

LSE001-2016: Diplomado de Sistemas Embebidos para Automatización y Robótica

Dr. Casimiro Gómez González
Facultad de Electrónica, UPAEP
correo: casimiro.gomez@upaep.mx
Tel: 2225 640517

Otoño de 2016

Prólogo

El presente material ha sido elaborado en el laboratorio de sistemas embebidos UPAEP, y se ha desarrollado con la experiencia de estudiantes y profesores que han colaborado en dicho laboratorio. Hay material propio de clases y otro material generado a través de proyectos de vinculación y consultaría. Cualquier comentario o corrección favor de enviarlo por correo al autor.

El autor
Casimiro Gómez González
Doctor en Ingeniería Mecatrónica
correo: casimiro.gomez@upaep.mx

Índice general

1. Conceptos Básicos de Sistemas Embebidos	7
1.1. Definición formal de Sistemas Embebidos	11
1.2. Historia	11

Capítulo 1

Conceptos Básicos de Sistemas Embebidos

Los últimos diez años mas o menos, el mundo de la computación se ha movido desde las máquinas de escritorio grandes y estáticas a los dispositivos embebidos pequeños y móviles. Sistemas de software corriendo en redes de móviles, y los dispositivos embebidos deben tener propiedades que no siempre requieren los sistemas tradicionales:

- Rendimiento cerca del óptimo
- Robustez
- Distribución
- Dinamismo
- Movilidad

Una de las diferencias en la ingeniería de software para sistemas embebidos es el conocimiento adicional que el ingeniero tiene de potencia eléctrica y electrónica; interfaces físicas de electrónica analógica y digital con la computadora; y diseño de software para sistemas embebidos y procesamiento digital de señales.

Cerca del 95 % de los sistemas de software son actualmente embebidos. Considera los dispositivos que tienes en casa de uso diario:

- Teléfono Celular, iPod, microondas

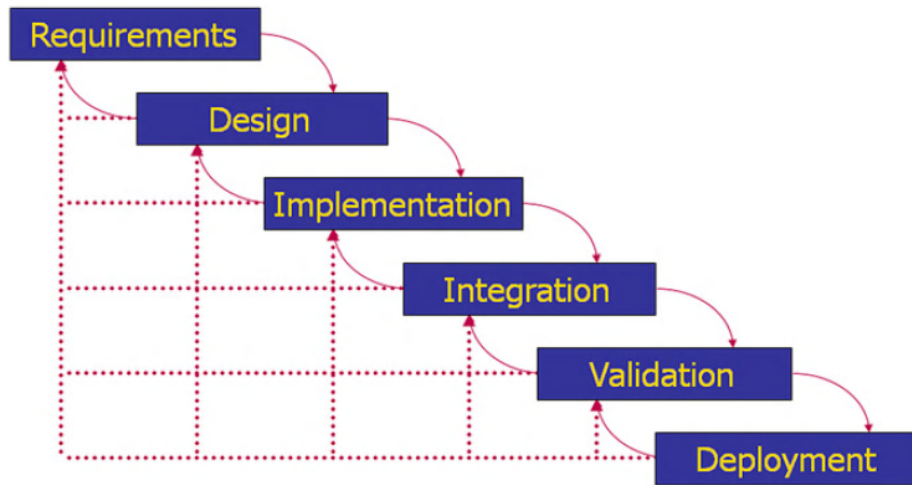


Figura 1.1: Modelo de desarrollo de software *Waterfall*

- receptor satelital de TV, receptor de TV de cable
- Unidad de control de automovil
- Reproductor de DVD

El desarrollo de software embebido usa los mismos modelos de desarrollo de software que las otras técnicas convencionales, incluyendo el modelo *Waterfall*, el modelo en *espiral* y el modelo *Agile*.

La fases principales para el desarrollo de sistemas embebidos se pueden describir:

1. **Definición del problema:** En esta fase se determina exactamente que quiere el cliente y el usuario. Esto incluye el desarrollo de un contrato con el cliente, dependiendo que tipo de producto esta siendo desarrollado. El objetivo de esta fase es especificar que producto de software se hará. Las dificultades incluyen: la solicitud del cliente por el producto incorrecto, el cliente no sabe sobre computación/software, lo cual limita la efectividad de esta fase, las especificaciones por lo regular son ambiguas, inconsistentes e incompletas.
2. **Arquitectura/diseño:** La arquitectura se refiere a la selección de los elementos de arquitectónicos, sus interacciones, y las restricciones de

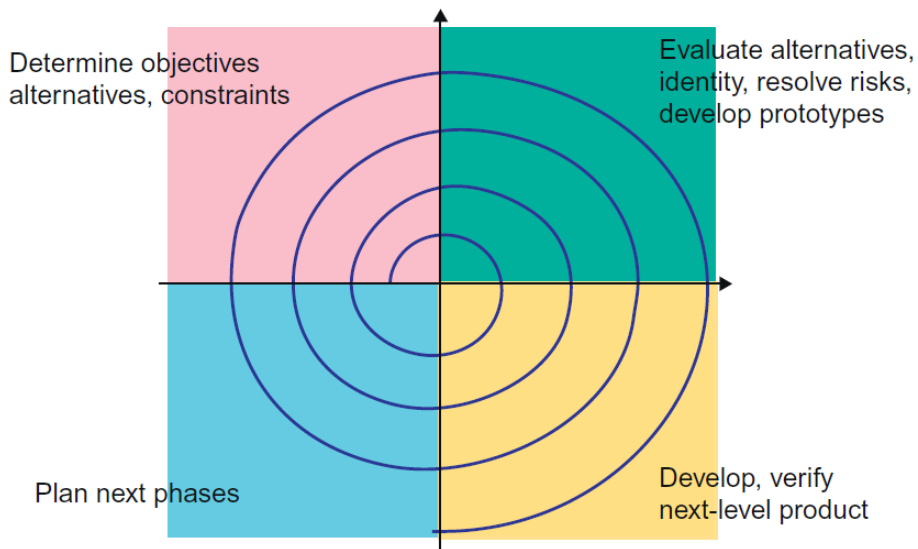


Figura 1.2: Modelo de desarrollo de software *Espiral*

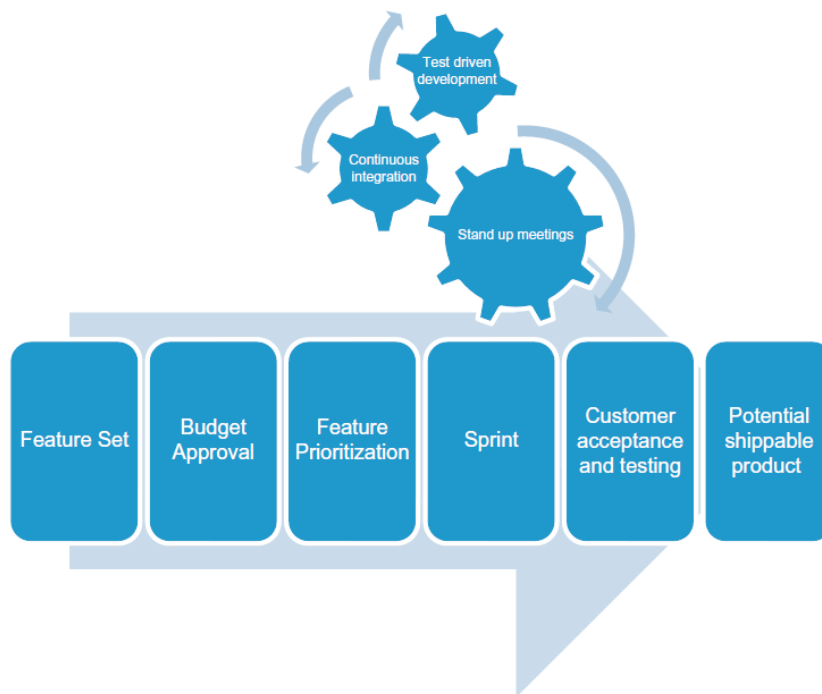


Figura 1.3: Modelo de desarrollo de software *Agile*

LSE001-2016

Elaboró: Dr. Casimiro Gómez González

aquellos elementos y sus interacciones necesarios para proporcionar un ambiente de trabajo con el cual satisfacer los requerimientos y servir como base para el diseño. El diseño se refiere a la modularización y detalle de interfaz entre los elementos de diseño, sus algoritmos y procedimientos, y los tipos de datos necesarios para soportar la arquitectura y satisfacer los requerimientos. Durante la fase de arquitectura y diseño, el sistema es descompuesto en módulos de software con interfaces. Durante el diseño, el equipo de software desarrolla las especificaciones de los módulos (algoritmos y los tipos de datos), manteniendo un récord de las decisiones de diseño y la trazabilidad, y especifica como los productos de software harán estas tareas. La dificultad principal durante esta fase incluye la falta de comunicación entre los diseñadores de módulos y terminan desarrollando un diseño que puede ser inconsistente, incompleto y ambiguo.

3. **Implementación:** Durante esta fase el equipo de desarrollo implementa los módulos y componentes y verifica que cumplan las especificaciones. Los módulos se combinan de acuerdo con el diseño. Las implementaciones especifican como los productos del software hacen su trabajo. Algunas de las dificultades principales incluyen los errores de interacción entre los módulos y los errores de integración que pueden influenciar la calidad y la productividad.
4. **Verificación y Validación(V&V):** Hay varias formas de V&V. Una forma es el “análisis”. El análisis puede ser en la forma de verificación estática, científica y formal y en revisiones informales. Las pruebas es una forma dinámica de V&V. Esta forma de pruebas viene en la forma de una caja blanca (se tiene acceso al código) y de caja negra (no hay acceso al código). Las pruebas pueden ser estructurales o de comportamiento. Hay pruebas estándares para realizar dependiendo del producto a desarrollar.

Mas y mas desarrollo de software para sistemas embebidos se esta moviendo al desarrollo basado en componentes. Este tipo de desarrollo es aplicado generalmente para componentes de tamaño razonable y reutilización a través del sistema, que es una tendencia creciente en sistemas embebidos. Los desarrolladores aseguran que estos componentes son adaptables a contextos variantes y extienden la idea mas allá del código a otros artefactos desarrollados también. Esta técnica cambia la ecuación desde “Integración, entonces

Bajar al Hardware” a “Bajar al Hardware, entonces Implementar”.

1.1. Definición formal de Sistemas Embebidos

Se trata de un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas.

1.2. Historia