Tema 1. Introducción a los Sistemas Críticos

Manuel Díaz mdiaz@uma.es

Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga

Introducción a los Sistemas Críticos

- Características. El papel del software
- Historia y evolución. Tecnología y accidents
- *Safety I* vs *Safety II*
- Terminología
- Estándares y certificación

Sistemas Críticos

- Sistemas informáticos cuyos posibles fallos tienen consecuencias indeseables para las personas u organizaciones, el medio ambiente, o la sociedad en su conjunto
  - *business critical*: un fallo del sistema puede tener consecuencias negativas para una empresa
    - » ej: dispositivos defectuosos
  - *mission critical*: un fallo del sistema puede hacer fracasar la misión del sistema que controla
    - » ej: satélite de telecomunicaciones
  - *safety critical*: un fallo del sistema puede producir la muerte o heridas graves a personas, o daños inadmisibles al medio ambiente
    - » ej. control de vuelo

El Software en los Accidentes

- El software se utiliza para sustituir humanos y componentes hardware/mecánicos en procesos cada vez más críticos
- Menos obvios son los peligros de:
  - Utilizar datos generados por software (no desarrollado como crítico) para tomar decisiones críticas
  - Aplicaciones no críticas (en principio) cuyos errores pueden tener consecuencias graves (error en una base de datos de pacientes)
  - Errores en aplicaciones en herramientas de ingeniería (cálculo de una estructura,…)
  - Usar metodologías/herramientas para sistemas críticos, que no han tenido en cuenta las características del software (por ej. el software no se desgasta, pero se puede actualizar!)

Seguridad y Fallos

- El desarrollo de Sistemas Críticos se ha centrado tradicionalmente en el análisis de los fallos

Análisis de la Seguridad ≠ Análisis de los Fallos

- Esta visión tradicional es simplista para los sistemas actuales
  - Los sistemas son tan complejos que es imposible predecir todas las posible interacciones que pueden producir fallos
  - En la mayoría de las ocasiones, ningún componente falla
  - La seguridad es una **propiedad emergente**

Propiedades Emergentes

- Son aquellas propiedades del sistema considerado como un todo, en lugar de como una colección de partes
- No están determinadas solo por las propiedades de las partes del sistema, sino también por la estructura del mismo
- Algunas propiedades emergentes son predecibles (el sistema se construye para que las cumpla)
- Otras propiedades emergentes son conocidas, pero no predecibles (el móvil tiene la propiedad de interferir en las comunicaciones de otros equipos, pero no sabemos en que contexto lo hará)
- Otras propiedades, simplemente son desconocidas (los diseñadores ni siquiera pensaron sobre ellas). Estas son las que causan más problemas.

Propiedades Emergentes

- Algunas propiedades emergentes son
  - Rendimiento
  - Fiabilidad
  - Seguridad (safety)
  - Seguridad (security)
  - Usabilidad
  - Mantenibilidad
- Además son propiedades no funcionales (no se refieren a ninguna función específica del sistema)
- Algunas de estas propiedades son más importantes que la propia funcionalidad del sistema

Modelos Causales

- El concepto de seguridad en ingeniería está íntimamente ligado a los modelos causales utilizados en el análisis de los accidentes
- Estos modelos se basan en la suposición de que existen patrones comunes en los accidentes y que estos no son el resultado de meros eventos aleatorios
- Los modelos causales asumen que los accidentes son el resultado de una cadena (o árbol) de eventos de fallo y errores humanos.
  - Las relaciones causales son directas y lineales
  - La selección de los eventos que se incluyen en la cadena y cuando se considera que un determinado evento es una **causa raíz** es normalmente subjetivo y obedece a aspectos tales como
    1. Evento familiar y fácilmente aceptable como explicación del accidente
    2. Desviación de un estándar
    3. Es el primer evento para el que se conoce una acción correctiva
    4. Es políticamente aceptable
    5. No existe información suficiente para continuar en la cadena causal

Cadenas de Eventos Causales

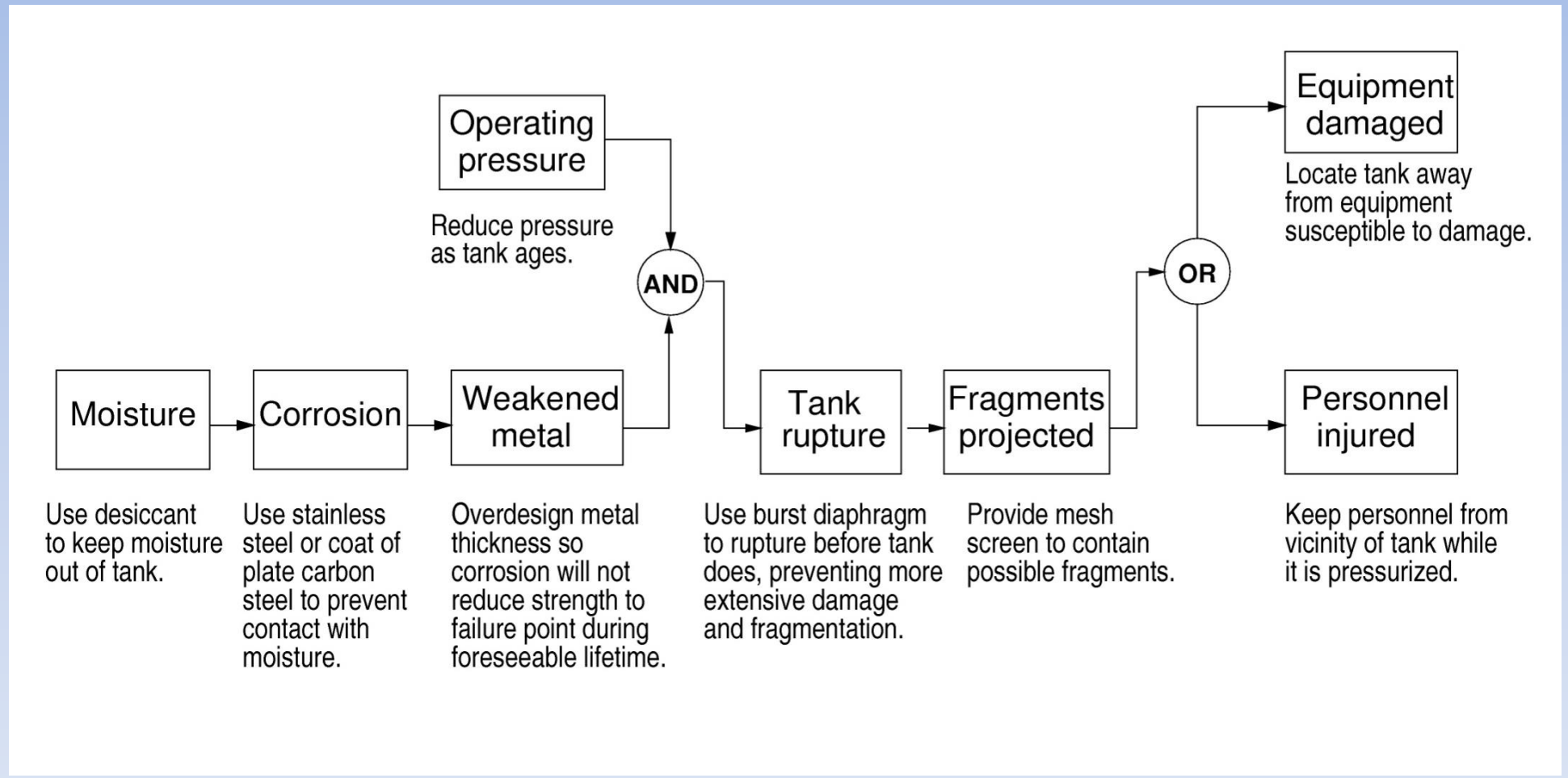
- Los eventos que se consideran casi siempre son fallos de componentes, errores humanos o eventos relacionados con la energía
- Son la base de de la mayor parte de las técnicas de análisis y

  – FTA, PRA, FMEA, Event Trees, etc.

diseño:

  – redundancia, márgenes de seguridad, sobrediseño, ….

Ejemplo de Análisis Causal

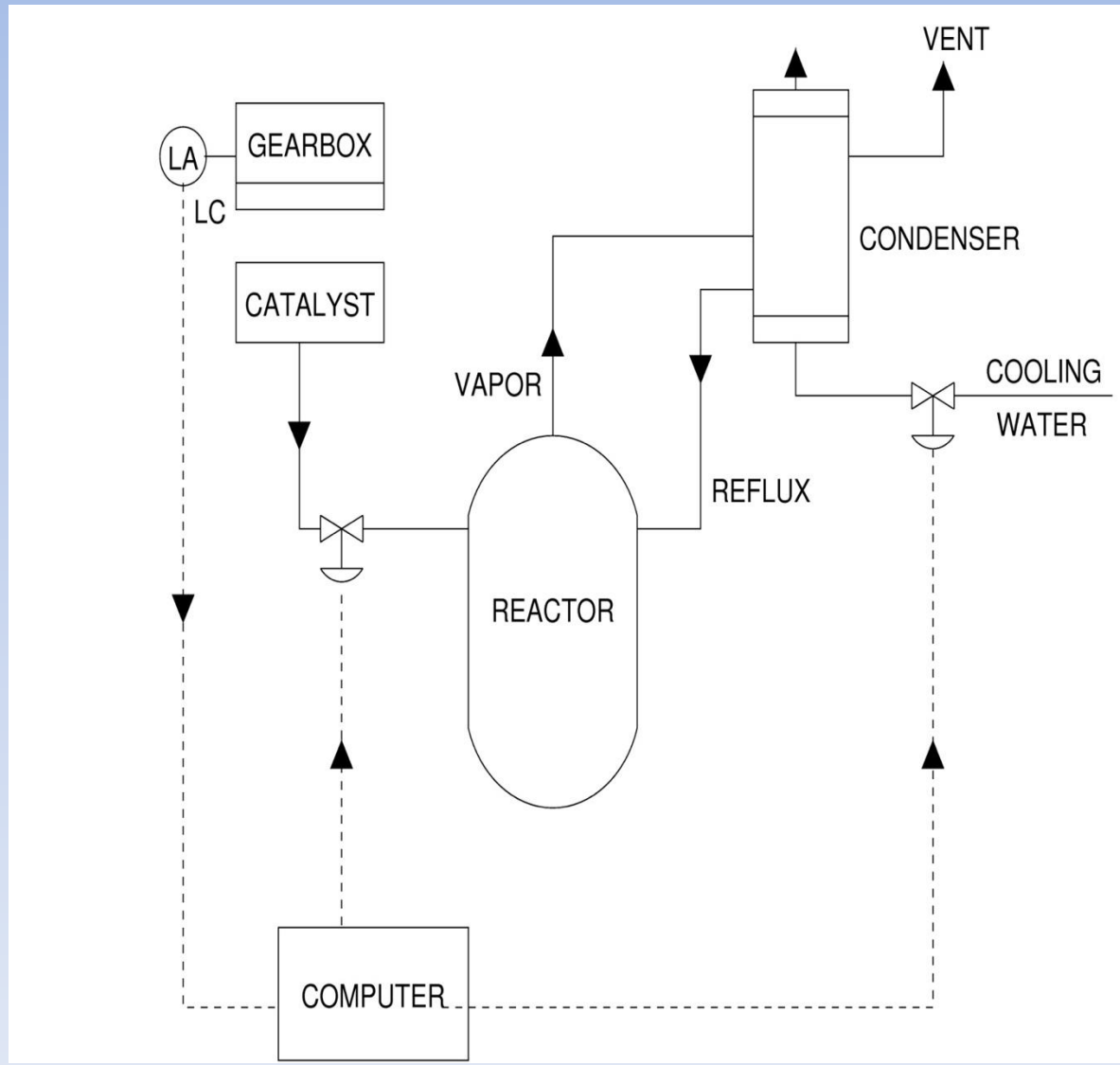


Objetivos Análisis Causal

- Existen dos objetivos en la investigación de un accidente:
  - Asignar responsabilidades
    - La responsabilidad o culpa no es un concepto de ingeniería
    - Aunque lo existan razones objetivas para distinguir la contribución de uno o varios factores en el accidente, las compañías de seguros/abogados simplifican el análisis para establecer la culpabilidad, normalmente identificando lo que denominan *causa próxima*
  - Entender por qué ocurrió para evitarlo en el futuro
    - En este caso identificar al responsable probablemente no proporcione información suficiente de la causa (por ej. el caso de la interacción de un operario con el software)

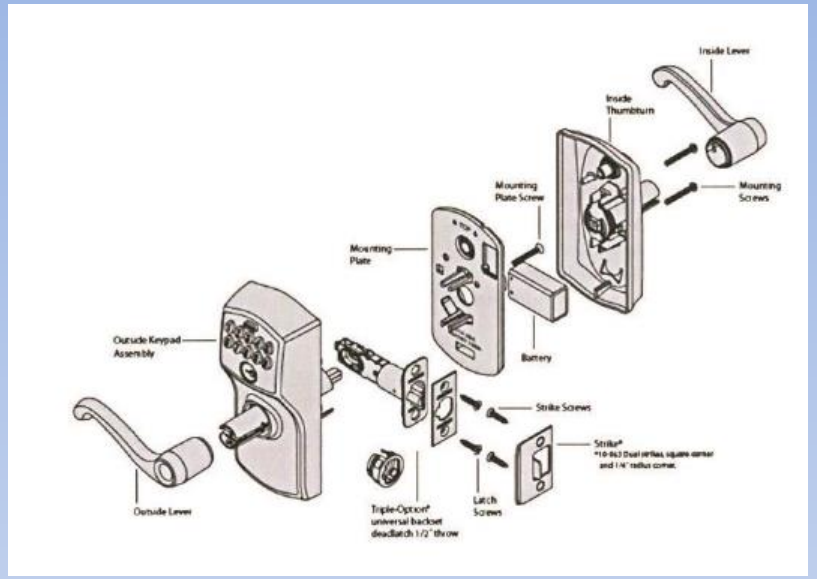
Seguridad y Análisis Causal

- El computador es el encargado de controlar el flujo del catalizador al reactor y el de agua al condensador para enfriar la reacción
- Hay sensores de entrada que alertan sobre posibles malfuncionamientos del sistema
- En los requisitos se establecía que en caso de fallo las variables de control se debían dejar como estaban y se debía hacer sonar una alarma.
- En el accidente, el catalizador acababa de ser añadido y el agua aún tenía un flujo bajo
- Un fallo del motor llevo al sistema al modo de "parada segura"
- El sistema se sobrecalentó y se abrió la válvula de seguridad, contaminando el entorno
- **NO FALLÓ NINGÚN COMPONENTE**, todos funcionaron según lo especificado



Safety I vs Safety II

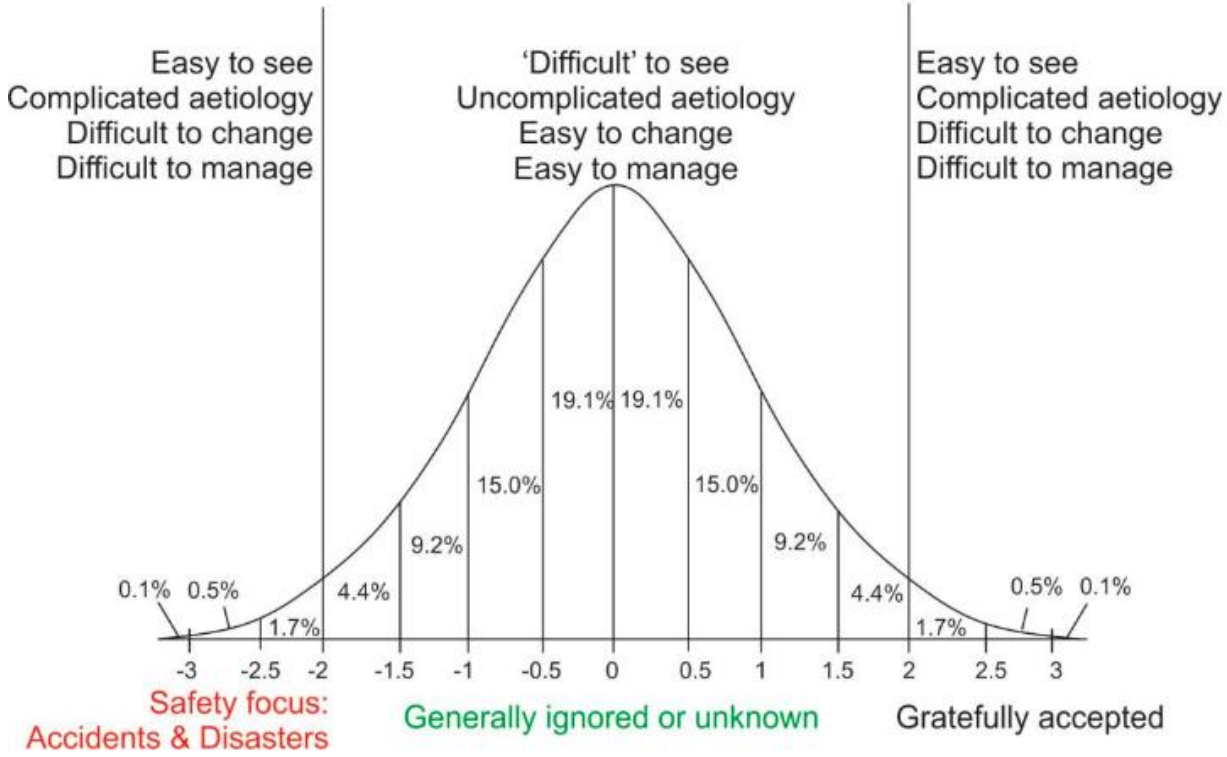
- En la aproximación más tradicional a la seguridad (Safety I) se supone que:
  - Los sistemas pueden descomponerse
  - Los componentes del sistema funcionaban de manera bimodal, ya fuera de manera correcta o incorrecta
- Esto conduce a descripciones detalladas y estables del sistema que permiten la búsqueda de causas y soluciones para los fallos de funcionamiento
- Pero estas suposiciones no se ajustan a la realidad actual:
  - Sistemas socio-tecnológicos complejos
  - El funcionamiento “normal” no es bimodal

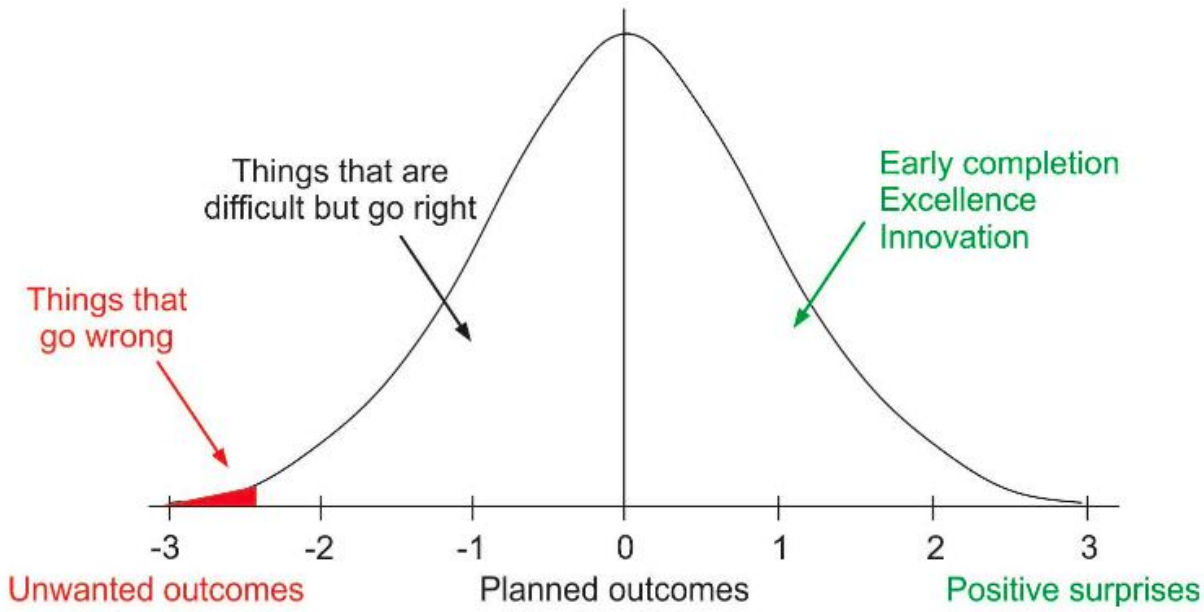


Safety I vs Safety II

- Como los sistemas actuales son cada vez más inmanejables, es imposible proporcionar una descripción completa de ellos
- Se requieren ciertos grados de variabilidad, flexibilidad o adaptabilidad para que el sistema funcione
- Los sistemas son seguros porque se pueden adaptar a condiciones cambiantes
- En lugar de analizar únicamente los pocos casos en los que las cosas salen mal, deberíamos analizar los numerosos casos en los que las cosas salen bien e intentar comprender cómo sucede eso

Safety I vs Safety II





Safety I vs Safety II

- Las prioridades de la gestión de la seguridad deben cambiar
  - En lugar de realizar investigaciones después del evento o esforzarse por reducir los resultados adversos, la gestión de la seguridad, analizar los eventos que salen bien y tratar de aprender de ellos
  - En lugar de aprender de los eventos en función de su gravedad, las personas deberían tratar de aprender de los eventos en función de su frecuencia
  - Y en lugar de analizar en profundidad los eventos graves individuales, las personas deberían explorar la regularidad de los muchos eventos frecuentes en general, para comprender los patrones en el rendimiento del Sistema
  - Una buena forma de comenzar sería reducir la dependencia del "error humano" como causa casi universal de los incidentes y, en cambio, comprender la necesidad de la variabilidad del rendimiento.

Más allá de los accidentes
SOTIF: Safety of the Intended Functionality

- El concepto surge en la industria de la automoción y hace referencia al correcto funcionamiento de los sistemas

Seguridad Activa

Seguridad Pasiva

Sistemas de Información

Sistemas mecánicos mejorados (dirección frenos,…)

Más allá de los accidentes
SOTIF: Safety of the Intended Functionality

- Peligros obvios:
  - Aceleración no intencionada
  - Activación no intencionada del airbag
  - Frenazo no intencionado
- No tan obvios: movimiento súbito del asiento

- The absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable missuse by persons is referred to as the Safety Of The Intended Functionality (SOTIF) ISO/PAS 21448:20129

Machine Learning y SOTIF

- En muchos casos cuando ocurre una situación peligrosa nada ha fallado, sino que la funcionalidad prevista ha sido la causa del problema
- Esto es especialmente relevante cuando el comportamiento del sistema ha sido aprendido, en lugar de haber sido programado explícitamente
- Es difícil determinar lo que el sistema ha aprendido realmente (problema de la interpretabilidad del modelo)
- Además, en la UE los individuos afectados por la decisión de un algoritmo tienen el derecho a una explicación de cómo se han llevado a cabo las decisiones que le han afectado

Machine Learning y SOTIF

- Nuevas dimensiones relativas a los datos tanto en calidad como en completitud
- ¿Cómo analizamos de manera adecuada la seguridad de estos sistemas?
- Cuando el sistema de encuentra con situaciones para las que no ha sido entrenado, ¿cómo debe comportarse?
- Aparece el problema de las distribuciones “heavy tail”
- No hay razón para entrenar los sistemas ante este tipo de situaciones:
  - El número de “sorpresas” latentes es muy alto
  - La frecuencia de estas “sorpresas” que son consideradas riesgos no asumibles es mayor que la tasa de fallos asumible por el sistema
- No tiene sentido “aprender” estas sorpresas, porque este aprendizaje no cambia el tamaño de la población latente de las mismas

Machine Learning y SOTIF

- El problema de probar un sistema crítico basado en aprendizaje automático no es nuevo (ISO/PAS 21448 2019) – PAS = Publicly Available Specification, no es un estándar
- Ejemplos de situaciones peligrosas
  - Ilusiones ópticas (cámara vs LIDAR)

  - Sistema de detección de objetos: el problema de la persona en patinete

Machine Learning y SOTIF

- Por definición, los escenarios peligrosos que pueden aparecer por problemas relacionados con la funcionalidad son difíciles de predecir
  - Si lo fueran, se contemplarían en el diseño
  - ISO/PAS 21448 clasifica los escenarios

	No seguros	Seguros
Conocidos	Área 2	Área 1
Desconocidos	Área 3	Área 4

  - Área 2: ¿Qué eventos peligrosos pueden surgir con los comportamientos no seguros de la funcionalidad?
  - Área 3 ¿Cuáles son las causas potenciales de dichos comportamiento?

Terminología
Peligro, Riesgo, Mitigación y Riesgo Residual

- El iceberg es el peligro
- El riesgo lo constituye el que barco pueda colisionar con él
- Una medida de mitigación puede ser pintarlo de amarillo fosforescente
- El riesgo residual, es que nos encontremos con el iceberg por la noche o con niebla

Terminología
Peligro, Riesgo, Mitigación y Riesgo Residual

- Más formalmente:
  - El **peligro** es algo pasivo que existe en el entorno y que puede ser la causa de un riesgo
  - Los **riesgos** son activos y pueden dar lugar a condiciones peligrosas
  - Cuando un riesgo es identificado y se considera suficientemente serio, necesita ser mitigado, es decir tomar una acción para reducirlo
  - Si aún así existe riesgo, se trata de un **riesgo residual**

Terminología
Peligro, Riesgo, Mitigación y Riesgo Residual

- Un ejemplo más informático:
  - La memoria puede ser una fuente de peligros
  - Algunos riesgos asociados a la memoria:
    - Efectos de los rayos cósmicos: cambio de un bit en una palabra de memoria
      - Cambio de modo de operación del sistema
      - Los códigos correctores permiten detectarlos/corregirlos
      - Errores residuales pueden ser:
        - » Varios errores simultáneos pueden no ser detectados
        - » La aplicación puede no manejar correctamente la interrupción provocada por el sistema de protección de memoria
        - » Los códigos correctores no incluyen las caches

Terminología
Peligro, Riesgo, Mitigación y Riesgo Residual

- Memoria llena
  - Una aplicación puede tener fugas de memoria que provocan su llenado
  - Una medida de mitigación es utilizar solo memoria estática y reservarla toda al comienzo de la ejecución
    - Existe el error residual de haber calculado mal el tamaño y que no haya sido detectado durante la fase de diseño y pruebas

Terminología
Disponibilidad, Fiabilidad y Predecibilidad

- Un sub-sistema software puede fallar básicamente de dos formas:
  - Puede no dar una respuesta a tiempo (fallo de disponibilidad)
  - Puede dar una respuesta incorrecta (fallo de fiabilidad)
- Incrementar la disponibilidad reduce la fiabilidad y viceversa
- Ej. consideremos un servidor que realiza algún cálculo
  - Podemos incrementar la fiabilidad haciendo que dos servidores realizen dicho cálculo y comparar las respuestas.
  - Es una técnica muuuy antigua y utilizada:

> When the formula is very complicated, it may be algebraically arranged for computation in two or more distinct ways, and two or more sets of cards may be made. If the same constants are now employed with each set, and if under these circumstances the results agree, we may be quite secure in the accuracy of them.
> **Charles Babbage, 1837**

Terminología
Disponibilidad, Fiabilidad y Predecibilidad

- Disponibilidad y confianza/predecibilidad están normalmente contrapuestas:
  - Un incremento en la fiabilidad normalmente reduce la disponibilidad (ej. en el caso anterior si uno de los servidores falla, la respuesta no estaria disponible)
    - Es fácil encontar casos en los que la fiabilidad es más importante que la disponibilidad, pero también ocurre al contrario (ej. un coche sin frenos y al que no le funciona el arranque)
    - Una alta fiabilidad (ej. probabilidad de fallo 10^{-x}) no garantiza un sistema seguro
    - El enfoque de trabajar solo en incrementar la fiabilidad de los componentes es un enfoque inapropiado en estos sistemas
  - En el caso de las pruebas ¿es más difícil saber si un servicio es accesible o fiable?
    - Las pruebas se centran en la fiabilidad, ¿por qué?
- El concepto de confianza (dependability) combina las dos propiedades.
  - Un sistema es confiable/predecible si está disponible y es fiable

Disponibilidad

- La disponibilidad se suele medir en “nueves”, como porcentaje de tiempo en el que un determinado servicio o sistema está disponible
- Una disponibilidad de 3 9´s se considera aceptable en un sistema convencional y cualquier cosa por encima se considera como alta disponibilidad

Disponibilidad %	Tiempo de fallo por día
90% (un nueve)	2.40 horas
99% (dos nueves)	14.40 minutos
99.9%	1.44 minutos
99.99%	8.69 segundos
99.999%	864 milisegundos

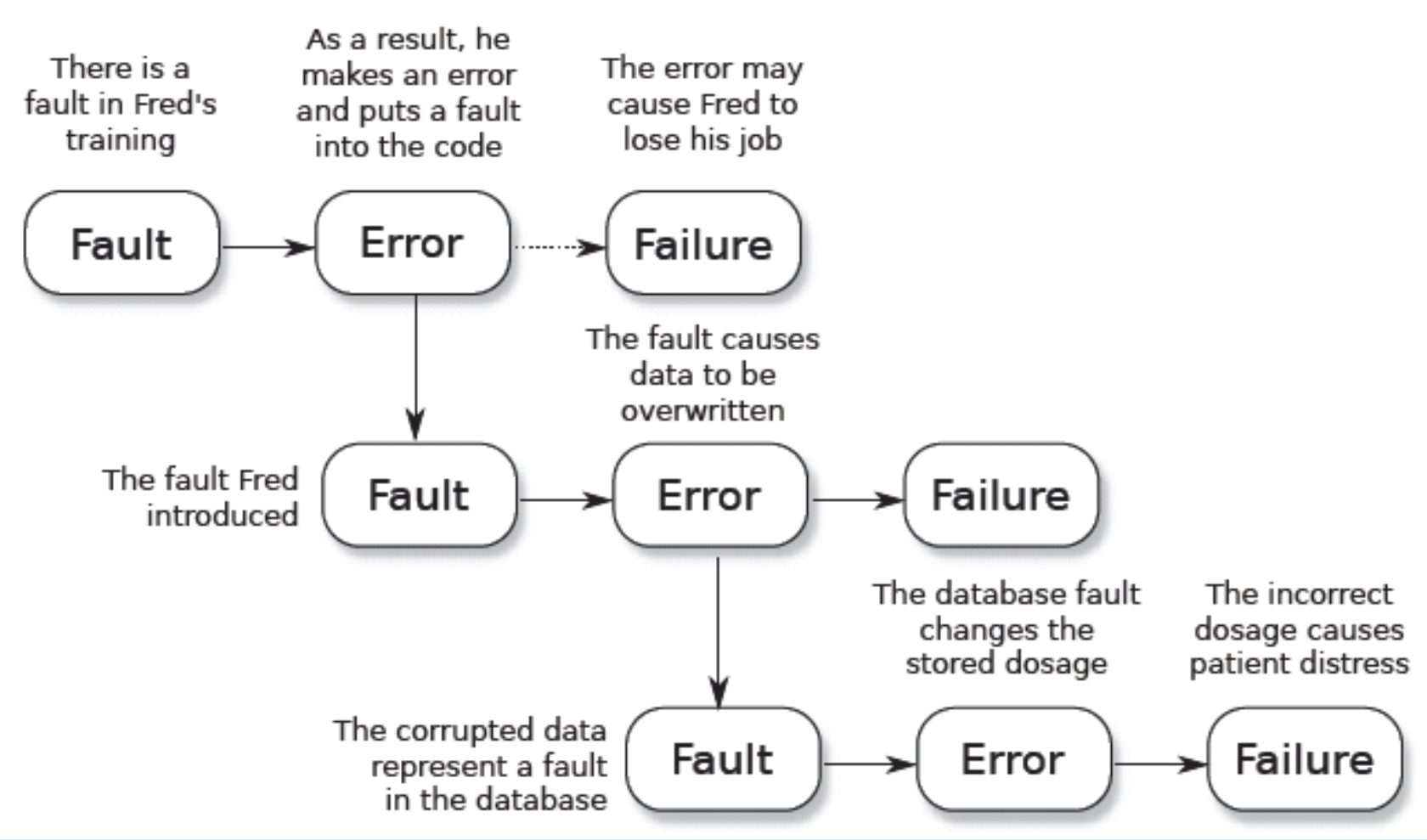
Seguridad funcional

- Un sistema puede mejorar su seguridad de dos formas:
  - porque incorpora sistemas pasivos de seguridad que evitan que el sistema pueda producir un daño por su diseño, incluso sin que ningún componente tenga un funcionamiento activo (seguridad pasiva)
  - Si para que el sistema sea seguro es necesario que uno o varios componentes activos funcionen, se tratará de *elementos de seguridad funcional* (functional safety)
- Ej. Un sistema de protección de un laser:
  - *Pasivo*: Evita que un usuario pueda mirarlo directamente por diseño físico
  - *Elementos de seguridad funcional*: software de monitorización que desconecta el laser si se desconecta la fibra

Defectos, Errores y Fallos
Fault, Error, Failure

> “ We want to reduce the number of faults inserted into the system, to detect and remove faults before they become errors, to detect errors and prevent them from becoming failures and, if a failure does occur, to handle it in the safest way possible”
>
> C. Hobbs 2016

Fallos y Errores
Fault, Failure, Error



Recuperación de errores hacia atrás y hacia adelante

- Recuperación hacia atrás:
  - Se devuelve el sistema a un estado que se sabe (o se cree) que es estable y correcto.
  - Cómo se trata el error que pudiera provocar el fallo (descartar la entrada, volverla a considerar,…) dependerá de la implementación
  - La mayor desventaja: hay que almacenar los estados de vuelta atrás
- Recuperación hacia adelante:
  - El sistema se conduce a un estado seguro independiente del estado anterior y de la entrada
  - No es necesario guardar información sobre punto de recuperación

Terminología
Sistemas Accidentales

- Un sistema accidental es aquel que no ha sido especificamente diseñado y probado para ser incorporado en otro sistema que puede tener requisitos críticos de seguridad
- El software empotrado suele formar parte de sistemas accidentales y podemos encontrar componentes que no han cosiderado que podían ser utilizados por varías hebras o que usan recursión y necesitan más pila de la que se ha reservado

Martyn Thomas described an experiment wherein (GPS) signals were deliberately jammed in part of the North Sea and a ship was taken through the area to see what would happen. As expected, the position of the ship became imprecise (sometimes the ship was in the middle of Norway, sometimes in the middle of Germany!), but unexpectedly, the ship's radar also stopped working.

The experimental team contacted the company that manufactured the radar and was told that GPS was not used within it. They then contacted the manufacturers of the components in the radar and found that one small component was using GPS timing signals.

Terminología
¿Qué consideramos software?

- ¿Es hardware un ASIC (Application Specific Integrated Circuit) que ha sido diseñado con una herramienta software?
- ¿Y un componente generado por una herramienta de modelado con 5000 líneas en C?
- Siguiendo la norma del *Rail Safety and Standard Board,* como regla general consideraremos:
  - *Hardware:* Número suficientemente pequeño de estados para poder ser cubiertos todos por técnicas de testing exahustivo
  - *Software* en cualquier otro caso
- Incluímos en el concepto de software, los datos de configuración
  - *'software' means computer programs and corresponding configuration data . . .*
  - *'configuration data' means data that configures a generic software system to a particular instance of its use;*

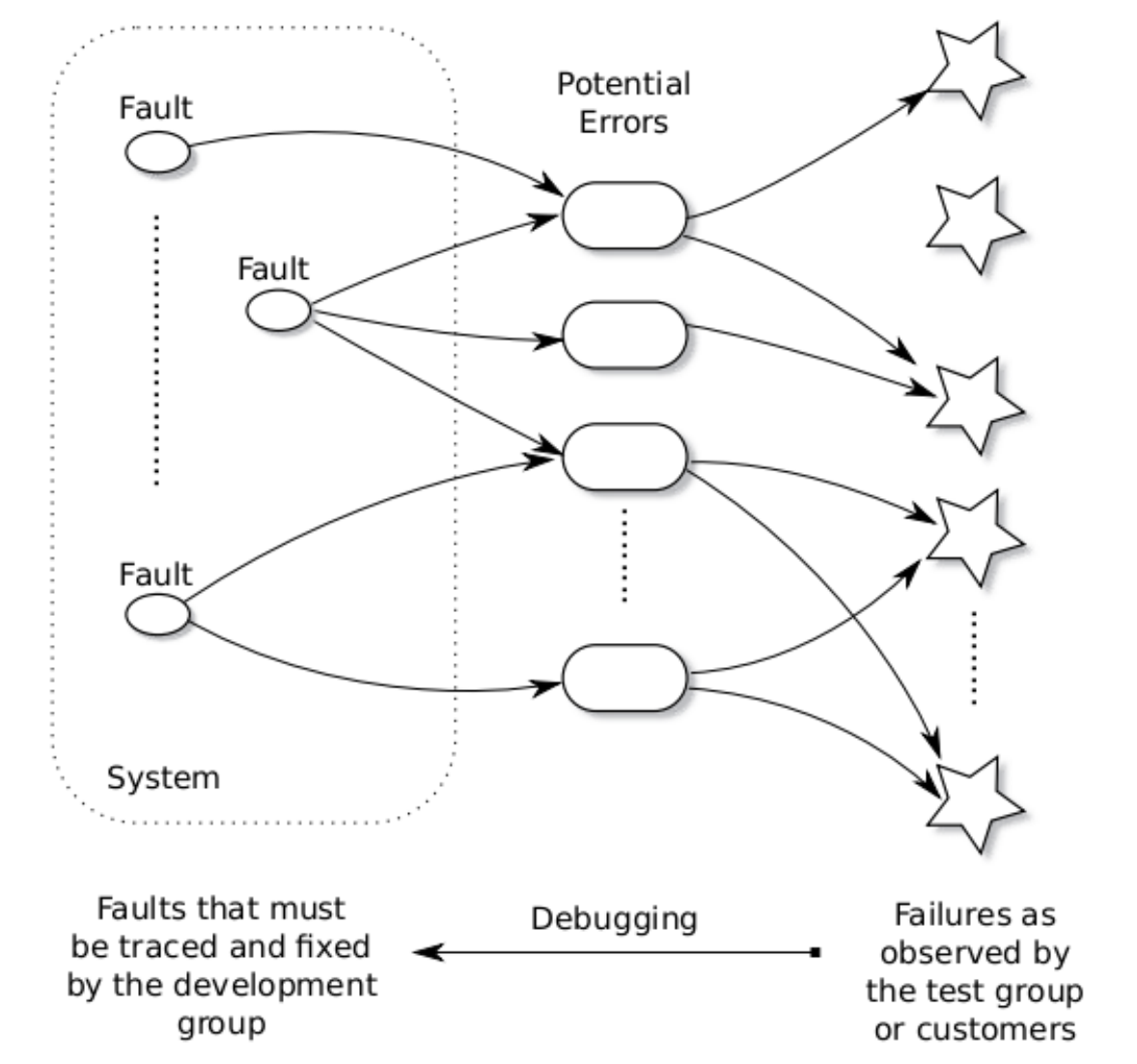
Terminología
Heisenbugs y *Bohrbugs*

- *Bohrbug*: error bien definido, determinista, con propiedades que no cambian cuando se añade código para depuración
- *Heisenbug*: error ocasional que aparece y desaparece, que puede cambiar en depuración. Suele ser causado por problemas de tiempo, concurrencia,… Típicamente el error sería reportado como:

  > "The eighth time that test case 1243 was run, the system crashed. We've run that test case several dozen times since, and no other failure has occurred. No memory dump is available. Please fix the bug."

- Una característica de estos defectos es que el mismo error interno puede dar lugar a diferentes fallos y que estos pueden aparecer más tarde y en otro lugar del código

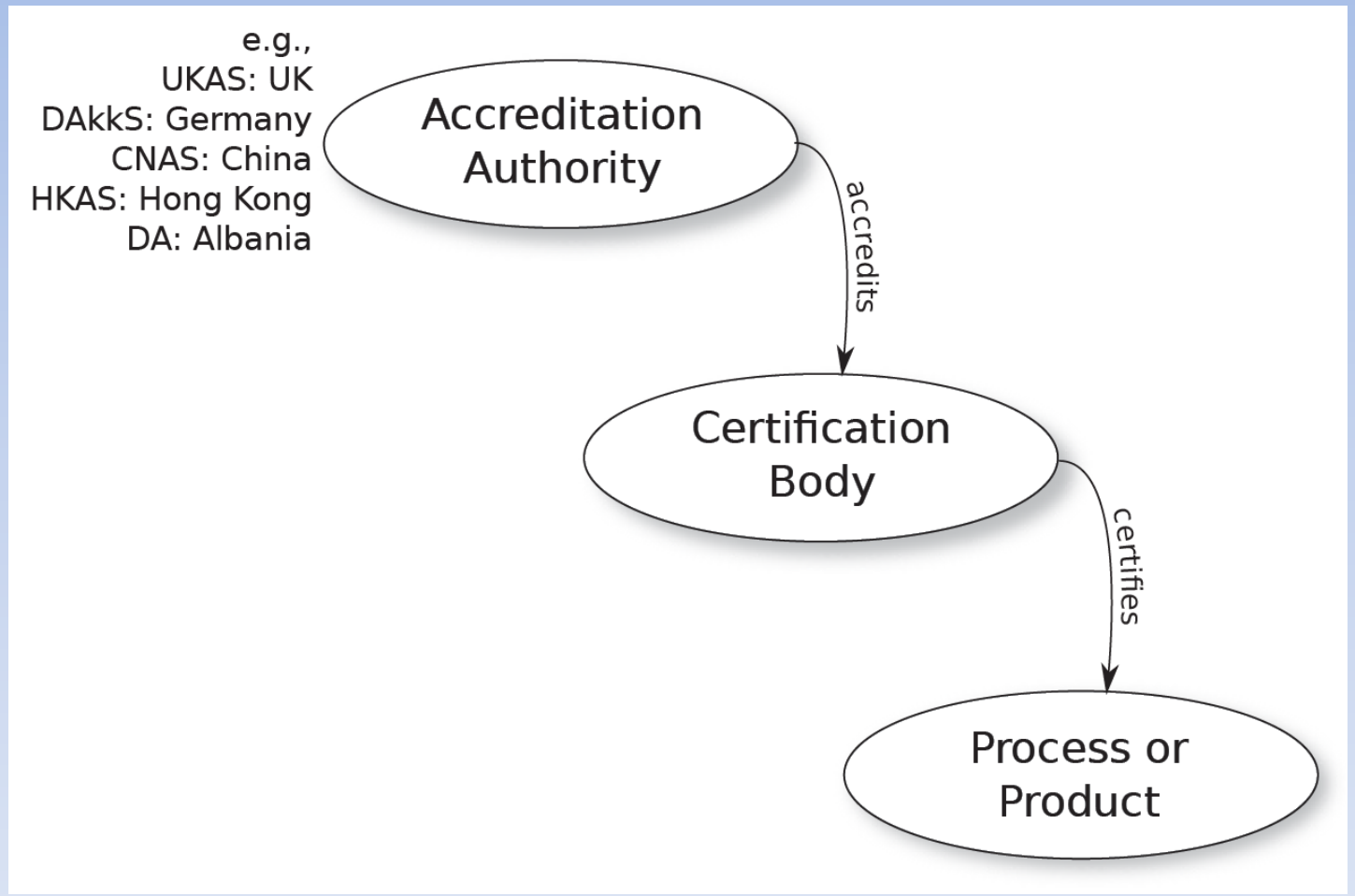
Terminología
Heisenbugs y *Bohrbugs*



Estándares de Seguridad y Certificación

- Existen muchos estándares y no está clara la frontera entre muchos de ellos
- También existen muchas organizaciones diferentes que promulgan estándares
  - Nos centraremos en solo los más relavantes de la International Electrotechnical Commission (IEC) y la International Organization for Standards (ISO)
  - Ambas están establecidas en Suiza y en muchas ocasiones colaboran en estándares conjuntos (ISO/IEC 42010 sobre descripción de arquitecturas)
  - Ambas organizaciones cobran (además un precio alto) por el acceso a los estándares (otras como IETF o DMTF no lo hacen)
  - ¿Por qué? Para evitar otras fuentes de financiación como los royalties por las certificaciones, que podrían ocasionar conflictos de interés
  - El proceso de elaboración de los estándares es similar (un país, un voto).

El Proceso de Certificación



Estándares de Seguridad y Certificación

- Según el estándar se requerirá:
  - El uso de diferentes técnicas (prevención, tolerancia, análisis y eliminación) en las diferentes etapas del ciclo de vida del software para cada nivel de criticidad
  - Que las mismas actividades, técnicas y métodos se utilicen para el software configurable y sus datos. Hay sistemas críticos que reutilizan software crítico genérico, pero configurable a través de sus datos y estos son críticos también
  - Diferentes niveles de independencia del personal para realizar algunas actividades
  - Exigencias respecto a la confianza necesaria respecto a la reutilización como parte del software crítico
  - Exigencias respecto a las herramientas utilizadas en el desarrollo, verificación y validación

Estándares de Seguridad y Certificación

- Certificación de un producto o proceso:
  - Una empresa puede auto-declarar que un producto o proceso cumple un determinado estándar mediante la confirmación de una compañia auditora
  - Alternativamente puede llevar a cabo la certificación frente a un organismo de certificación
- Acreditación de los organismos de certificación
  - Cualquier empresa, en principio, puede acreditarse como organismo de certificación
  - La acreditación se lleva a cabo frente a las autoridades nacionales de certificación
  - La acreditación es para un objetivo específico (ej. Hardware -no software- bajo el IEC 61508, SIL)

Estándares de Seguridad y Certificación

- Acreditación de los acreditadores
  - Cada uno de los paises que forman parte del International Accreditation Forum (IAF), tiene uno o más miembro que acreditan organismos de certificación
  - No hay ningún ente superior, así qué

    quis custodiet ipsos custodes?
  - Cada miembro examina a los demás en un ciclo continuo de evaluaciones mutuas
  - Los costes de certificación son elevados y la tasa de éxito es baja (ej. IEC 61508 alrededor del 25% en 2010)

Estándares de Seguridad y Certificación

- ¿Por qué necesitamos estándares?
  - Obvio en conectores, protocolos,..
  - En el caso de la seguridad es menos obvio y generalmente se generan como respuesta a accidentes:

Historically, the creation of standards (and not just safety-related ones) has often been driven by disasters. A disaster occurs and the public demands that it never occur again….
The response is the creation of a standard to which industry must comply, with the intent of raising the quality of products and reducing the chance of a repetition. In this sense, the standard provides protection for a public that does not understand the engineering process.

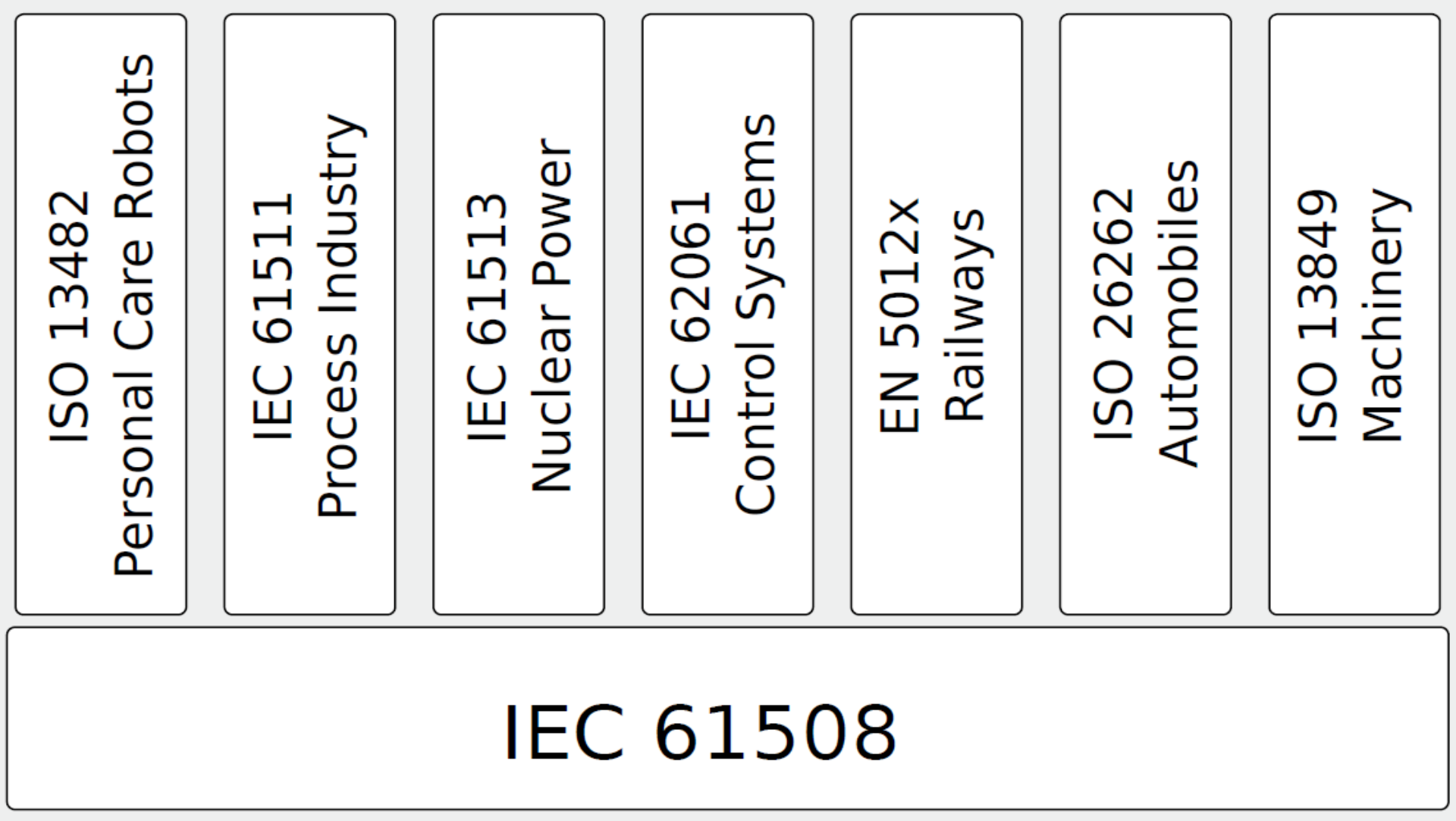
Estándares de Seguridad Funcional
IEC 61508

- *Functional safety of electrical/electronic/programmable electronic safety-related systems*
  - Tiene 7 partes, las tres primeras normativas y las demás informativas
  - La parte normativa cubre todos los aspectos de gestión de proyectos y desarrollo de hardware y software
  - Dos ediciones hasta el momento: 2000 y 2010
  - El modelo utiliza como referencia un equipo industrial que contiene hardware y software que puede dar lugar a distintos fallos y una función de seguridad que lo monitoriza y lo mueve a un estado seguro cuando una situación de peligro aparece.

IEC TS 61508-3-2:2024 26/8/2024

Requirements and guidance in the use of mathematical and logical techniques for establishing exact properties of software and its documentation

Estándares de Seguridad Funcional
IEC 61508



Estándares de Seguridad Funcional
IEC 61508

- El nivel de dependencia entre el IEC 61508 y otros estándares específicos de la industria varía entre éstas.
- En el de ferrocarriles EN 5012x, el estándar está fuertemente basado en el mismo y hay múltiples tablas directamente copiadas sin modificación.
- El de automoción ISO 26262, también copia muchas de las tablas, con cambios de distinta entidad.
- Sin embargo, otros como el ISO 13482 (Robótica) están mucho menos basados en él.

IEC 61508
SIL: Safety Integrity Level

- IEC 61508 define un conjunto de niveles de integridad (SILs) que se basan puramente en la probabilidad de fallo, ligando los conceptos de fallos y seguridad (modelo causal, no basado en teoría de procesos como STAMP)
- Se distinguen dos tipos de sistemas de seguridad:
  - Sistemas que solo se ejecutan bajo demanda. Se dividen a su vez entre aquellos que se ejecutan de forma infrecuente (menos de una vez al año) y aquellos que se invocan con más frecuencia.
    - Ej, un sistema de defensa contra la fusión del núcleo de un reactor nuclear. SIL 1 en este sistema significa una probabilidad de fallo menor a 0.1 y SIL4 menor que 0.0001
  - Sistemas que se ejecutan de forma continua o bajo frecuencia con una demanda mayor que una vez a la año
    - Aquí SIL1 significa probabilidad de fallo menor que 10^{-5} por hora de funcionamiento y SIL4 10^{-8}

IEC 61508
Probabilidades de fallo SIL4

Hours	Years	Probability of a failure
100,000	11	0.001
1,000,000	114	0.00995
10,000,000	1,141	0.09516
100,000,000	11,408	0.63212
1,000,000,000	114,077	0.99995

- Esto son estadísticas, por lo que para poder discutir sobre ellas necesitamos muestras de fallos para el análisis
- Claramente esto parece imposible para dispositivos de SIL4, salvo excepciones como los componentes usados en coches (si se realizara una monitorización efectiva…)

IEC 61508

SIL: Safety Integrity Level

	Técnica/Medida*	Ref.	SIL1	SIL2	SIL3	SIL4
1	Detección de fallo y diagnóstico	C.3.1	---	R	HR	HR
2	Códigos de detección y corrección de los errores	C.3.2	R	R	R	HR
3a	Programación por reafirmación de averías	C.3.3	R	R	R	HR
3b	Técnicas basándose en dispositivos externos de seguridad	C.3.4	---	R	R	R
3c	Programación diversa	C.3.5	R	R	R	HR
3d	Bloque de recuperación	C.3.6	R	R	R	R
3e	Recuperación hacia atrás	C.3.7	R	R	R	R
3f	Recuperación hacia delante	C.3.8	R	R	R	R
3g	Mecanismos de recuperación del fallo por relanzamiento de ejecución	C.3.9	R	R	R	HR
3h	Casos de memorización ejecutada	C.3.10	---	R	R	HR
4	Degradación "elegante"	C.3.11	R	R	HR	HR
5	Inteligencia artificial – corrección de fallo	C.3.12	---	NR	NR	NR
6	Reconfiguración dinámica	C.3.13	---	NR	NR	NR
7a	Métodos est MASCOT,					
7b	Métodos ser					
7c	Métodos est CSP, HOL,					
8	Herramienta					
NOTA –	Convien requisito CEI 615					
*	Las técnicas y me alternativas se in					

	Técnica/Medida*	Ref.	SIL1	SIL2	SIL3	SIL4
1	Diagramas de bloques lógicos/funcionales	Véase la nota	R	R	HR	HR
2	Diagramas de secuencia	Véase la nota	R	R	HR	HR
3	Diagramas de flujo de datos	C.2.2	R	R	R	R
4	Autómatas de estados finitos/diagramas de cambios de estado	B.2.3.2	R	R	HR	HR
5	Redes de Petri temporales	B.2.3.3	R	R	HR	HR
6	Tablas de decisión/de verdad	C.6.1	R	R	HR	HR
NOTA – Los diagramas de bloques del softwares/funcionales y los diagramas de secuencia se describen en la Norma CEI 61131-3.						
* Las técnicas y medidas apropiadas se deben seleccionar en función del nivel de integridad de seguridad.						

ISO 26262

- Especialización de IEC 61508 para coches (no autobuses o camiones)
- Tiene 10 partes, las primeras nueve publicadas en 2011 y la 10 que es una guía para aplicar las anteriores en 2012
- Las partes 2-7 describen los procesos para desarrollo de productos:
  - Gestión de la seguridad (Parte 2)
  - Concepto del producto (Parte 3)
  - Desarrollo a nivel de sistema (Parte 4)
  - Desarrollo de hardware (Parte 5)
  - Desarrollo de software (Parte 6)
  - Producción (parte 7)

ISO 26262

- La parte 6, como el estándar anterior incluye tablas de técnicas recomendadas y altamente recomendadas para cada ASIL (Automotive Satety Integrity Level). Existen 16 tablas con solapamientos, pero no idénticas.
- Los ASIL no están basados en probabilidad de fallo por hora de uso, sino que se consideran tres aspectos:
  - ¿Qué tipo de heridas puede provocar este evento? S0(sin heridas) a S3(pone en peligro la vida)
  - ¿Qué probabilidad de ocurrir el evento tiene en condiciones normales? E0 (increible) a E4 (alta probabilidad). E1 se aplica a eventos que ocurren menos de una vez al año y E2 varias veces al año.
    - Este índice dependen de la localización
  - ¿Cuántos conductores podrían controlar el evento? C0 (controlable en general) a C3 (difícil de controlar o incontrolable)

ISO 26262

Severity	Likelihood	Controllability		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

ISO 26262

- La tabla indica los requisitos frente a los que la función de seguridad tiene que ser evaluada
- Muchas de las entradas indican QM (Quality Management) y lo que indican es que no hay requisito de seguridad funcional asociado según el estándar y que el componente tiene que ser desarrollado según el proceso de gestión de la calidad establecido (ej. IATF 16949)
- Un sistema completo normalmente presenta una mezcla de ASILs
  - Tradicionalmente ha existido una división (normalmente física) entre las funciones relativas a seguridad y otras como entretenimiento, navegación etc,
  - AutoSAR (AUTomotive Open System Architectura) indica el tipo de separación que debe existir entre los componentes según los requisitos
  - Hoy en día el aislamiento “físico” de los componentes es imposible debido a la reducción de costes obtenida mediante la combinación de componentes hardware (cada vez más potentes) y técnicas como la fusión de sensores

ISO 26262

- ¿Cómo puede un diseño garantizar que los subsistemas no interfieren?
  - El anexo D de ISO 26262-6 y el anexo F de IEC 61508-3 consideran esta cuestión y listan las fuentes principales de interacción que deben ser evitadas durante el diseño
    - Tiempo: ¿Cómo podemos garantizar que un subsistema no absorbe el tiempo del procesador y provoca postergación en otro?
    - Deadlocks y livelocks: ¿Cómo los prevenimos por diseño?
    - Memoria: ¿Cómo garantizamos que un sistema no sobreescribe la memoria de otro?
    - Mensajes: si dos sistemas intercambian mensajes, ¿cómo mantenemos la seguridad ante la perdida, corrupción, retraso o duplicación de éstos?

ISO 26262

- Aunque la lista es útil no es completa.
- La mayor parte de los sistemas operativos modernos proporcionan soporte para solucionar los problemas (protección de memoria, protocolos de herencia de prioridad,..)
- No está resuelto el problema en el caso de los sistemas multi-núcleo
  - Existen resultados de imposibilidad de diseño para la solución de algunos de estos problemas
- Hay interacciones que no están bajo el control de los subsistemas y del propio sistema operativo:
  - Caches de instrucciones y datos
  - Técnicas como la reordenación de las instrucciones en los pipelines, pre-fetching complejo,…
  - Coprocesadores gráficos, controladores NVRAM,…

ISO 26262
SEooCs

- **SEooC: Safety Element out of Context**
- Un SEooC es un elemento que va a ser usado en un sistema del automóvil, pero que no ha sido específicamente diseñado para ese sistema
  - Ej. Necesitamos una base de dato y queremos usar MySQL
- En este caso hay que especificar claramente que se espera del elemento y garantizar que el SEooC cumple los requisitos, bien por si mismo (out of the box) o después de una modificación
- Si el SEooC ha sido desarrollado por una tercera parte, entonces, hay que monitorizar el informe de errores proporcionado y llevar a cabo un análisis de impacto para determinar si alguno de ellos afecta a la seguridad del sistema
  - Imaginemos un componente como un kernel de Linux, con listas de errores muy largas y dinámicas….
  - El término SOUP en IEC 62304 (estándar para dispositivos médicos) de forma similar

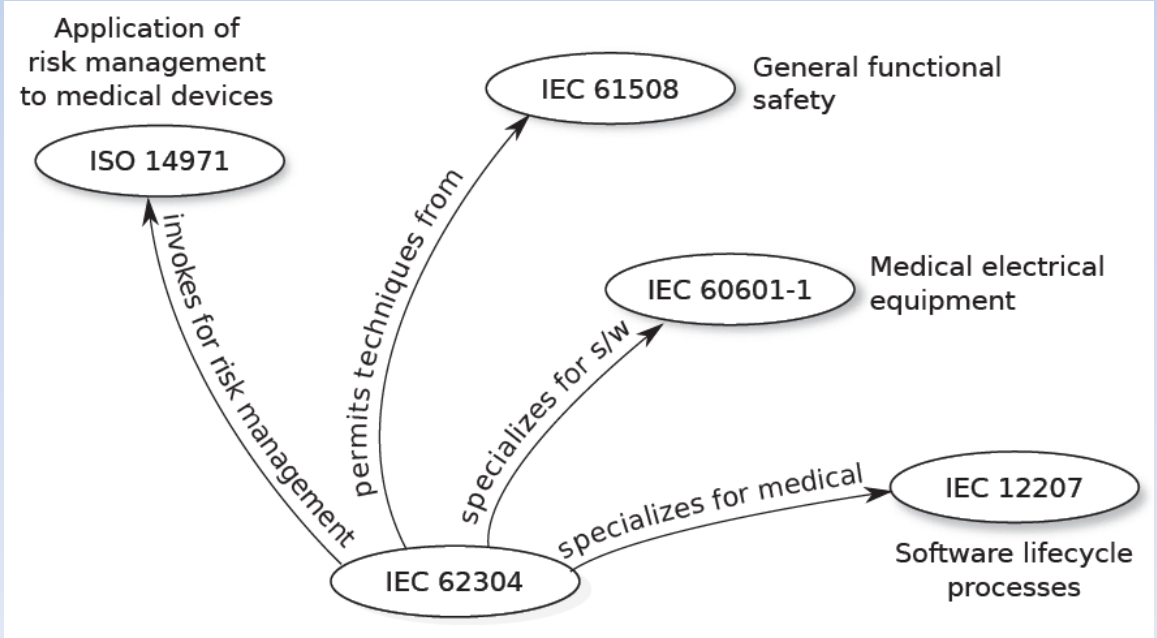
IEC 62304

- Este estándar no es estrictamente un estándar de seguridad funcional
  - Se centra más en el ciclo de vida de los procesos de desarrollo software para dispositivos médicos
- Define tres clases de riesgos equivalentes a los SILs/ASILs
  - Clase A: no hay posibilidad de herir o dañar al paciente, al operador o a los acompañantes por fallo en el software
  - Clase B: existe la posibilidad, pero los daños/heridas no son graves
  - Clase C: en otro caso
- El debate más interesante es el relativo a un comentario adicional sobre el software

> If a dangerous condition may arise from a failure of the software system to behave as specified, the probability of such failure shall be assumed to be 100 %

IEC 62304

- No se indica en que periodo de tiempo
  - Esto es necesario para incorporarlo en el modelo de fallos
- El estándar se centra en los procesos y define los siguientes:
  - Proceso de desarrollo
  - Mantenimiento
  - Gestión de la configuración
  - Resolución de problemas, incluyendo la obligación de información al usuario de errores potencialmente peligrosos



Procesos y Estándares

- En esta asignatura nos centramos más en los aspectos técnicos que en los relativos a procedimientos y procesos, lo que no quiere decir que estos no sean relevantes…

> "…strong, auditable processes are necessary if we are to have confidence in the product, but the fact that a process has been followed in developing a product does not tell you anything about the properties of the product itself."
>
> Martyn Thomas, "Engineering Judgement", 2004