Instituto
Nacional
de Tecnología
Industrial

Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC

Jorge Cogno
jac@inti.gob.ar

`roadmap` de la presentación

- Algunas de nuestras preguntas 'ingenuas' y el título de esta presentación
- Cuales son las alternativas para la validación del software embebido en dispositivos de medición?
- Validación de proceso software o de producto software? Aportes de ambas visiones al *'qué'* y *'cómo`*
- Alcances y limitaciones de MID y OIML TC5
- Herramientas de validación según WELMEC
- Modelos y clases de riesgo
- Aporte del framework CMMi para la opción `H`: validación del sistema de gestión de la calidad
- conclusiones

- Preguntas ‘ingenuas’
  1. Validación de software relevante desde el punto de vista de la metrología legal. Validación de producto o de proceso? *(separación de software? En ese caso, cómo se diseñan las interfaces de software que aseguren esa separación?)*
  2. La aprobación de modelo requiere que el NB disponga del código fuente del software relevante desde el punto de vista de la metrología legal?
  3. Cómo asegurar que, en la verificación primitiva, el software relevante desde el punto de vista de la metrología legal corresponde exactamente al de la aprobación de modelo? *(y es necesario ser tan estrictos?)*
  4. Cómo asegurar que, con posterioridad a la verificación primitiva, el software no ha sido intervenido?
  5. Cómo asegurar la integridad y autenticidad de datos metrológicamente relevantes transmitidos por sensores remotos? *(entonces, la separación de software debe incluir también la separación de datos? Y qué sucede con la integridad de los parámetros operativos del software de los dispositivos?)*

Esta presentació
conferencias Dr.
Grottker (PTB)
27/08/08

Dr. Zisky (PTB)
y 28/08/08

PKI y Criptograf
curva elíptica

Análisis de casos
prácticos MsC.
Benavídez
(CENAM) 28/08

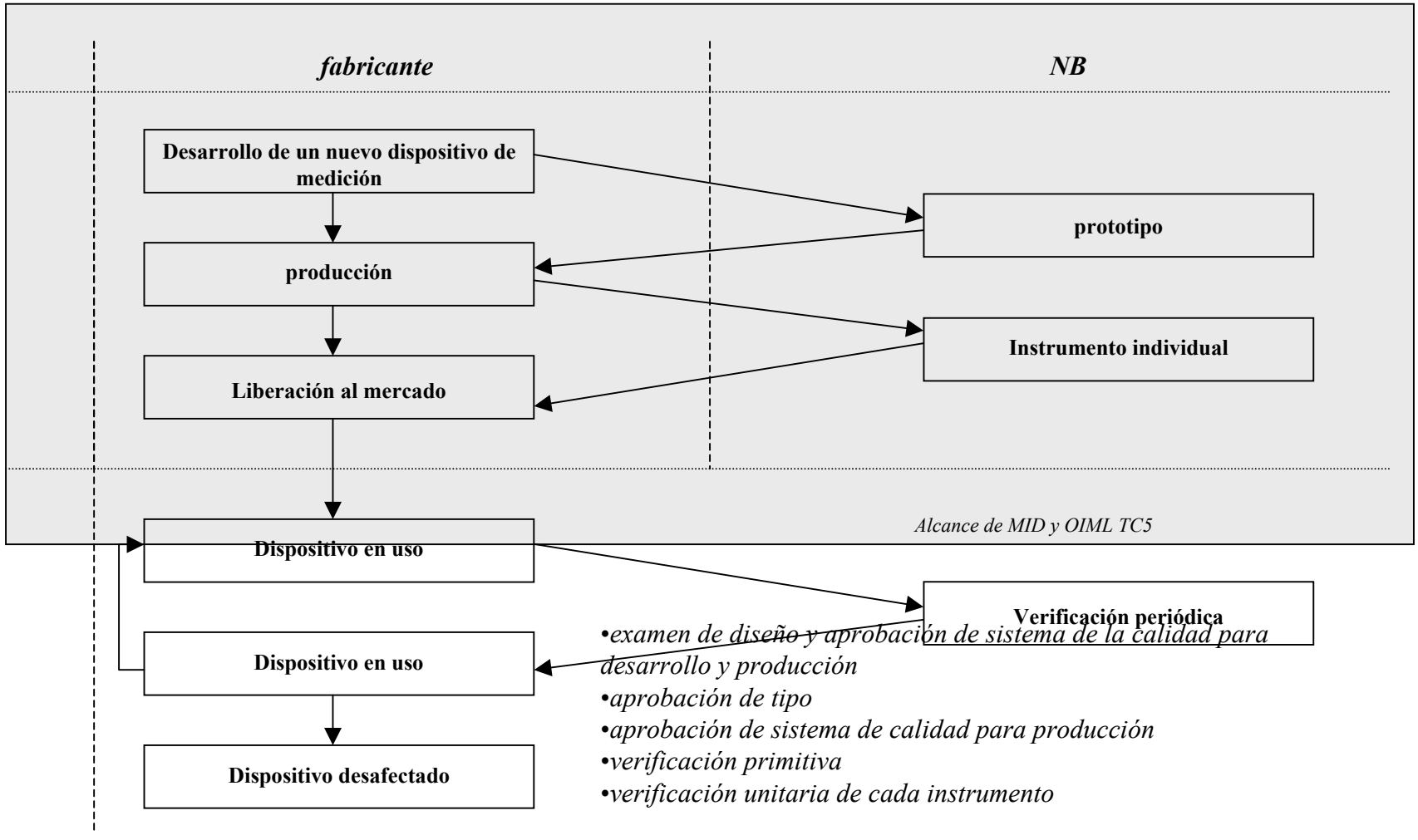
El software puede ser evaluado como un producto final o a través del proceso por el cual se lo desarrolla y mantiene
Exégesis del 'dogma CMMi´: debemos esperar que el producto software no sea mejor que el proceso
(existen heréticos: XProgramming, Agile)

Las llamadas herramientas de validación definidas para el software embebido
que vamos a analizar de inmediato estarían (a primera vista) centradas en el producto.
Este enfoque sería consistente con el enfoque tradicional de los procedimientos de aprobación por parte de laboratorios de ensayos independientes, que ponen el mayor peso en la demostración
de exactitud de medición y tolerancia a la sobrecarga del equipo en su conjunto.
Esta es una visión pragmática que parte de la propia definición de validación
(demostración de aptitud para el uso previsto)

Sin embargo …

La implementación cada vez mayor de funcionalidades del equipo de medición
a través de software aumenta su complejidad.
Es difícil asegurar que un instrumento de medición relativamente complejo
haya sido puesto a prueba en todas sus posibles configuraciones
Por lo tanto, deberíamos considerar volver (con reparos) al `dogma´, y decir que
un proceso de desarrollo bien definido suplementaría un resultado de ensayo,
que nunca tendrá una cobertura del 100% de todos los casos imaginables

Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC



Foco en el proceso	Foco en el producto

Measurement Instruments Directive (MID) Módulos para evaluación de conformidad

Módulo D Declaración de conformidad de tipo basada en aseguramiento de la calidad del proceso de producción

en ensayo

Módulo H Declaración de conformidad basada en un completo aseguramiento de la calidad

Módulo H1 Declaración de conformidad basada en un completo aseguramiento de la calidad más examen de diseño

Cláusula ISO/IEC 90003

Aplicables a nivel 'global' de la organización

4. Sistema de gestión de la calidad
5. Responsabilidad de la dirección
6. Gestión de los recursos

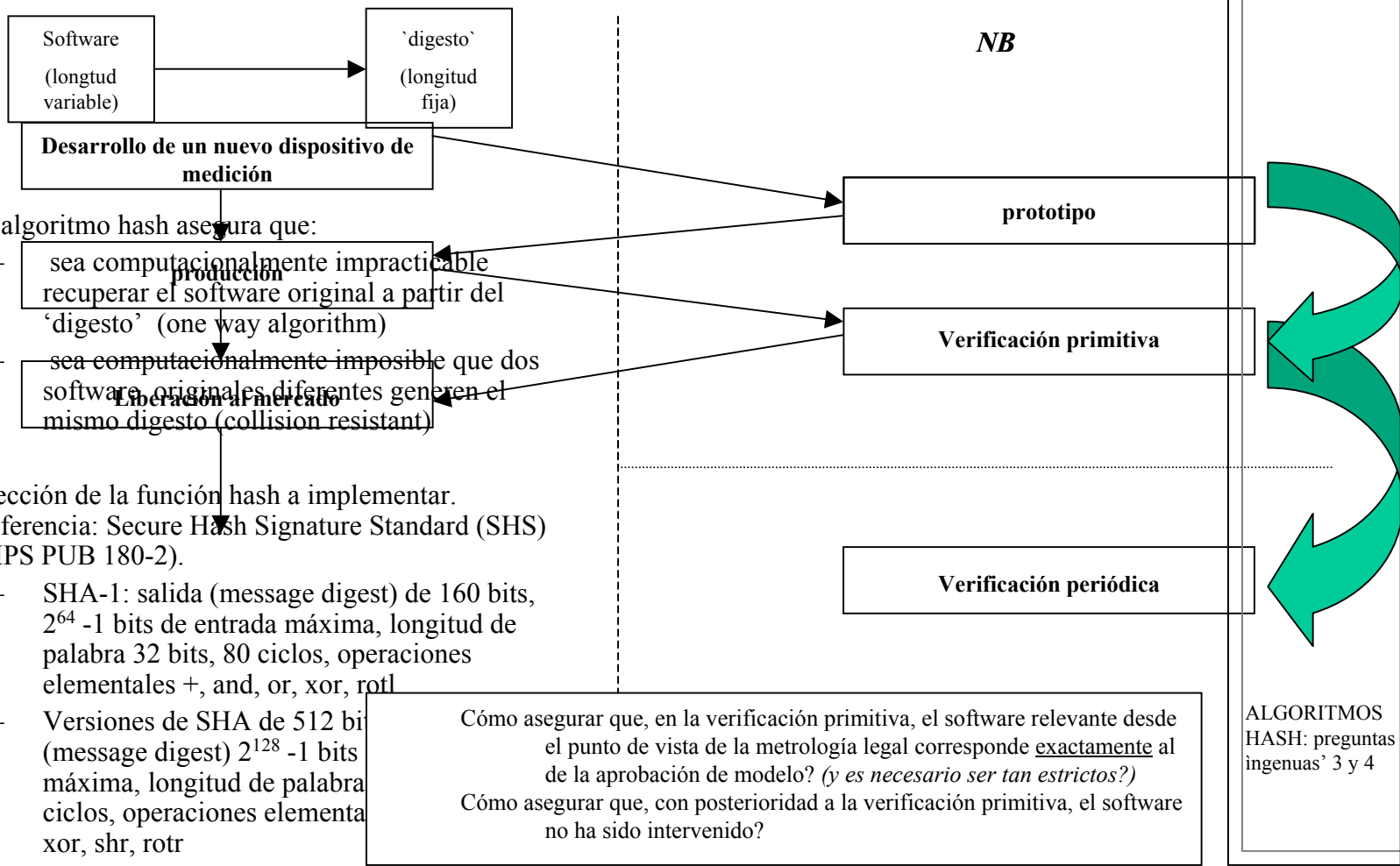
Aplicables a nivel proyecto / producto

7. Realización del producto

7.3.7 Control de cambios de diseño y desarrollo
7.5.3 Identificación y trazabilidad
7.2.1 Determinación de requerimientos relacionados con el producto
7.3.2 Inputs de diseño y desarrollo
7.3.3 Outputs de diseño y desarrollo
7.3.5 Verificación de diseño y desarrollo
7.4.3 Verificación de producto adquirido
7.6 Control de dispositivos de medición
7.3.6 Validación de diseño y desarrollo
7.3.6.2 Testing

8. Medición, análisis y mejora

8.2.4 Medición de producto
8.4 Análisis de datos
8.2.1 Satisfacción de cliente
8.2.2 Auditoría interna
8.2.3 Monitoreo y medición de procesos
8.3 Control de producto no conforme
8.5.1 Mejora continua
8.5.2 Acciones correctivas
8.5.3 Acciones preventivas



- El algoritmo hash asegura que:
  - sea computacionalmente impracticable recuperar el software original a partir del 'digesto' (one way algorithm)
  - sea computacionalmente imposible que dos software originales diferentes generen el mismo digesto (collision resistant)
- Elección de la función hash a implementar. Referencia: Secure Hash Signature Standard (SHS) (FIPS PUB 180-2).
  - SHA-1: salida (message digest) de 160 bits, 2^{64} -1 bits de entrada máxima, longitud de palabra 32 bits, 80 ciclos, operaciones elementales +, and, or, xor, rotl
  - Versiones de SHA de 512 bi (message digest) 2^{128} -1 bits máxima, longitud de palabra ciclos, operaciones elementa xor, shr, rotr

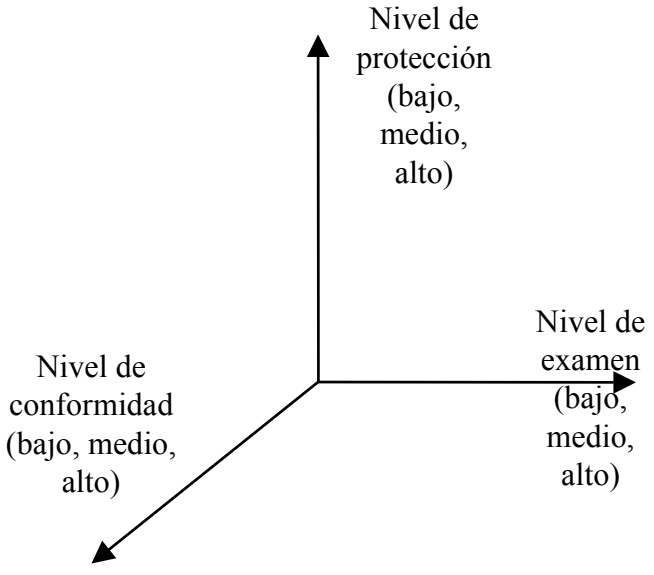
Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC

Instituto Nacional de Tecnología Industrial
INTI

Foco en el proceso	Foco en el producto
ISO 90003: Software Engineering – Guidelines for the Application of ISO 9001:2000 to computer software CMMI + SCAMPI: Capability Maturity Model Integration + Standard CMMI Appraisal Method for Process Improvement ISO/IEC 15504 (SPICE): Software Process Improvement and Capability Determination	ISO/IEC 9126-1: 2001. Software engineering- Product quality- ISO/1EC 14598-1: 1999, Information technology – Software product evaluation – ISO/IEC 25051.2:2006, Software engineering – Software product quality requirements and evaluation- SQUARE- Requirements for quality of commercial off-the-shelf (COTS)

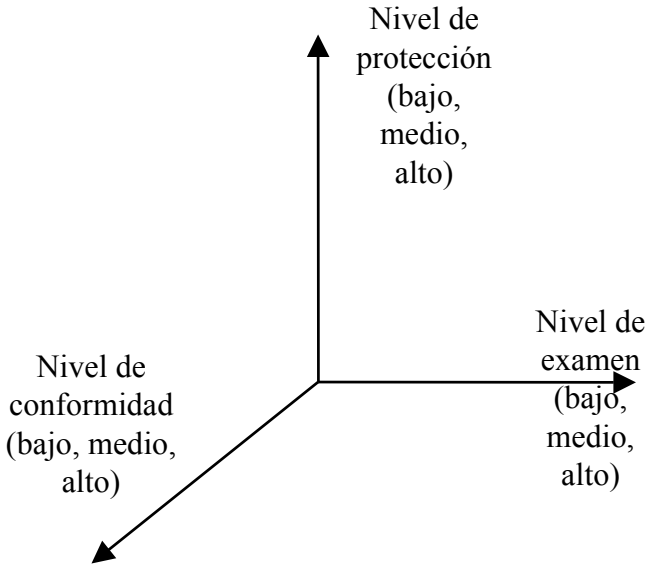
WELMEC

	CIWT Code Inspection and Walkthrough	**SMT** Software Module Testing	**AD** Analysis of Documentation and Specification and Validation of the Design	**VFTM** Validation by Functional Testing of the Metrological Functions	**VFTSw** Validation by Functional Testing of the Software Functions	**DFA** Metrological Data Flow Analysis
Aplicabilidad	Alto requerimiento de examen	Alto requerimiento de examen	siempre	Demostración de correctitud de algoritmos	Alto requerimiento de protección	• Alto requerimiento de conformidad
Característica	Estática (no requiere ejecución de código)	Dinámica (requiere ejecución de código)	Estática (no requiere ejecución de código)	Dinámica (requiere ejecución de código)	Dinámica (requiere ejecución de código)	Estática (no requiere ejecución de código)
Precondición	**Código fuente** (sólo software legalmente relevante)	**Código fuente** `testing bed` (compiladores, juegos de datos de prueba)	Documentación de fabricante: especificación de funciones accesibles externamente, interfaces, y soporte criptográfico	Manuales operativos, patrones de calibración, instrumentos de medición	Manuales operativos	**Código fuente**
Conclusiones	Implementación de algoritmos compatible con documentación de software y de acuerdo con requerimientos?	Implementación de algoritmos de acuerdo con requerimientos?		Resultados dentro de MPE?	Protección adecuada (parámetros operativos, datos almacenados)?	Validación de separación de software?



Grado de examen de software
Bajo: aprobación de tipo en base a la prueba funcional del instrumento. No se requiere prueba adicional del software.
Medio: Además del requisito anterior, se examina el software en base a su documentación, incluyendo descripción funcional y descripción de parámetros operativos. Se realizan pruebas (spot checks) de funciones soportadas por software para verificar la plausibilidad de la documentación y la efectividad de las medidas de protección.
Alto: Además de los requisitos anteriores, se realiza una prueba en profundidad del software, basada en el conocimiento del código fuente

Grado de protección de software
Bajo: no se requieren medidas de protección contra cambios intencionales
Medio: se protege el software contra cambios intencionales efectuados utilizando herramientas simples, como por ejemplo editores de texto
Alto: se protege el software contra cambios intencionales efectuados utilizando herramientas sofisticadas (debuggers, ingeniería reversa)



Grado de protección de software	Grado de examen de software	Grado de conformidad de software	Riesgo clase …
bajo	bajo	bajo	A
medio	medio	bajo	B
medio	medio	medio	C
alto	medio	medio	D
alto	alto	medio	E
alto	alto	alto	F

Requerimiento de código fuente para la aprobación de tipo

- *caudal de agua (C)*
- *caudal de otros líquidos (B,C,D)*
- *caudal de gas (C)*
- *potencia eléctrica activa (C)*
- *balanzas (B,C,D)*
- *taxímetros (C)*
- *instrumentos de medición dimensional (B,C)*
- *analizadores de gases de escape (B,C)*

Grado de conformidad de software

Bajo: la funcionalidad del software implementado para cada instrumento individual está en conformidad con la documentación aprobada

Medio: además del requerimiento anterior, dependiendo de las características técnicas, algunas partes del software se fijan en la aprobación de tipo, esto es son inalterables sin la aprobación delNB.

Alto: El software implementado en cada instrumento individual es idéntico al de aprobación de tipo.

CMMi

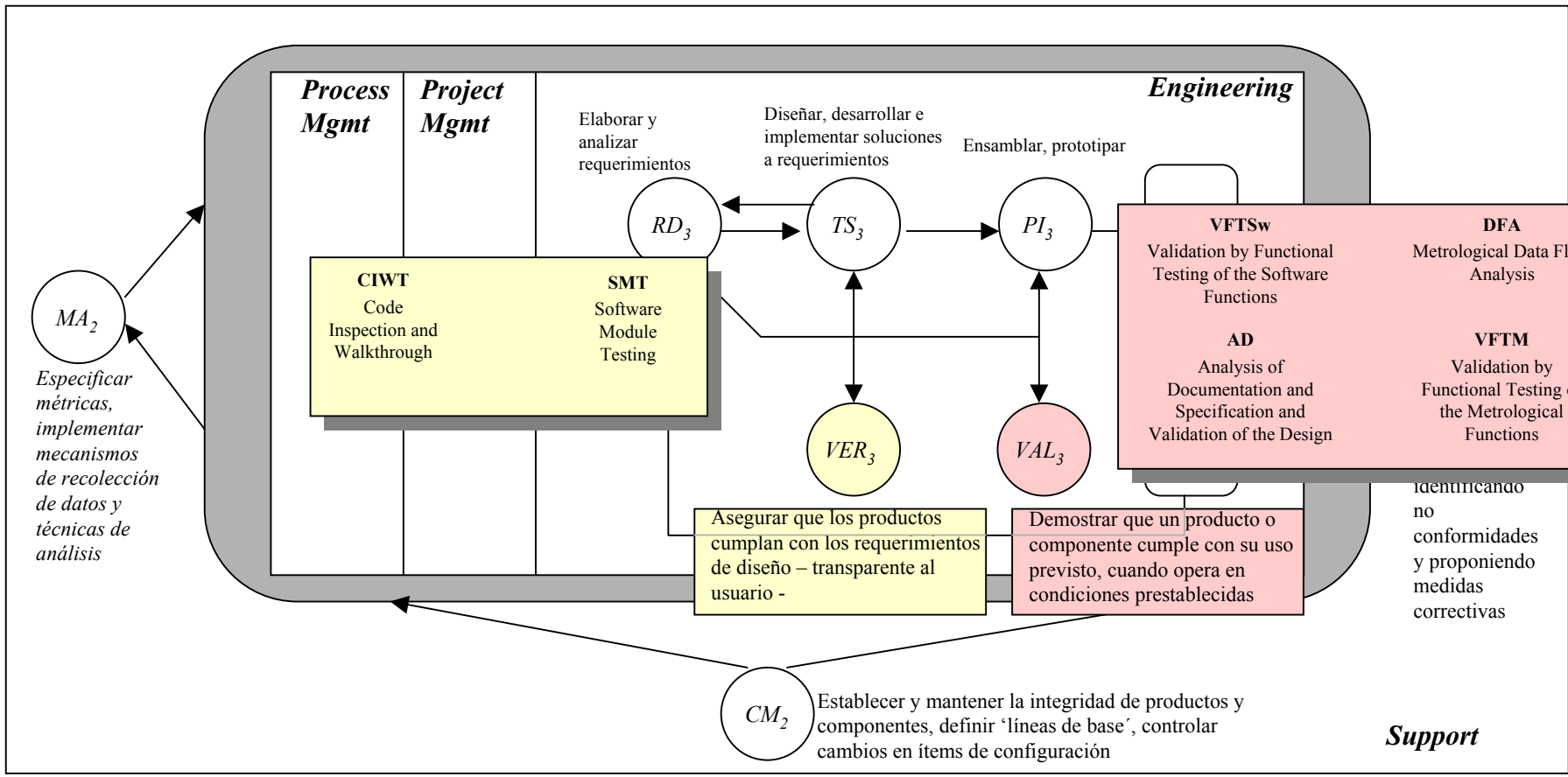
CMMI	Guía ISO 90003 (7. Realización del producto, 8. Medición, análisis y mejora)
CM	7.3.7 Control de cambios de diseño y desarrollo 7.5.3 Identificación y trazabilidad
RD	7.2.1 Determinación de requerimientos relacionados con el producto 7.3.2 Inputs de diseño y desarrollo
TS	7.3.3 Outputs de diseño y desarrollo
VER	7.3.5 Verificación de diseño y desarrollo 7.4.3 Verificación de producto adquirido 7.6 Control de dispositivos de medición
VAL	7.3.6 Validación de diseño y desarrollo 7.3.6.2 Testing
MA	8.2.4 Medición de producto 8.4 Análisis de datos
PPQA	8.2.1 Satisfacción de cliente 8.2.2 Auditoría interna 8.2.3 Monitoreo y medición de procesos 8.3 Control de producto no conforme 8.5.1 Mejora continua 8.5.2 Acciones correctivas 8.5.3 Acciones preventivas

Areas de proceso Nivel 2	Areas de proceso Nivel 3	Areas de proceso Nivel 4	Areas de proceso Nivel 5
E: Requirements Mgmt (REQM)	**E. Requirements Development (RD)**	PM: Organizational Process Performance (OPP)	PM: Organizational Innovation & Deployment (OID)
PjM: Project Planning (PP)	**E: Technical Solution (TS)**	PjM: Quantitative Project Mgmt (QPM)	S: Causal Analysis & Resolution (CAR)
PjM: Project Monitoring & Control (PMC)	**E: Product Integration (PI)**		
PjM: Supplier Agreement Mgmt (SAM)	**E: Verification (VER)**		
S: Measurement &Analysis (MA)	**E: Validation (VAL)**		
S: Process & Product Quality Assurance (PPQA)	PrM: Organizational Process Focus (OPF)		
S: Configuration Mgmt (CM)	PrM: Organizational Process Definition (OPD)		
	PrM: Organizational Training (OT)		
	PjM: Integrated Project Mgmt (IPM)		
	PjM: Risk Mgmt (RSKM)		
	PjM: Integrated Teaming (IT)		
	PjM: Integrated Supplier Mgmt (ISM)		
	S:Decision Analysis & Resolution (DAR)		
	S:Organizational Environment for Integration (OEI)		

Áreas de proceso elegidas (Nivel 2 – gestionado -)	Procesos específicos elegidos	Áreas de proceso elegidas (Nivel 3 – definido -)	Procesos específicos elegidos
MA Medición y Análisis	**SP1.1 Establecer objetivos de medición (a nivel proyecto, producto, proceso)** **SP1.2 Especificar métricas** **SP2.2 Analizar datos de medición**	**RD Desarrollo de Requeri-mientos**	**SP1.1 Definir necesidades (QFD, β-testing, diagrama UML casos de uso, ingeniería reversa)** **SP2.2 Asegurar requerimientos producto – componente (restricciones de diseño, performance)** **SP2.3 Identificar requerimientos de interfaces (internas y externas de producto)** **SP3.1 Establecer concepto operacional (secuencia de eventos posibles en el uso del producto: diagramas UML estado y actividad)**
CM Gestión de Configura-ción	**SP1.1 Establecer ítems de configuración** **SP1.2 Establecer un sistema de gestión de configuración** **SP1.3 Crear `líneas de base'**	**TS Solución Técnica**	**SP1.2 Plantear conceptos operacionales y escenarios (diagrama UML casos de uso)** **SP2.1 Diseñar producto – componente (definir standards de diseño, métricas – complejidad ciclomática, acoplamiento de rutinas)** **SP2.4 Realizar análisis 'hacer – comprar – reusar`**
PPQA Asegura-miento de la Calidad de Procesos y Productos	**SP1.1 Evaluar objetivamente procesos** **SP1.2 Evaluar objetivamente productos y servicios**	**PI Integración de Producto**	**SP1.3 Establecer procedimientos y criterios para integración de producto** **SP2.1 Revisar completitud de descripciones de interfaces (consistencia, requerimientos de cambios)**

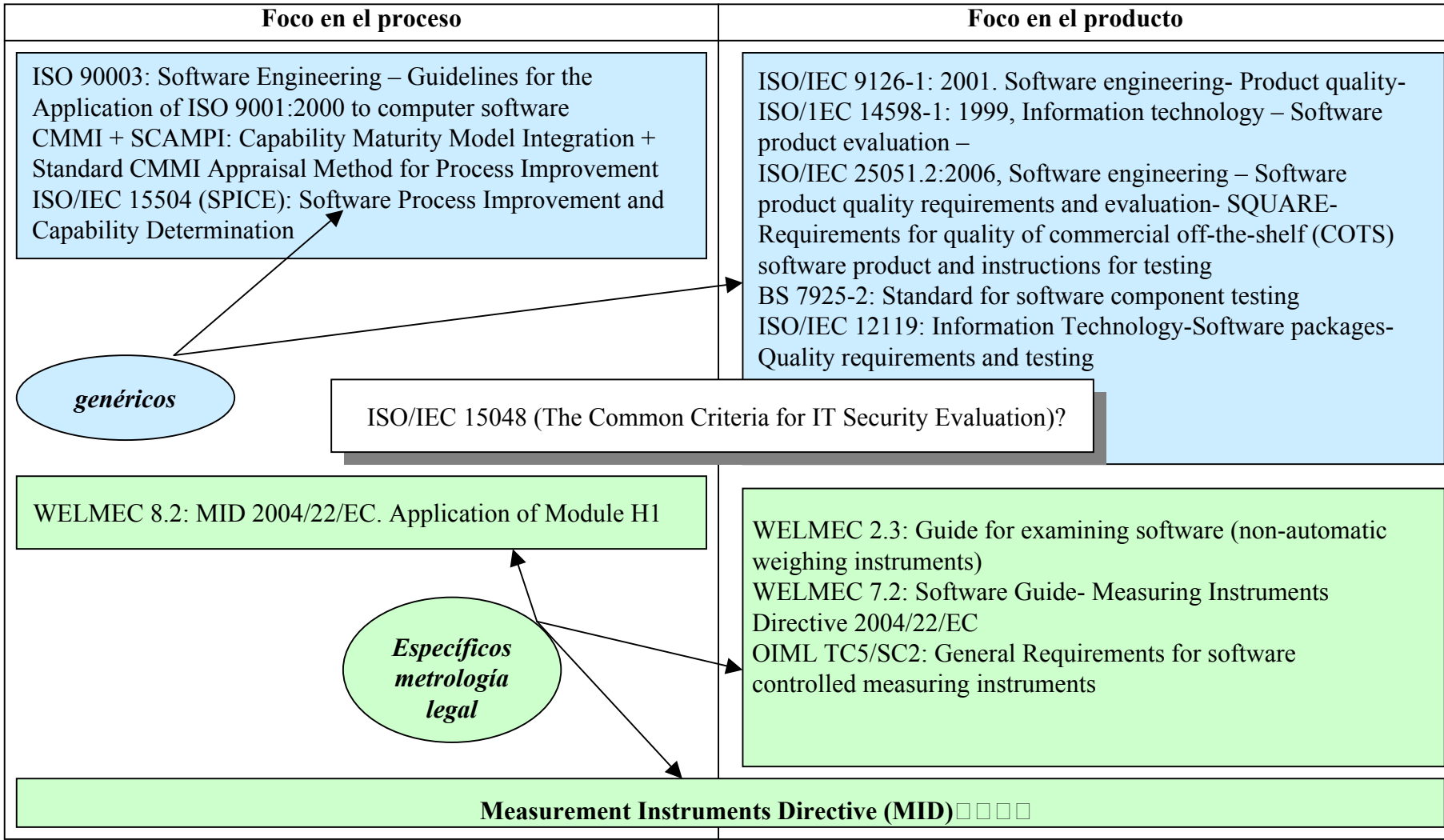
SAM (Gestión de Acuerdos con Proveedores – Area de Proceso de Nivel 2)

Foco en el proceso



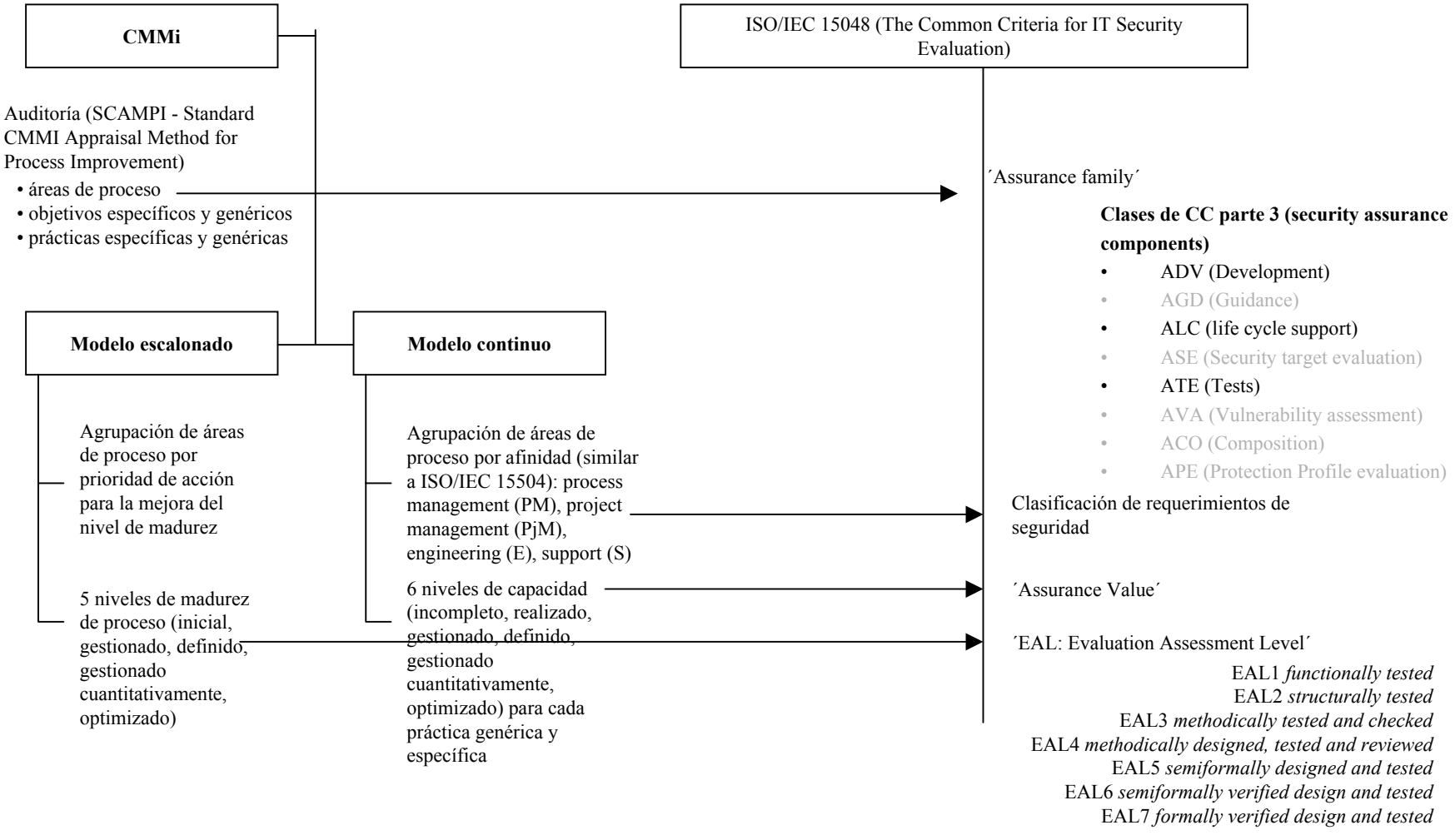
Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC

Instituto Nacional de Tecnología Industrial
INTI

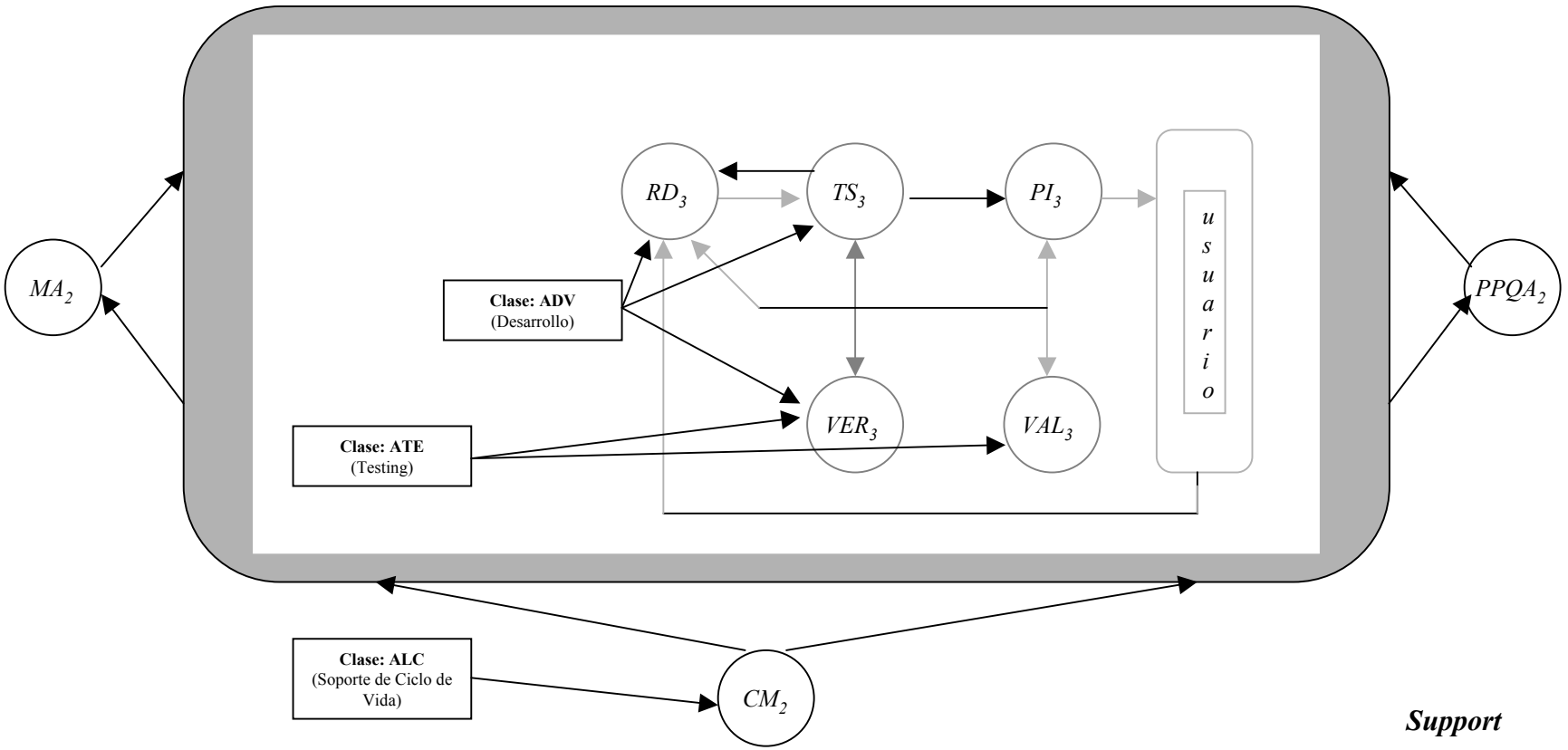


Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC

Instituto Nacional de Tecnología Industrial
INTI



ISO/IEC 15408:2005: Information Technology- Security Techniques- Evaluation criteria for IT security. (Common Criteria for Information Technology security evaluation)



Validación de software: qué, por qué, cómo. Herramientas de validación bajo el framework CMMi y su relación con la definición de clases de riesgo de WELMEC

Instituto Nacional de Tecnología Industrial
INTI

CMMi

ISO/IEC 15048 (The Common Criteria for IT Security Evaluation)

Auditoría (SCAMPI - Standard
C
P

EAL 4
- **completa especificación funcional de interfaces**
- **documentación de guía**
- **descripción de diseño a nivel de módulos**
- **testing independiente + evidencia de testing de desarrollador basado en la especificación funcional**

ADV_ARC	1	*Security architecture description*
ADV_FSP	4	*Complete functional specification*
ADV_IMP	1	*Implementation representation of the TSF*
ADV_TDS	3	*Basic modular design*
ALC_CMC	4	*Production support, acceptance procedures and automation*
ALC_CMS	4	*Problem tracking CM coverage*
ALC_DEL	1	*Delivery procedures*
ALC_DVS	1	*Identification of security measures*
ALC_LCD	1	*Developer defined life-cycle model*
ALC_TAT	1	*Well-defined development tools*
ATE_COV	2	*Analysis of coverage*
ATE_DPT	2	*Testing: security enforcing modules*
ATE_FUN	1	*Functional testing*
ATE_IND	2	*Independent testing - sample*

´Assurance family´

Clases de CC parte 3 (security assurance compon
- ADV (Development)
- AGD (Guidance)
- ALC (life cycle support)
- ASE (Security target evaluation)
- ATE (Tests)
- AVA (Vulnerability assessment)
- ACO (Composition)
- APE (Protection Profile evaluation)

ificación de requerimientos de
ridad

Prioridades de acción:

- **especificación funcional (UML diagrama de casos de uso)**
- **CM**

urance Value´

EAL: Evaluation Assessment Level´

EAL1 *functionally tested*
EAL2 *structurally tested*
EAL3 *methodically tested and checked*
EAL4 *methodically designed, tested and reviewed*
EAL5 *semiformally designed and tested*
EAL6 *semiformally verified design and tested*
EAL7 *formally verified design and tested*

- Conclusiones

  - Validación de software embebido en dispositivos de medición es posible a través de enfoque `producto` o enfoque `proceso'

  - El código fuente del software embebido puede ser requerido para la validación, de acuerdo con la definición de clase de riesgo del dispositivo, pero siempre debe verificarse durante la verificación primitiva la concordancia del software con el registrado en la aprobación de modelo

  - El framework CMMi es una herramienta conceptual útil en general para el desarrollo y mantenimiento de software embebido en equipos de medición, y en particular para el caso de la llamada opción H de validación