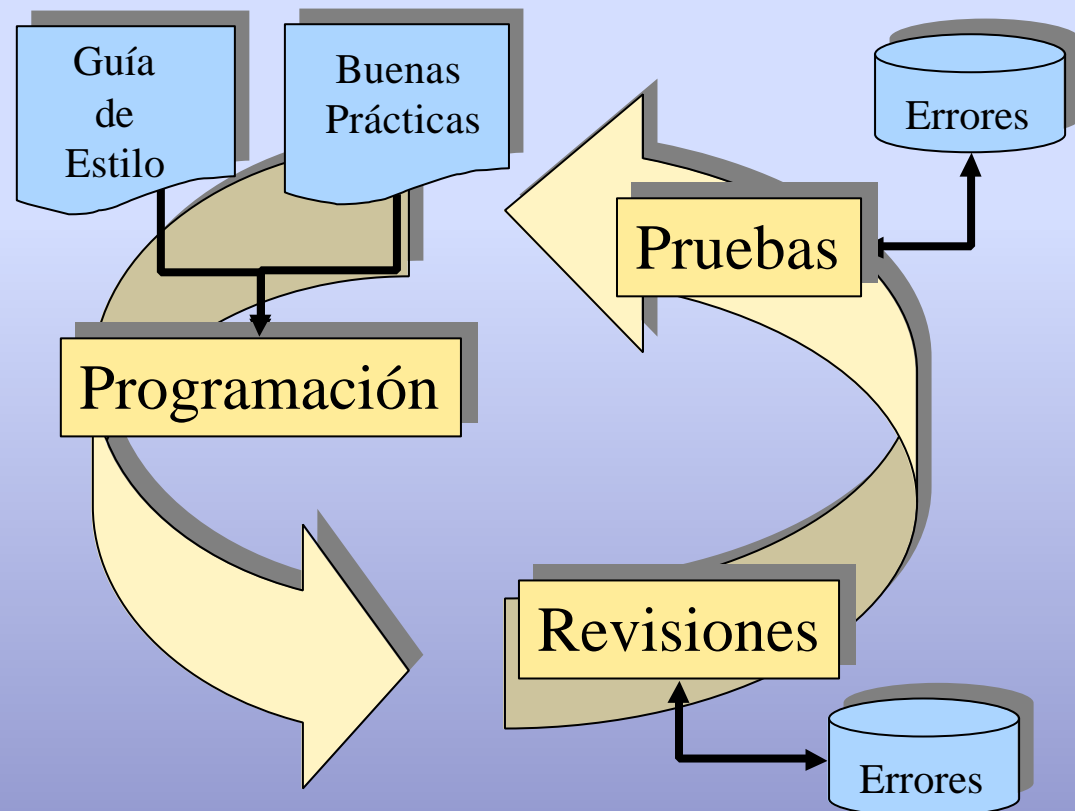
- Incremental
- En base a proyectos piloto

Mejoras en la Fase de Programación

- Estandarizar el código:
Guía de Estilo
- Formación: Buenas prácticas de Programación
- Revisar el código con el objetivo de detectar errores
- Aprender de los errores cometidos anteriormente

➔ Mejorar la calidad del código

Departamento de I+D



Mejoras en la Fase de Programación

Guía de estilo

- Aumentar la legibilidad
- Unificar el estilo entre los programadores
- Normas de obligado cumplimiento excepto para:
 - Código antiguo
 - Código externo

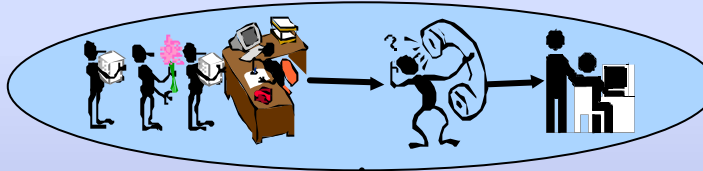
Buenas prácticas en la programación

- Instaurar un conjunto de buenas prácticas
- Mejorar la legibilidad del código
- Facilitar el mantenimiento

Revisiones de Código

- Basado en Fagan Inspections
- Detectar los errores pronto
- Compartir conocimiento entre los desarrolladores
- Garantizar aplicación Guía de Estilo y Buenas Prácticas

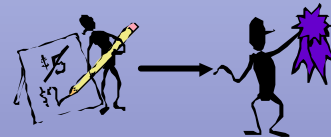
Preparación



Reunión de Revisión



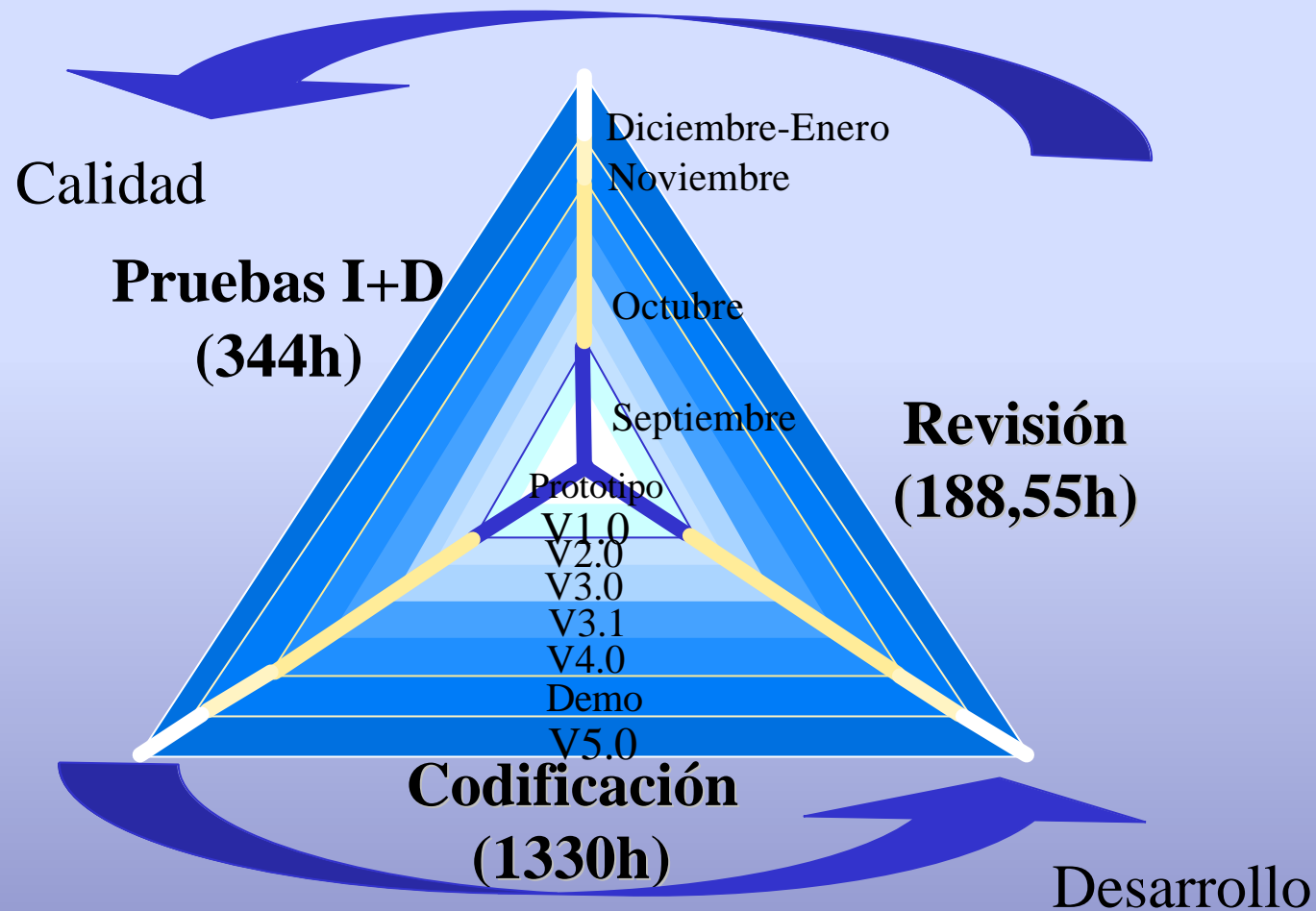
Corrección de errores



Pruebas

- Equipo dedicado en I+D
- Implicados en el proyecto desde la fase de requisitos
- Pruebas realizadas a las diferentes releases
- Feedback temprano a los desarrolladores

Proyecto Osciloscopio



Fase de Codificación

- El desarrollo incremental permite una mayor visibilidad sobre el progreso del producto
- Las revisiones y las pruebas de versiones intermedias se realizan en paralelo con el desarrollo:
 - Permite detectar errores pronto
 - Permite ir aprendiendo durante la ejecución del proyecto
 - Las actividades de calidad tienen menor impacto en el tiempo de desarrollo:
 - Revisiones de código + Pruebas = 141 días naturales (retraso del proyecto: 4,7 meses; si se realizan en paralelo 6 meses).

Revisiones de Código

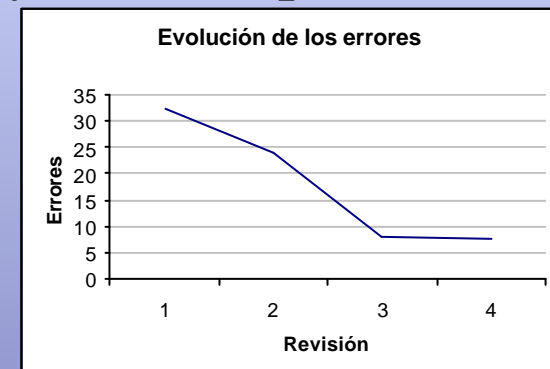
- 36 ficheros revisados (24.649 LOC)
- 10 personas involucradas
- 15 reuniones de revisión (3 re-revisiones)
- 188,55 horas
- 287 errores:
 - 123 errores menores (guía de estilo)
 - 164 errores mayores
- 110 sugerencias
- Efectividad: 1,52 errores por hora (2,11 incluidas sugerencias)
- Coste: 0,66 horas por error (0,47 incluidas sugerencias)

Revisiones de Código

- Las revisiones en las que han participado más de 4 personas han tenido un mayor coste pero no han detectado más errores. Cuatro personas resulta suficiente para realizar una revisión.
- El coste de las re-revisiones ha sido superior a la media. Salvo que el fichero cambie sustancialmente con una revisión informal (1 revisor) puede ser suficiente.
- Tiene que existir un revisor que se encargue de los errores de guía de estilo, de manera que el resto de revisores se dedique a detectar errores mayores.
- Es importante que el responsable de la revisión, compruebe los ficheros una vez corregidos los errores.
- Las revisiones de código han resultado ser una técnica útil para compartir experiencia entre programadores.

Revisiones de Código

- Las revisiones de código ayudan a estandarizar el código y a documentarlo de manera adecuada.
- La clasificación de errores propuesta inicialmente no ha resultado adecuada:
 - Resulta difícil clasificar los errores de las revisiones de código
 - No se pueden utilizar los mismos criterios de clasificación que en las pruebas
- El número de errores encontrados disminuye tras las primeras inspecciones (del código de un autor):
 - Por la realimentación
 - Se presta más atención al desarrollo

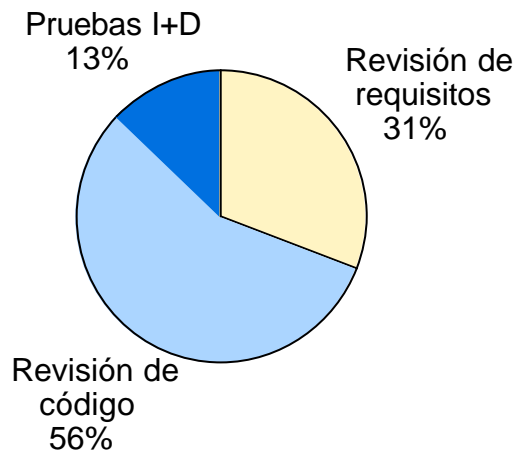


Pruebas en I+D

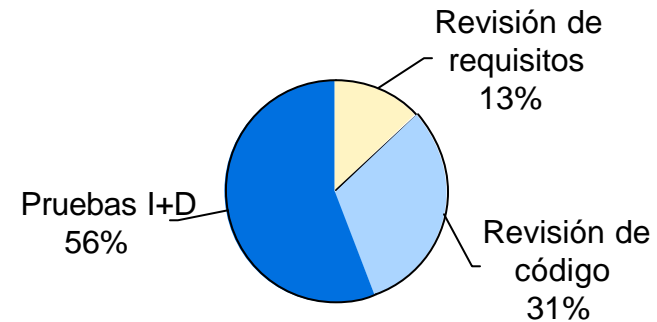
- 2 personas involucradas
- 344 h
 - Aprendizaje
 - Preparación de las pruebas
 - Ejecución
- 65 errores
 - 18 graves
 - 35 medios
 - 12 menores
- Efectividad: 0,19 errores por hora (0,86 ponderando los errores)
- Coste: 5,29 horas por error (1,16 ponderando los errores)

Actividades de Calidad

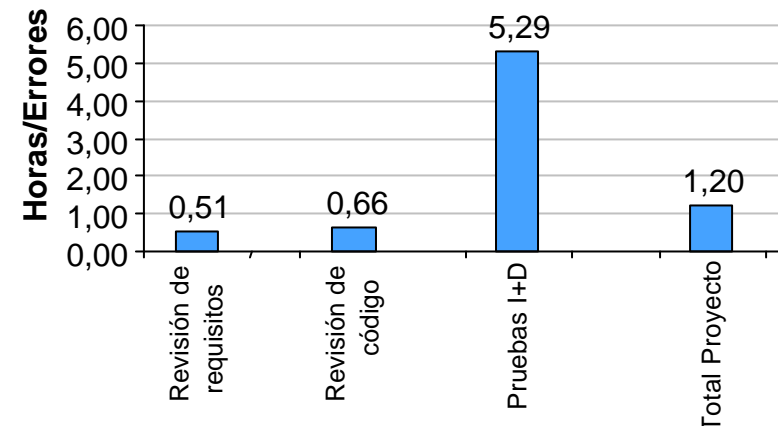
Q.3 Porcentaje de errores



Porcentaje de esfuerzo en actividades de calidad



Q.4 Coste de detección de un error



Actividades de Calidad

- El coste de detección de un error es mayor según avanza el ciclo de vida:
 - Realizar revisiones tempranas para detectar los errores cuanto antes
- Las revisiones de código y las pruebas resultan ser técnicas complementarias:
 - En general, detectan diferentes tipos de errores
- Resulta muy difícil establecer unos criterios para clasificar errores que resulten válidos para todas las actividades de calidad.

Líneas Futuras

- Determinar el punto en el cual se deben realizar las revisiones del código y las pruebas:
 - Compromiso entre estabilidad de los ficheros y realización lo antes posible de las actividades de calidad
- Identificación de criterios para la realización de las distintas iteraciones en el desarrollo
- Extender las revisiones y las pruebas en I+D a todo el código (también a mantenimientos)
- Mejorar el proceso de feedback
 - Hay que extraer información de los errores detectados en fases posteriores o en campo para mejorar la programación y las revisiones de código.