Diseño de Sistemas Embebidos para Aplicaciones Ferroviarias

Ing. Sergio Gallina (UNCA)

Dr. Ing. Ariel Lutenberg (UBA, CONICET, ACSE)

AGENDA – parte 1

Dr. Ing. Ariel Lutenberg (UBA, CONICET, ACSE)

- **Contexto y motivación**
- **¿Por qué es tan díficil y valioso?**
- **Metodología y normativa**
- **Fases del ciclo de vida**
- **Eligiendo nuestro primer caso**

Contexto y motivación

- **El Proyecto CIAA busca desarrollar soluciones con alto valor agregado y capacitar en estos temas a estudiantes, docentes, profesionales e investigadores.**

- **Está en desarollo la CIAA-Safety y es atractivo usarla.**

- **Hay dos subsidios relacionados con esta cuestión:**

  - **PDTS CIN-CONICET Nº151** "Desarrollo de una Computadora Industrial Abierta Argentina (CIAA) para aplicaciones que requieran de seguridad funcional certificada", $200.000 (2016/2017).
  - **PDE UBA Nº23** "Controlador electrónico para barreras automáticas ferroviarias con nivel de integridad de seguridad certificable hasta SIL4". $75.000 (2016/2017)

¿Por qué es tan difícil y valioso?

- **El mal diseño de un sistema ferroviario puede causar cientos de victimas fatales y miles de heridos:**

Ciudad	Año	Fallecidos	Heridos	Descripción	¿Evitable?
Benavidez	1970	236	400	Choque de trenes	Si
Castelar	2013	3	315	Choque de trenes	Si
Flores	2011	11	228	Choque con omnibus	Si
Once	2012	51	702	Choque con fin de vía	Si

¿Por qué es tan difícil y valioso?

- **Cada accidente tiene magnitud considerable.**
- **Comparar con los accidentes por inhalación de monoxido de carbono, que en Argentina causan más de 200 victimas fatales cada año y cientos de heridos, pero rara vez involucran a más de 5 personas.**
- **Esta característica está ligada al concepto de "riesgo".**
- **La norma ferroviaria UNE EN 50126, que se introducirá en unos minutos, define al riesgo como:**

  "*La tasa probable de ocurrencia de un peligro que ocasione daño, y el grado de severidad de dicho daño*".

¿Por qué es tan difícil y valioso?

- **Los sistemas ferroviarios certificados que minimizan esos riesgos tienen un valor enorme:**
  - un controlador automático de barreras ya instalado puede costar hasta U$S 200.000.
- **Ninguna empresa argentina fabrica los componentes básicos certificados necesarios (todo es importado).**
- **En Argentina hay más de 13.000 Pasos a Nivel y sólo el 3% cuenta con un sistema automático de control de barreras, el 7% es manual y el 90%"nada".**
- **La instalación de las barreras automáticas necesarias demandaría importaciones por millones de dólares.**

Metodología y normativa

- **Para minimizar los riesgos existen normas que fijan metodologías.**

- **Algunas de las más utilizadas son las normas de la comunidad europea para aplicaciones ferroviarias: por ej. la serie UNE 50126, 50128, 50129, etc.**

  50126: Especificación y demostración de RAMS.
  50128: Software para control y protección del ferrocarril.
  50129: Sist. electrónicos de seguridad p/ la señalización.

- **Existen otras normas, como las estadounidenses FRA 49:236, AREMA, IEEE 730, 8XX, 1012, 10XX, 1483, 1558.**

- **Una de las cosas más importantes que definen es el ciclo de vida que debe utilizarse.**

Metodología y normativa

- La UNE 50126 define al ciclo de vida del sistema como:

  ”una secuencia de fases, que contienen tareas y abarcan la vida completa de un sistema desde su concepto inicial hasta su retiro y eliminación”.

  “El ciclo de vida proporciona una estructura para la planificación, la gestión, el control y la supervisión de todos los aspectos de un sistema, incluido la RAMS, a medida que el sistema avanza a través de sus fases, con el fin de entregar el producto adecuado al precio correcto dentro del plazo acordado.”

Metodología y normativa

- El término RAMS corresponde al acrónimo de:
  - Reliability (Fiabilidad): probabilidad de que un elemento pueda realizar una función requerida en condiciones determinadas durante un intervalo de tiempo determinado.
  - Availability (Disponibilidad): capacidad de un producto de hallarse en situación de realizar una función requerida en condiciones determinadas en un momento dado o durante un intervalo de tiempo señalado, suponiendo que se faciliten los recursos externos requeridos.

Metodología y normativa

- **El término RAMS corresponde al acrónimo de:**
  - **Maintianability (Mantenibilidad): probabilidad de que una acción dada de mantenimiento activo, correspondiente a un elemento en unas condiciones de utilización dadas, pueda ser llevada a cabo en un intervalo establecido de tiempo cuando el mantenimiento se realiza en condiciones establecidas y se utilizan procedimientos y recursos establecidos.**
  - **Safety (Seguridad): Ausencia de riesgo inaceptable de daño.**

Fases del ciclo de vida

- El ciclo de vida que propone la UNE 50126 es:

1. Concepto
2. Definición del sistema y Condiciones de aplicación
3. Análisis de riesgos
4. Requisitos del Sistema
5. Distribución de los Requisitos del Sistema
6. Diseño y Desarrollo
7. Producción
8. Instalación

Fases del ciclo de vida

- El ciclo de vida que propone la UNE 50126 es:

9. Validación del Sistema (incluyendo Aceptación de seguridad y Puesta en Servicio)
10. Aceptación del Sistema
11. Operación y Mantenimiento
12. Supervisión de Ejecución
13. Modificación, Realimentación
14. Retirada del servicio y Eliminación

Fases del ciclo de vida

- **Para cada una de las fases la norma indica:**
  - **Objetivos**
  - **Entradas**
  - **Requisitos**
  - **Entregas**
  - **Verificación**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Objetivos: “el objetivo de esta fase debe ser el de desarrollar un nivel de comprensión del sistema suficiente para permitir que todas las tareas siguientes del ciclo de vida RAMS se cumplan satisfactoriamente”.**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Entradas: “las entradas para esta fase deben incluir toda la información relevante y, en los casos en que proceda, los datos necesarios para cumplir los requisitos de la fase; por ejemplo, las declaraciones del alcance y la finalidad correspondientes al proyecto”.**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Requisito 1: "adquirir, en el contexto de la ejecución RAMS, un entendimiento de:**
    1. **El alcance, el contexto y la finalidad del sistema;**
    2. **El entorno del sistema, incluidos los:**
       - **aspectos físicos;**
       - **posibles problemas relacionados con la interfaz del sistema;**
       - **aspectos sociales;**
       - **aspectos políticos;**
       - **aspectos legislativos;**
       - **aspectos económicos.**
    3. **las implicaciones generales RAMS para el sistema."**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Requisito 2: “debe ser el de examinar:**
    1. **Las implicaciones RAMS para cualquier análisis financiero del sistema;**
    2. **Las implicaciones RAMS para cualesquiera estudios de viabilidad del sistema.”**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Requisito 3: “identificar fuentes de peligros que pudieran afectar el rendimiento RAMS del sistema, incluidas:**
    1. **La interacción con otros sistemas;**
    2. **La interacción con seres humanos.”**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Requisito 4: "obtener información acerca de:**
    1. **Anteriores requisitos RAMS y el anterior rendimiento RAMS en sistemas similares y / o relacionados;**
    2. **Fuentes identificadas de peligros para el rendimiento RAMS;**
    3. **La Política y los Objetivos de seguridad actuales de la Autoridad Ferroviaria;**
    4. **Legislación en materia de seguridad."**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Requisito 5: “ definir el alcance de los requisitos de gestión para sucesivas tareas RAMS del ciclo de vida del sistema”.**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Entregas: "los resultados de esta fase se deben documentar, junto con cualesquiera suposiciones y justificaciones realizadas durante la fase.**

    En las entregas se debe incluir una estructura de gestión adecuada para poner en práctica los requisitos RAMS correspondientes a las fases 2, 3 y 4 del ciclo de vida.

    Las entregas de esta fase son una aportación clave de información para sucesivas fases del ciclo de vida".

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Verificación: "las siguientes tareas de verificación se deben emprender dentro de esta fase:**
    1. **Evaluación de la idoneidad de la información y, cuando proceda, de los datos y otras estadísticas utilizadas como información aportada para tareas RAMS dentro de esta fase;**
    2. **Evaluación de la idoneidad de la declaración de entorno del sistema definida en el Requisito 1;**

Fases del ciclo de vida

- **Por ejemplo, para la Fase de Concepto:**
  - **Verificación: "las siguientes tareas de verificación se deben emprender dentro de esta fase:**
    3. **Evaluación de la integridad de la relación de fuentes de peligros definida en el Requisito 3;**
    4. **Ealuación de la idoneidad de los métodos, herramientas y técnicas utilizadas en la fase;**
    5. **Evaluación de la competencia de todo el personal que desempeñe tareas en la fase."**

Nuestro primer caso de estudio

- **En el marco del Proyecto UBA-PDE Nº23 se optó por realizar un primer proyecto relativamente simple en el área de las aplicaciones ferroviarias con los objetivos:**
  - **Aprendizaje:** adquirir conocimientos y experiencias de nivel básico e intermedio en relación con las aplicaciones ferroviarias de seguridad crítica.
  - **Educativos**: presentar un caso de estudio de acceso libre para formar a estudiantes y docentes en las metodologías para el desarrollo de sistemas críticos.
  - **Vinculación**: establecer relaciones con empresas, organismos e instituciones ligadas al sector ferroviario a partir del abordaje de un caso concreto de aplicación.

Nuestro primer caso de estudio

- **En este marco se propusieron tres posibles proyectos:**
  - **Sistema de hombre vivo:** consiste en un sistema electrónico que permite detectar si el conductor del ferrocarril está atendiendo adecuadamente a su tarea.

Nuestro primer caso de estudio

- **En este marco se propusieron tres posibles proyectos:**
  - **Sistema de hombre vivo:** consiste en un sistema electrónico que permite detectar si el conductor del ferrocarril está atendiendo adecuadamente a su tarea.
  - **Controlador de velocidad**: es un equipo que detiene el ferrocarril si este supera el límite máximo de velocidad establecido.
  - **Monitor de barreras**: es un sistema que determina el estado general de la barrera (posición del brazo, temperatura del motor, carga de la batería, etc.).

Nuestro primer caso de estudio

- **Para elegir el proyecto a realizar propusimos un índice:**

	Probabilidad de éxito	Impacto	Plazo	Esfuerzo requerido	Indicador PIPE
Hombre vivo	5	9	6	5	1350
Control de velocidad	4	8	4	3	384
Monitor de barreras	7	7	8	7	**2744**

- **El monitor a su vez está ligado al objeto del UBA-PDE Nº23.**

- **Lo aprendido con el Monitor podrá ser adaptado y utilizado en otros proyectos ferroviarios.**

Nuestro primer caso de estudio

- **El monitor de barreras:**
  - **Estado del brazo: ALTO, BAJO, TRANSICIÓN, ERROR.**
  - **Accionamiento electromecánico o hidráulico**, con levas indicadoras de posición.
  - **Si la energía eléctrica resulta interrumpida**, el brazo deberán adquirir la posición horizontal.
  - **Debe reportar el estado a la central.**
  - **Incluye batería auxiliar.**

Nuestro primer caso de estudio

- Para **más información** sobre la aplicación de la UNE 50126 al Monitor de Barreras ir a:

  http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=proyecto:casosdeuso:aplicacionescriticas:aplicacionesferroviarias:monitordebarreras:contextoymotivacion

- Están todos **invitados a colaborar** con esto!

¿Preguntas?

AGENDA – parte 2

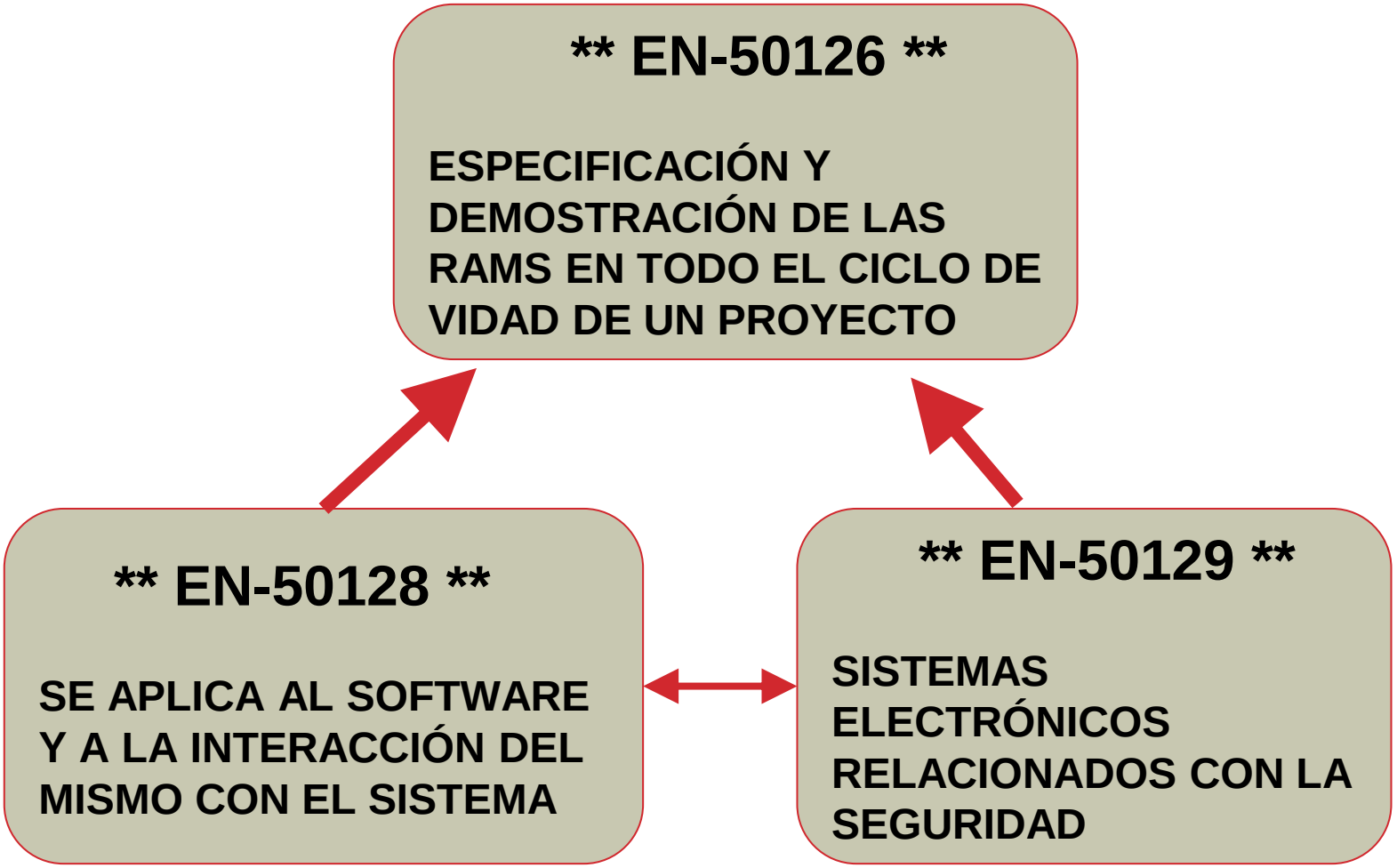
Ing. Sergio Gallina (UNCA)

- **Contexto de desarrollo de software bajo normas NE-5012x**
- **Pasos para la aplicación de la norma en el desarrollo del software**
  - **Personal**
  - **Especificación de Requisitos**
  - **Diseño e implementación**
  - **Verificación**
  - **Integración Hardware / Software**

SOFTWARE PARA SISTEMAS DE CONTROL BAJO NORMAS EN-50128

FASE 6 y 7 DE LA NORMA EN-50126

CONTEXTO


** EN-50126 **
ESPECIFICACIÓN Y DEMOSTRACIÓN DE LAS RAMS EN TODO EL CICLO DE VIDAD DE UN PROYECTO
** EN-50128 **
SE APLICA AL SOFTWARE Y A LA INTERACCIÓN DEL MISMO CON EL SISTEMA
** EN-50129 **
SISTEMAS ELECTRÓNICOS RELACIONADOS CON LA SEGURIDAD

GENERALIDADES EN-50128

- **Se centra en los métodos para desarrollar software**
- **Esta norma dio lugar a una norma genérica CEI 61508**
- **Identifica 5 niveles de integridad, siendo el nivel 0 el mínimo o aquel software que no hace referencia a la integridad y 4 el máximo**
- **El estado actual del arte es tal que ni la aplicación de métodos para garantizar la calidad, ni la aplicación de técnicas de software tolerante a fallos pueden garantizar seguridad absoluta**

GENERALIDADES EN-5012x

Al aplicar la norma se pretende

- **Dejar evidencia de la gestión de la calidad**
- **Dejar evidencia de la gestión de la seguridad**
- **Dejar evidencia de la seguridad funcional y técnica**

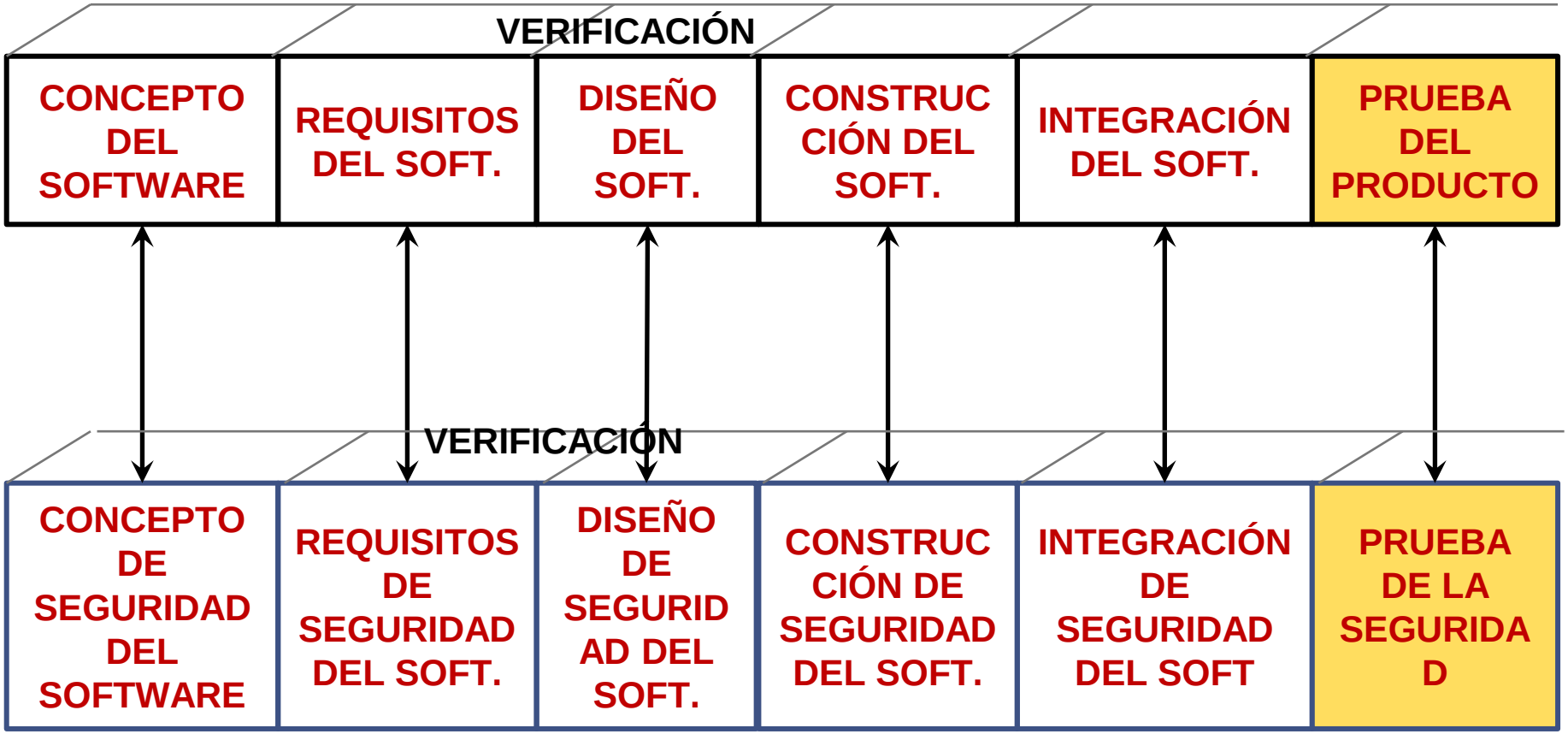
PRINCIPIOS DE DESARROLLO ACEPTADOS

- **Diseño descendente top-down**
- **Modularidad**
- **Verificación de cada fase del ciclo de vida de desarrollo**
- **Módulos y librerías verificados**
- **Documentación clara y auditable**
- **Ensayos de validación**
- **OTROS**

PASOS PARA LA APLICACIÓN DE LA NORMA

1. **Definir la especificación de requisitos de seguridad y en paralelo considerar la arquitectura del software (Cap 5, 8 y 9)**
2. **Diseño, desarrollo y ensayo del software de acuerdo al plan de aseguramiento de la calidad, al nivel de integridad de seguridad y al ciclo de vida del software (Cap 10)**
3. **Integrar el software en el hardware específico (Cap 12)**
4. **Validar el software (Cap 13)**
5. **Mantenimiento (Cap 16)**

PASOS PARA LA APLICACIÓN DE LA NORMA



ESPECIFICACIÓN DE REQUISITOS DE SEGURIDAD

Se deben generar especificaciones de

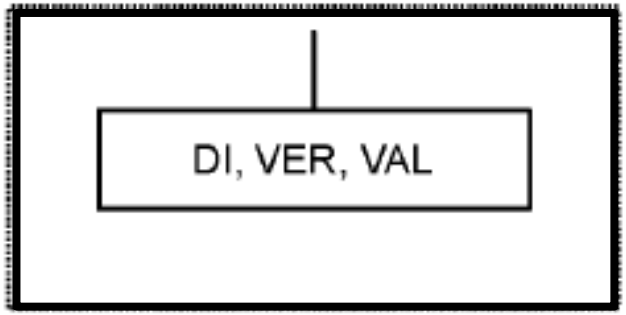
- **Requisitos del sistema**
- **Requisitos de la seguridad**
- **Requisitos de la arquitectura y el plan de seguridad**

NIVEL DE INTEGRIDAD DE SEGURIDAD DEL SOFTWARE	DESCRIPCIÓN DE LA INTEGRIDAD DE SEGURIDAD DEL SOFTWARE
4	MUY ALTA
3	ALTA
2	MEDIA
1	BAJA
0	No relacionado con la seguridad

PERSONAL Y RESPONSABILIDADES

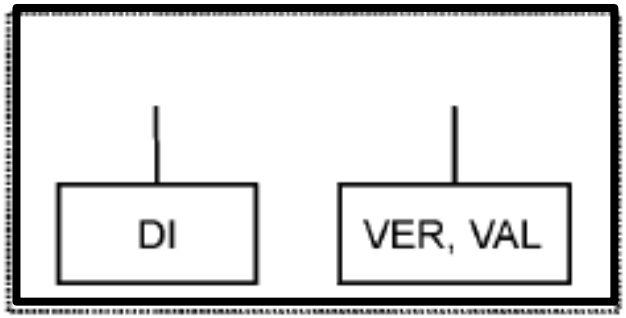
- **El proveedor y el responsable de desarrollo deben implementar partes relevantes de la ISO 9001**
- **Se debe incluir evidencia de la experiencia**
- **El verificador y el validador puede ser la misma persona pero no el desarrollador y no deben depender del director del proyecto**

NIVEL 0



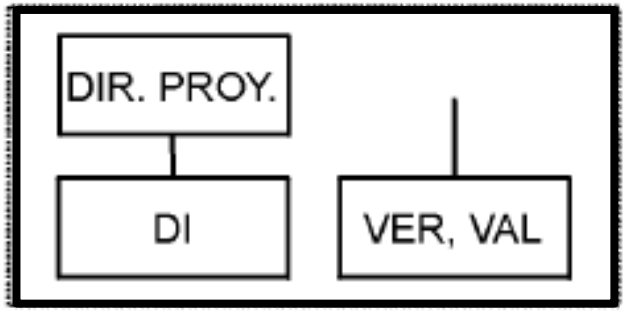
EVALUADOR

NIVELES 1 Y 2



EVALUADOR

NIVELES 3 Y 4



EVALUADOR

ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE

Para poder especificar los requisitos del software se requiere

- **Requisitos del sistema**
- **Requisitos de seguridad del sistema**
- **Descripción de la arquitectura del sistema**
- **PLAN DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE**

Documentos de salida (entregables)

- **Especificación de requisitos del software**
- **Especificación de ensayos de requisitos del software**

ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	Métodos Formales, incluyendo por ejemplo CCS, CSP, HOL, LOTOS, OBJ, Lógica Temporal, VDM Z y B	B.30	-	R	R	AR	AR
2	Métodos Semi-Formales	D.7	R	R	R	AR	AR
3	Metodología Estructurada, incluyendo por ejemplo JSD, MASCOT, SADT, SDL, SSADM y Yourdon	B.60	R	AR	AR	AR	AR

La especificación de requisitos, requiere siempre la descripción en lenguaje natural y la notación formal (matemática) que refleje tal especificación

ARQUITECTURA DEL SOFTWARE

Deberá cumplir con las especificación de requisitos del software para el nivel de seguridad especificado

Se deben identificar todos los componentes del software y verificar

→ si son propios o de terceros

→ si han sido validados

→ si son nuevos o reutilizados

ARQUITECTURA DEL SOFTWARE

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	Programación Defensiva	B.15	–	R	R	AR	AR
2	Diagnosis y Detección de Defectos	B.27	–	R	R	AR	AR
3	Códigos de Corrección de Errores	B.20	–	–	–	–	–
4	Códigos de Detección de Errores	B.20	–	R	R	AR	AR
5	Programación con Detección de Fallos	B.25	–	R	R	AR	AR
6	Técnicas de Bolsa de Seguridad	B.54	–	R	R	R	R
7	Programación con Diversidad	B.17	–	R	R	AR	AR
8	Recuperación en Bloque	B.50	–	R	R	R	R
9	Recuperación Regresiva	B.5	–	NR	NR	NR	NR
10	Recuperación Progresiva	B.32	–	NR	NR	NR	NR
11	Mecanismos de Recuperación de Fallos por Reintento	B.53	–	R	R	R	R
12	Memorización de Casos Ejecutados	B.39	–	R	R	AR	AR
13	Inteligencia Artificial – Corrección de Fallos	B.1	–	NR	NR	NR	NR

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

Obtener un software capaz de ser analizado y verificado

Integración del software

Finalizada esta fase se debe disponer de

- **Documentación**
- **Código fuente**
- **Resultado de la verificación (ensayos)**

Mantener al mínimo el tamaño y la complejidad del software

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	Métodos Formales incluyendo por ejemplo CCS, CSP, HOL, LOTOS, OBJ, Lógica Temporal, VDM, Z y B	B.30	–	R	R	AR	AR
2	Métodos Semi-Formales	D.7	R	AR	AR	AR	AR
3	Metodología Estructurada incluyendo por ejemplo JSD, MASCOT, SADT, SDL, SSADM y Yourdon	B.60	R	AR	AR	AR	AR
4	Aproximación Modular	D.9	AR	O	O	O	O
5	Estándares de Diseño y Codificación	D.1	AR	AR	AR	O	O
6	Programas Analizables	B.2	AR	AR	AR	AR	AR
7	Lenguaje de Programación Fuertemente Tipado	B.57	R	AR	AR	AR	AR
8	Programación Estructurada	B.61	R	AR	AR	AR	AR
9	Lenguaje de Programación	D.4	R	AR	AR	AR	AR
10	Subconjunto del Lenguaje	B.38	–	–	–	AR	AR
11	Traductor Validado	B.7	R	AR	AR	AR	AR
12	Traductor Probado por el Uso	B.65	AR	AR	AR	AR	AR
13	Librería de Módulos y Componentes Comprobados/Verificados	B.40	R	R	R	R	R

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	Existencia de normas de Codificación	B.16	AR	AR	AR	AR	AR
2	Pautas para el Estilo de Codificación	B.16	AR	AR	AR	AR	AR
3	Ausencia de Objetos Dinámicos	B.16	–	R	R	AR	AR
4	Ausencia de Variables Dinámicas	B.16	–	R	R	AR	AR
5	Uso Limitado de Punteros	B.16	–	R	R	R	R
6	Uso Limitado de Recursividad	B.16	–	R	R	AR	AR
7	Ausencia de Saltos Incondicionales	B.16	–	AR	AR	AR	AR

Finalidad:

- Estructurar el código y la documentación.
- Evitar las personalizaciones

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

Lenguaje de programación adecuado

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	ADA	B.62	R	AR	AR	R	R
2	MODULA-2	B.62	R	AR	AR	R	R
3	PASCAL	B.62	R	AR	AR	R	R
4	Fortran 77	B.62	R	R	R	R	R
5	'C' o C++ (sin restricción)	B.62	R	–	–	NR	NR
6	Subconjunto de C o C++ con normas de codificación	B.62 B.38	R	R	R	R	R
7	PL/M	B.62	R	R	R	NR	NR
8	BASIC	B.62	R	NR	NR	NR	NR
9	Ensamblador	B.62	R	R	R	–	–

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

¿Que se debe evitar?

- **Saltos incondicionales a excepción de llamadas a subrutinas**
- **Recursividad**
- **Punteros o cualquier objeto dinámico**
- **Declaración / Inicialización implícita de variables**

La programación en assembler no está bien considerada debido a su fuerte orientación al hardware

VERIFICACIÓN Y ENSAYO

Es obligatoria la creación de un plan de verificación del software

- **Selección de estrategias y técnicas**
- **Seleccion y utilizacion de equipos**
- **Selección de la documentación**
- **Evaluación de los resultados**
- **Evaluación de la fiabilidad del sistema**
- **El grado de cobertura que se logró en el ensayo**

Se debe poder demostrar (y quedar documentado) que se cumple con la fiabilidad, prestaciones y seguridad.

VERIFICACIÓN Y ENSAYO

TÉCNICA/MEDIDA		Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1	Ensayo Formal	B.31	–	R	R	AR	AR
2	Ensayos Probabilísticos	B.47	–	R	R	AR	AR
3	Análisis Estático	**D.8**	–	AR	AR	AR	AR
4	Análisis y Ensayos Dinámicos	**D.2**	–	AR	AR	AR	AR
5	Evaluaciones	B.42	–	R	R	R	R
6	Matriz de Trazabilidad	B.69	–	R	R	AR	AR
7	Análisis de Efectos de Errores software	B.26	–	R	R	AR	AR

1. Análisis de flujo de control
2. Análisis de flujo de datos
3. Ensayos y revisiones de diseño

1. Casos de ensayo a partir de valores extremos
2. Modelado de prestaciones
3. Ensayos basados en la estructura

INTEGRACIÓN HARDWARE / SOFTWARE

Demostrar que el hardware y el software interactúan correctamente

Durante esta etapa se debe

Confeccionar un plan de ensayos

Documentar el resultado de su ejecución.

INTEGRACIÓN HARDWARE / SOFTWARE

TÉCNICA/MEDIDA	Ref	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1 Ensayos Funcionales y de Caja-negra	**D.3**	AR	AR	AR	AR	AR
2 Ensayos de Prestaciones	**D.6**	–	R	R	AR	AR

Requisitos:

1 Para el nivel de integridad de seguridad del software 0, la técnica 1 deberá ser la aprobada.

2 Para los niveles de integridad de seguridad del software 1, 2, 3 ó 4, la combinación de técnicas aprobada deberá ser 1 y 2.

1. Tiempos de respuesta y limitaciones de memoria
2. Ensayo de avalancha / estres
3. Requisitos de prestaciones

1. Análisis de valores extremos
2. Simulación de procesos
3. Prototipado / Animación

MANTENIMIENTO

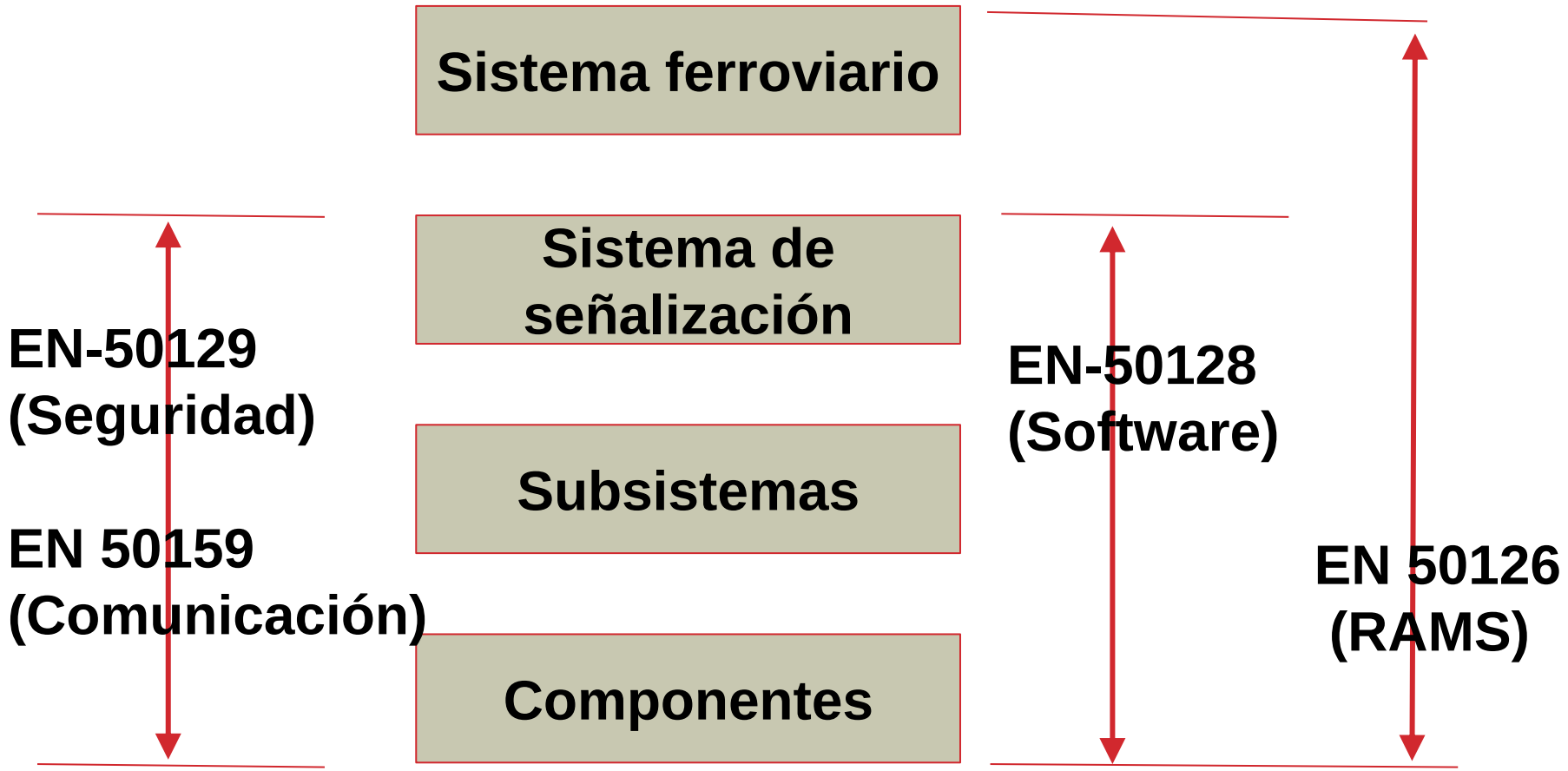
El mantenimiento debe ser tal que después de esta tarea el software se comporta en la forma requerida, preservando el nivel de integridad de la seguridad y la confiabilidad

Incluye:

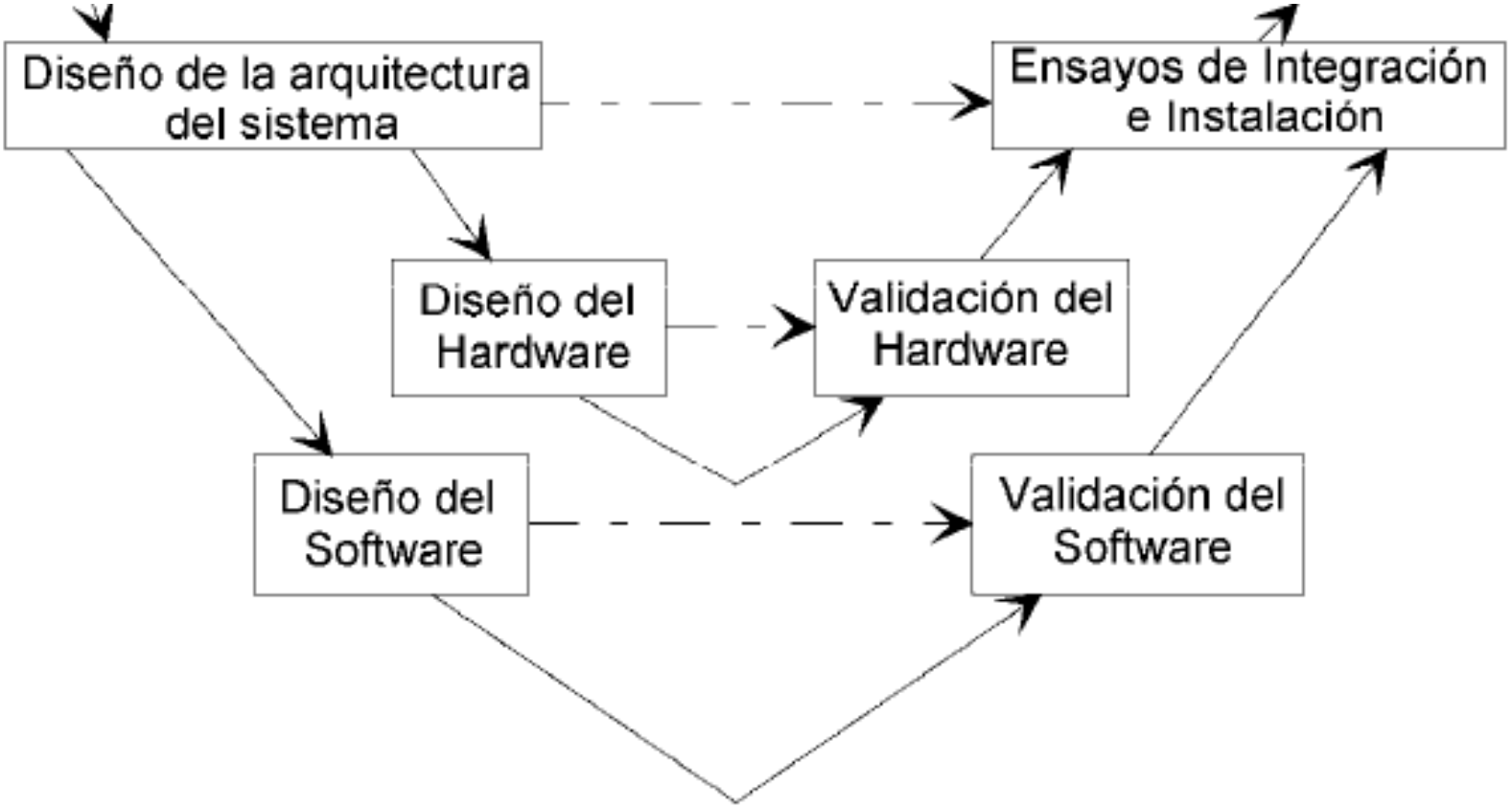
- **Correcciones**
- **Mejoras**
- **Adaptaciones**

Como mínimo los mantenimientos se deben llevar a cabo de acuerdo a las normas ISO 9000-3

CASO ESPECIAL DE LOS SISTEMAS DE SEÑALIZACIÓN



CASO ESPECIAL DE LOS SISTEMAS DE SEÑALIZACIÓN



BENEFICIOS - RIESGOS

- **El desarrollo de software crítico es más caro**

- **Pero en determinadas actividades hay que decidir**

¿Qué riesgos se asumen y quien los asume si no se adoptan las medidas necesarias?

- **Por otro lado, como usuarios**

¿Sabemos el riesgo que asumimos al usar o depender de un sistema?

¿Preguntas?

Muchas gracias!