



# **Certificazione di Sistemi Software Safety-Critical**

**F. Battini  
IEEE Computer Society  
Italy Chapter**

# Obiettivo dell'Intervento



Questo intervento e' finalizzato alla presentazione dei principi alla base delle certificazioni di sistemi software safety-critical.

**Riferimento principale: la norma DO-178B/ED-12B  
Software Considerations in Airborne Systems  
And Equipment Certification**

**Durante la presentazione i seguenti punti verranno toccati:**

- **Aspetti relativi allo sviluppo di Sistemi (Software) safety-critical**
  - Classificazione degli eventi di avaria
  - Fault prevention vs. Fault removal
- **Le normative di Progetto**
  - I principi generali alla base delle *consideration* per la safety in campo avionico
  - Obiettivi di processo vs requisiti normativi
- **Come considerare e trattare :**
  - Il Riuso di componenti software
  - L'Uso di componenti commerciali off-the-shelf
  - Il rischio legato all'utilizzo di strumenti di sviluppo

# Introduzione al Problema

Technical Note CMU/SEI-2004-TN-016

Charles B. Weinstock, John B. Goodenough, John J. Hudak; May 2004

In 1999 the President's (Clinton) Information Technology Advisory Committee issued a report that included the following statement:

*Software is the new physical infrastructure of the information age. It is fundamental to economic success, scientific and technical research, and national security. The Nation needs robust systems, but the software our systems depend on is fragile.*

*Software fragility is its tendency not to work properly—or at all—for long enough periods of time or in the presence of uncontrollable environmental variation.*

*Fragility is manifested as unreliability, lack of security, performance lapses, errors, and difficulty in upgrading. ...*

*We have become **dangerously dependent on large software** systems whose behavior is not well understood and which often fail in unpredicted ways.*

**[PITAC 99]** President's Information Technology Advisory Committee. *Information Technology Research: Investing in Our Future*. <[http://www.hpcc.gov/pitac/report/pitac\\_report.pdf](http://www.hpcc.gov/pitac/report/pitac_report.pdf)> (1999).

# Software Safety Critical: un tragico esempio

Technical Note CMU/SEI-2004-TN-016

Charles B. Weinstock, John B. Goodenough, John J. Hudak; May 2004



In 1996 the maiden flight of the European Space Agency's Ariane-5 heavy-lift rocket ended in failure.

# Software Safety Critical: un tragico esempio

Technical Note CMU/SEI-2004-TN-016

Charles B. Weinstock, John B. Goodenough, John J. Hudak; May 2004

- This failure occurred in spite of the effort that went into making the **system dependable**.
- The hardware was **redundant** and the relevant software, **certified as trustworthy** during the successful development of the Ariane-4, was **reused unchanged**.
- Indeed, it was not considered wise to change software that had worked well.
- However, Ariane-5 had a significantly different flight envelope than did Ariane-4 and an unhandled software exception caused the rocket to self-destruct.
- This exception resulted from an overflow that occurred during the conversion of a 64-bit floating-point number to a 16-bit signed integer value.
- **The error was missed at several stages of development**.
- It was not caught in unit testing because no trajectory data was provided in the requirements.
- The error was not caught in integration testing because such testing was considered to be difficult and expensive, and **the software was considered reliable**.
- The error was not caught by inspection because the implementation assumptions were not documented.
- This is but one of many examples of software problems that could have been prevented had sufficient attention been paid to the details. However, there are lots of “details” in a large system, and it is not always obvious which ones are important to the dependable operation of the system.
- Furthermore, it is difficult to keep track of all of the details even if you can identify them. .

# Introduzione al Problema

Normalmente in campo elettronico, si tende a sfruttare al massimo l'ultimo grido tecnologico.

Quando si ha a che fare con sistemi "delicati" dobbiamo sempre porci la seguente domanda:

*" quali tecniche preventive stiamo adottando:  
(1) per evitare di introdurre degli errori di progetto  
(2) e per avere quei margini di sicurezza per cui il sistema rimane "stabile" anche  
in condizioni non previste - e non prevedibili - a priori ?"*

Specialisti di tecnologie devono sapere cosa vuol dire lavorare con dei vincoli diciamo "normativi" che sulla base delle esperienze passate dettano una serie di comportamenti progettuali adatti.

E' questa una situazione tipica in aree di ingegneria piu mature (civile, navale, meccanica, elettrica ecc) dove, per esempio:

- sapere progettare una struttura iperstatica deve essere congiunto con l'applicazione di opportuni criteri di sicurezza, con regole antisismiche delle costruzioni civili ecc.
- sapere disegnare un traghetto veloce deve essere affiancato alle regole di sicurezza marittima del RINA
- saper fare una macchina da 300 km/h deve andare a compromessi con i criteri di viabilita' e del codice stradale

## **L'alternativa e' disporre di un prodotto come:**

- una formula uno ad alto rischio,
- un transatlantico come il Titanic che non ha abbastanza scialuppe.



# Introduzione ai Sistemi Safety Critical

# Sistemi Safety Critical

## Sicurezza/Safety & Affidabilita'

**Safety; sicurezza nell'utilizzo di un oggetto**  
*la capacita' di un sistema di essere affidabile rispetto a modalita' critiche di comportamento*

**Security; protezione del contenuto**  
*la capacita' di un sistema di essere protetto da malfunzionamenti intenzionali*

### La Safety ha una definizione probabilistica

Safety. La probabilita' che un sistema non subisca avarie con possibili conseguenze catastrofiche

Il concetto di **Sicurezza/Safety** viene sempre accompagnato dal concetto di **Affidabilita'**.

Tuttavia, esiste una differenza significativa fra i due concetti:

- **l'affidabilita'** e' la probabilita' che un sistema funzioni correttamente (porti a termine il compito per cui e' progettato) per un stabilito intervallo di tempo in condizioni operative stabilite. (ISO 9126)
- **la safety** e' la probabilita' che il prodotto non devii significativamente nel suo comportamento in condizioni complementari a quelle previste

⇒ Deviazioni anche significative dallo scenario operativo di riferimento non innescano comportamenti indesiderati (mishaps).

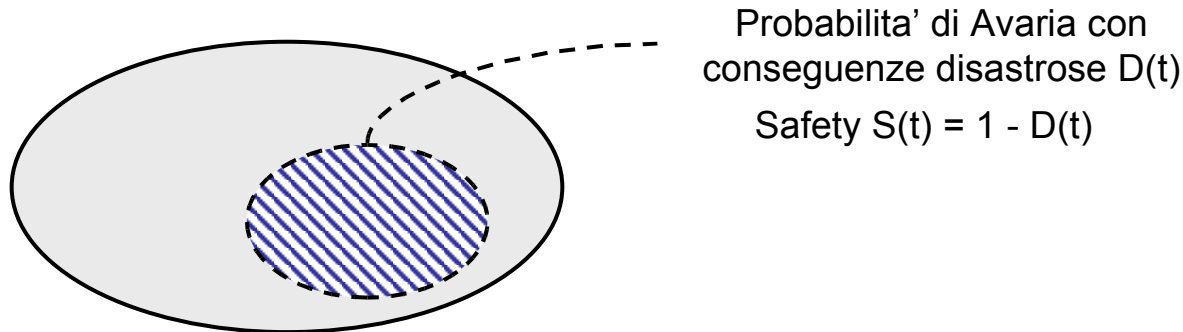
# Sistemi Safety Critical

## Safety & Affidabilita'

Il concetto di **Safety** a volte puo' essere meno restrittivo rispetto al concetto di **Affidabilita'**.

- Una presa di corrente a cui e' stata tolta potenza elettrica e' certamente "Safe"  
.....ma non riuscirà a portare a termine il compito per cui e' stata progettata

In un certo senso la **Safety** sembra imporre dei vincoli meno stringenti rispetto a quelli di **Affidabilita'**:



Probabilita' di Avaria  $F(t)$

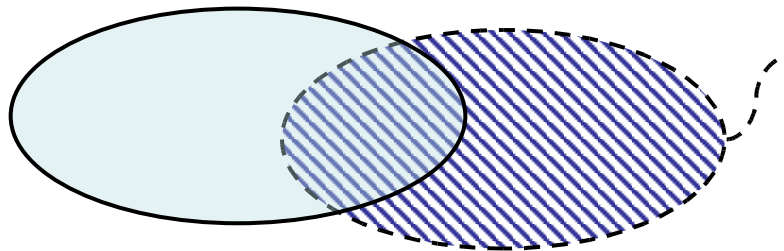
Affidabilita'  $R(t) = 1 - F(t)$

# Sistemi Safety Critical

## Safety & Affidabilita'

In realta' per la definizione stessa di Affidabilita' (la probabilita' che un sistema funzioni correttamente per un stabilito intervallo di tempo) fa riferimento a "condizioni operative stabilite"

Ma non e' detto che le condizioni operative stabilite dal progetto corrispondano alle reali condizioni operative del sistema mentre lavora

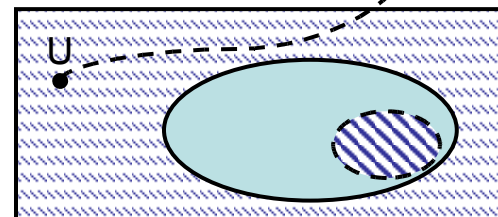


Probabilita' di Avaria con conseguenze disastrose  $D(t)$

considerando tutti i possibili utilizzi !

utilizzi non specificati e quindi potenzialmente disastrosi

Probabilita' di Avaria  $F(t)$ ; malfunzionamento rispetto alle specifiche



utilizzi specificati potenzialmente disastrosi

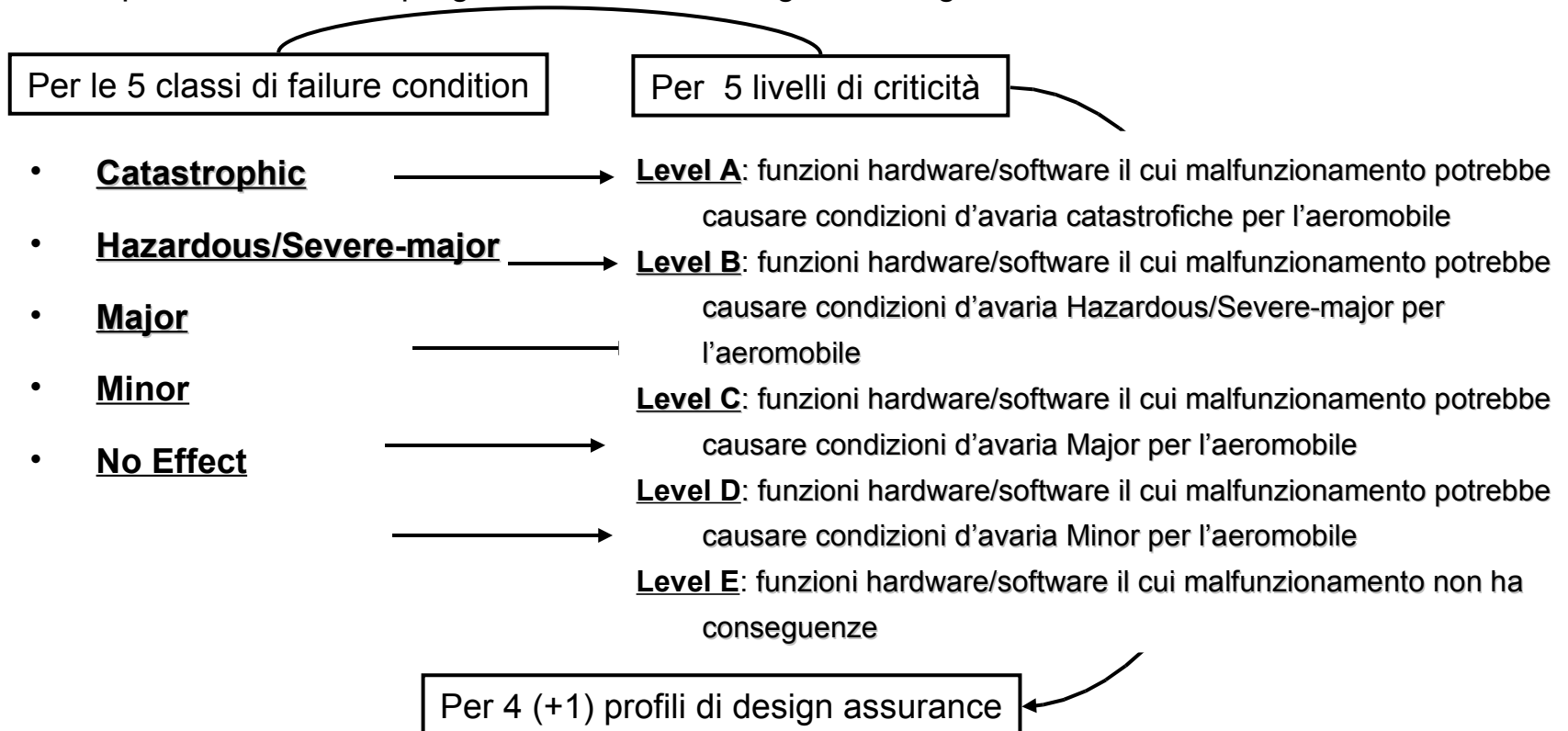
utilizzi specificati sicuri

# Sistemi Safety Critical

## Analisi di Safety

E' quindi necessario:

- Classificare gli eventi d'avaria per gravita';
- Associare ad ogni classe un livello di criticita' sull'insieme delle funzioni coinvolte;
- Per ogni livello di criticita' definire un profilo di obiettivi di *design assurance* per prevenire errori di progettazione con conseguenze negative sul sistema.



# Sistemi Safety Critical

## Analisi di Safety

Questi processi di System Safety Assessment sono orientati a:

- stabilire gli obiettivi di safety sul sistema



**System Development Assurance Process**

- determinare che l'insieme delle singole funzioni sia coerente con il raggiungimento degli obiettivi di safety sul sistema



**Da obiettivi di safety a safety requirements sui singoli equipaggiamenti**

- stabilire gli obiettivi di controllo sullo sviluppo in funzione del livello di safety da raggiungere



**Design Assurance Level (DAL) – DO-178B e DO-254**  
**Safety Integrity Level – SIL – secondo Def-Stan-00-56**

# Sistemi Safety Critical

## Analisi di Safety

### Da obiettivi di safety a safety requirements sui singoli equipaggiamenti

Supponiamo che per un particolare sistema si stimi un'affidabilità  $R$

Esempio se  $R=0,99$  si prevede l'occorrenza di un'avaria ogni 100 manovre.

#### Conseguenze:

- La conseguenza dell'avaria (che non e' nulla) deve essere tale da non pregiudicare funzioni vitali

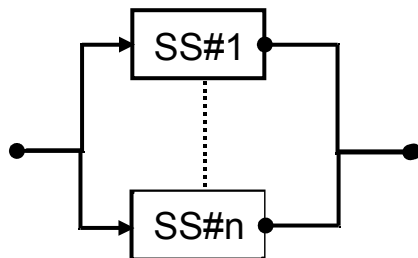


#### Definizione della classe di failure condition

- se il sistema e' composto da  $n$  sotto-sistemi, questi dovranno comporre la loro singola affidabilità  $\rho$  in maniera coerente.



$$R = \prod_k \rho_k = \rho^n \quad \text{in cascata (AND condition)}$$



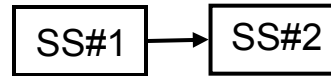
$$R = 1 - \prod_k (1 - \rho_k) = 1 - (1 - \rho)^n \quad \text{in parallelo (OR condition)}$$

# Sistemi Safety Critical

## Analisi di Safety



Motori: questi dovranno comporre la loro singola affidabilità  $\rho$



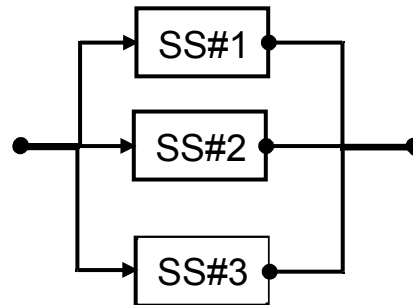
in cascata (AND condition)

il sistema deve avere sempre **entrambi** i motori funzionanti

$$R = \prod_k \rho_k = \rho^2$$

Per  $\rho = 0,9999$   $R = 0,9998$

### Condizione d'avaria catastrofica



in parallelo (OR condition)

il sistema e' triplamente ridondato

$$R = 1 - \prod_k (1 - \rho_k) = 1 - (1 - \rho)^3$$

Per  $\rho = 0,99$   $R = 0,999999$

### Condizione d'avaria grave ma recuperabile

# Sistemi Safety Critical

## Conseguenze di un'Avaria

La classificazione delle conseguenze e' un punto fondamentale per la definizione della sicurezza

In campo aeronautico, si considerano 5 classi di avaria (failure condition) cosi' descritte:

**Catastrophic:** condizione d'avaria che potrebbe pregiudicare la sicurezza del volo e dell'atterraggio

**Hazardous/Severe-major:** condizione d'avaria che potrebbe ridurre:

- la funzionalità dell'aeromobile per una **drastica** riduzione nei margini di sicurezza; o
- la capacità dell'equipaggio ad ovviare condizioni operative avverse per affaticamento sulle operazioni richieste

**Major:** condizione d'avaria che potrebbe ridurre:

- la funzionalità dell'aeromobile per una significativa riduzione nei margini di sicurezza o
- la capacità dell'equipaggio ad ovviare condizioni operative avverse per incremento del carico di lavoro sulle operazioni richieste.

**Minor:** condizione d'avaria che non riduce significativamente i margini di sicurezza ne' richiede all'equipaggio operazioni al di là del loro addestramento

**No Effect:** condizione d'avaria che non inficia la funzionalità dell'aeromobile ne' richiede all'equipaggio operazioni particolari

Esistono 3 processi di system safety assessment

- **FHA:** Functional Hazard Assessment
- **PSSA:** Preliminary System Safety Assessment
- **SSA:** System Safety Assessment



# Gli Aspetti relativi allo Sviluppo Software di Sistemi Safety-Critical

# Sviluppo Software Safety Critical

## La catena di eventi negativi

**Definizioni** [vedi per esempio Laprie].

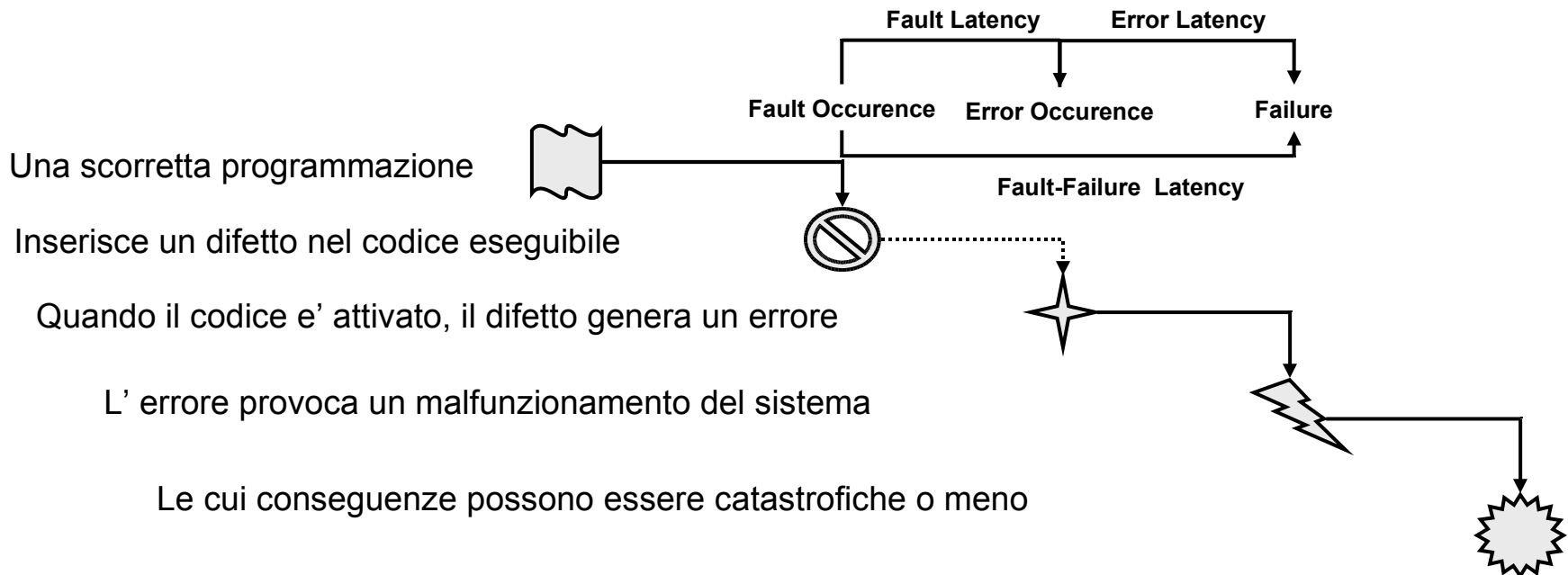
Avaria (Failure). E' la deviazione del sistema dal comportamento specificato (desiderato)

Errore (Error). Inconsistenza fra lo stato specificato di sistema e quello effettivo.

→ Un errore provoca un'avaria

Difetto (Fault). Causa presunta di un errore

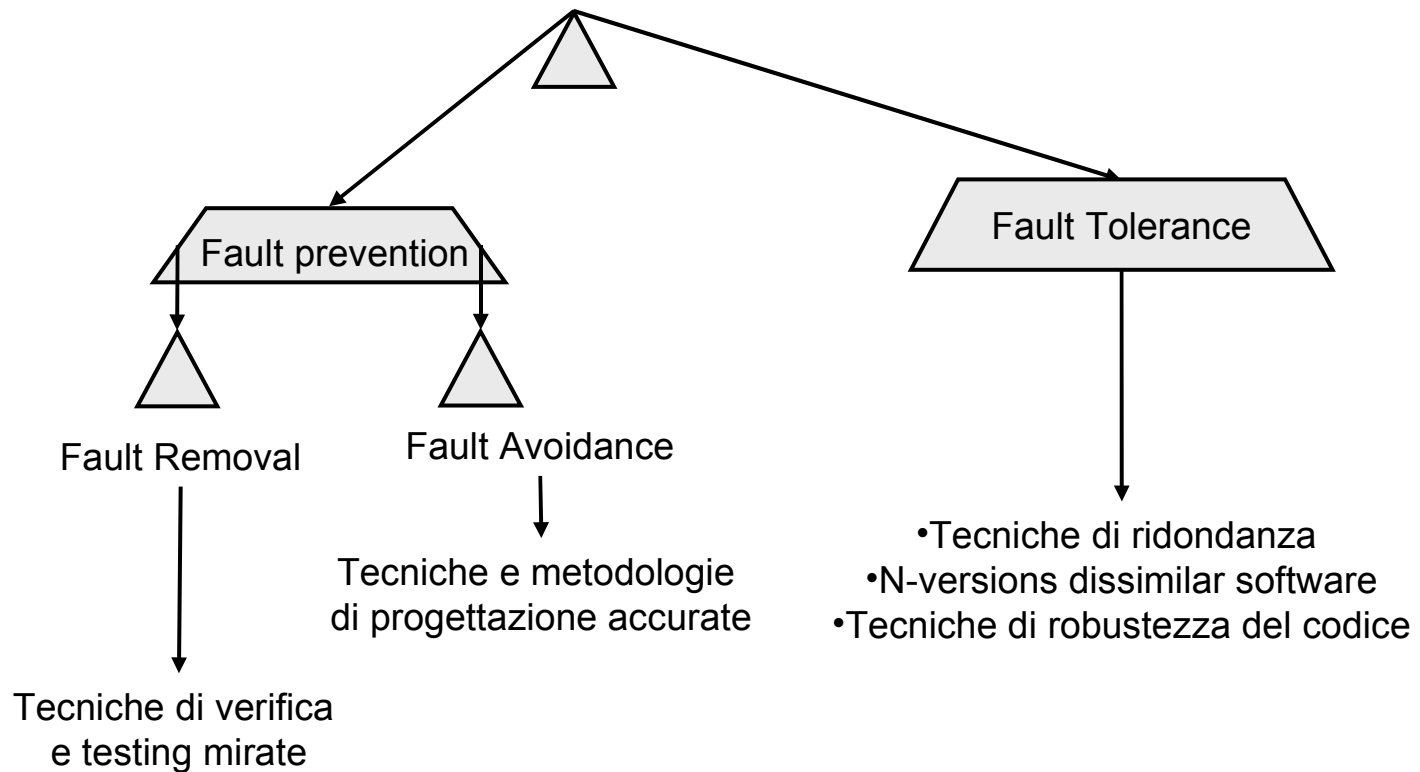
Incidente (Mishaps). Conseguenze di un'avaria



# Sviluppo Software Safety Critical

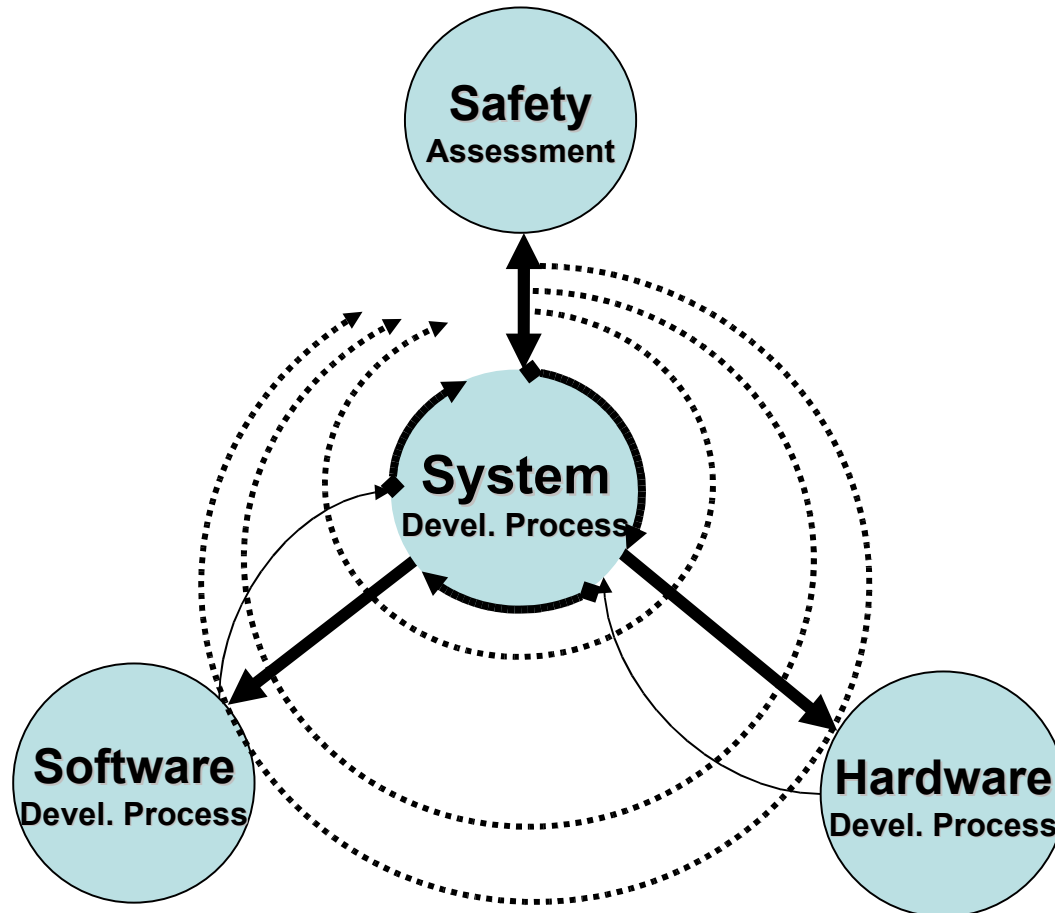
## La catena di eventi negativi

Per spezzare questa catena di eventi negativi una corretta condotta di progetto per i sistemi software safety critical deve dunque basarsi su principi di:



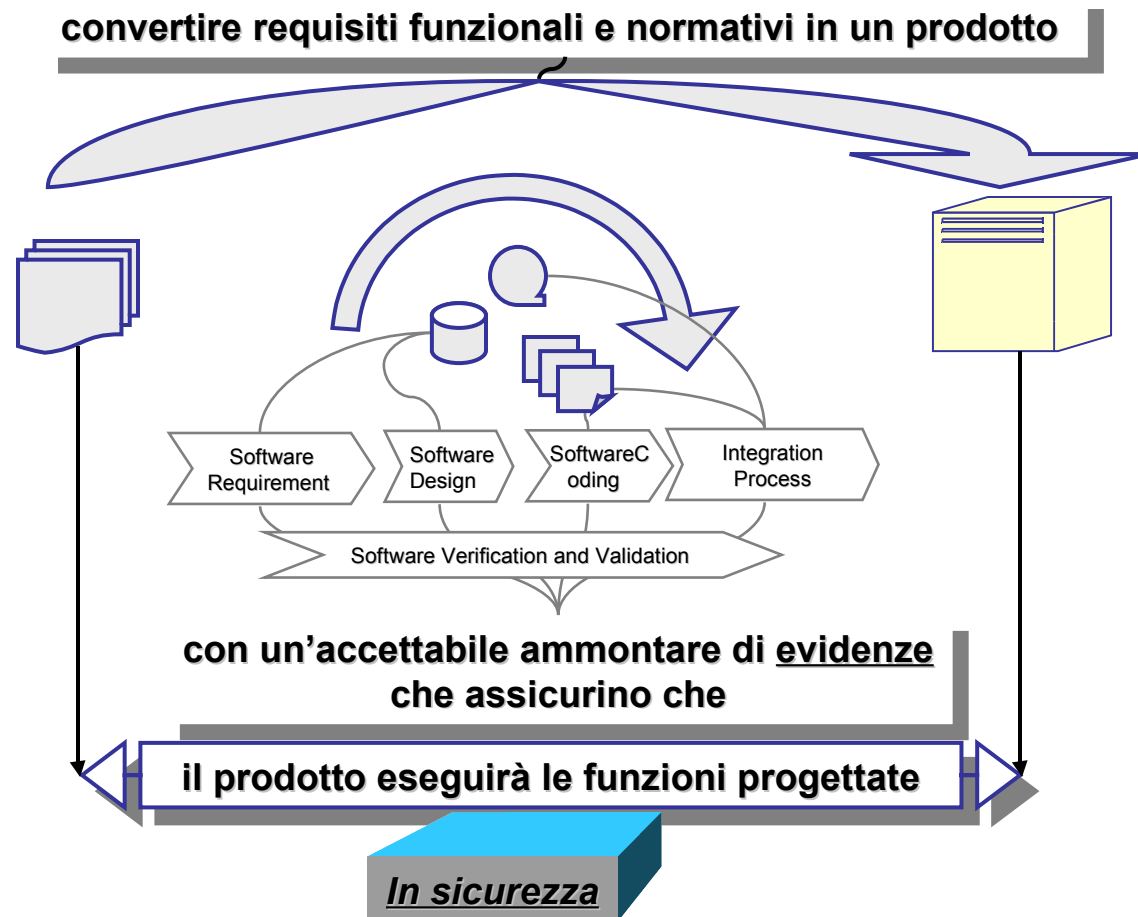
# Sviluppo Software Safety Critical

## Cicli di Vita – Modello a Spirale



# Le Normative di Progetto sui Processi di Sviluppo

Gli obiettivi dei processi di sviluppo del software sono sintetizzabili come segue:





# Le Normative di Progetto

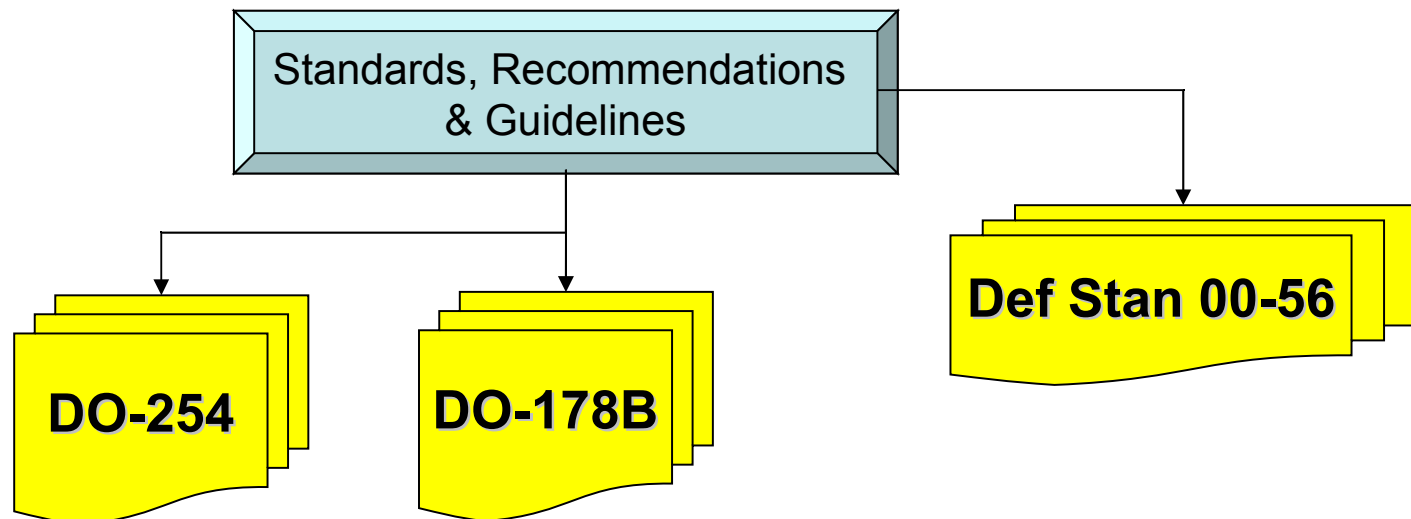
# Sviluppo Software Safety Critical

## Le normative di Progetto

La certificazione di un prodotto ha senso in un ambito normativo.

Tale normativa riflette lo stato dell'arte per evitare comportamenti progettuali "pericolosi" , nel senso della produzione per un dato livello di safety seguendo una precisa Design Assurance

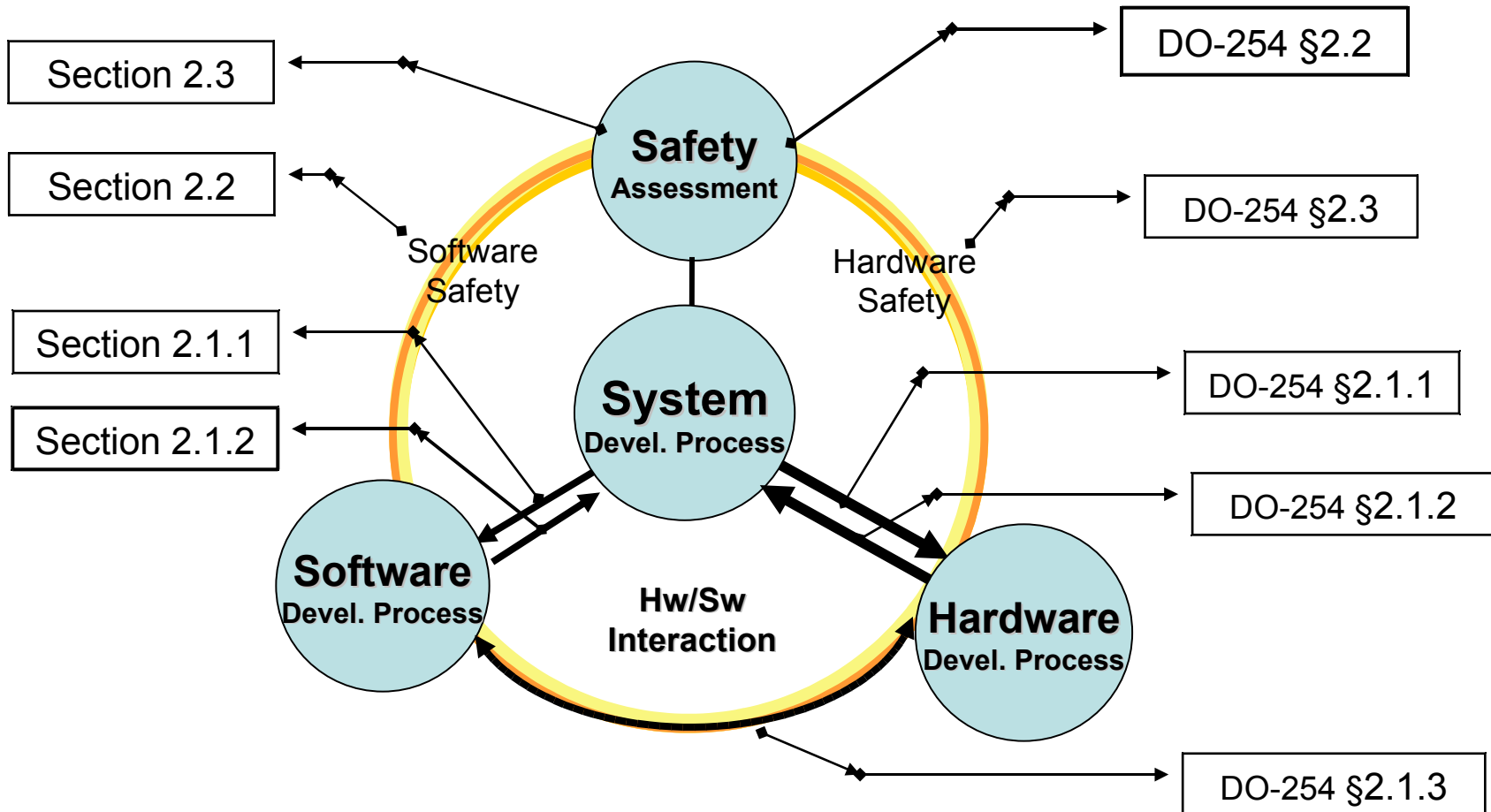
Fondamentale e' il fatto che il rigore della certificazione non deve essere sempre quello massimo (over-design), ma deve essere modulato in funzione del livello di safety appropriato.



# La Norma DO-178B – Sezione 2

## System Aspects of S/W Design Assurance

La sezione 2 e' quella che stabilisce le interazioni fra i vari processi di sviluppo:  
software **v** System **S** hardware **S** software



# Obiettivi di Processo

**La DO-178B fissa una serie di obiettivi da soddisfare**

**Questi sono riferiti all'impostazione dei processi di:**

- **Pianificazione - Software Planning Process**
- **Sviluppo - Software Development Process**
- **Configurazione – S/W Configuration Management Process**
- **Qualità - Software Quality Assurance Process**
- **Certificazione - Certification Liaison Process**

**e alla verifica dei relativi output dai processi di:**

- **Software Requirement Process**
- **Software Design Process**
- **Software Coding & Integration Process**
- **Integration Process**
- **Verification Process**

## I Dati da Produrre

**La DO-178B richiede una serie di dati da produrre come evidenze del raggiungimento degli obiettivi di Design Assurance**

A differenza degli standard basati su DRL (document-driven) dove tutti i progetti devono produrre le stesse tipologie di documentazione, nella DO-178B le evidenze sul processo sono mostrate:

- attraverso documenti a contenuti fissati ma con struttura libera
- attraverso dati estratti da qualche database

ma soprattutto la tipologia dei dati da esporre dipende:

- Dal livello di design assurance prescelto (Level A,B,C,D)
- Dagli obiettivi di processo richiesti per quel livello di design assurance

Passaggio da standard document-driven a guideline process-driven

# Obiettivi di Processo per Livello di Design Assurance

**L'ammontare degli obiettivi di processo da soddisfare, verificare e riportare con adeguate evidenze dipende dal livello di Design Assurance.**

Level A: 66 obiettivi

Level B: 65 obiettivi

Level C: 58 obiettivi

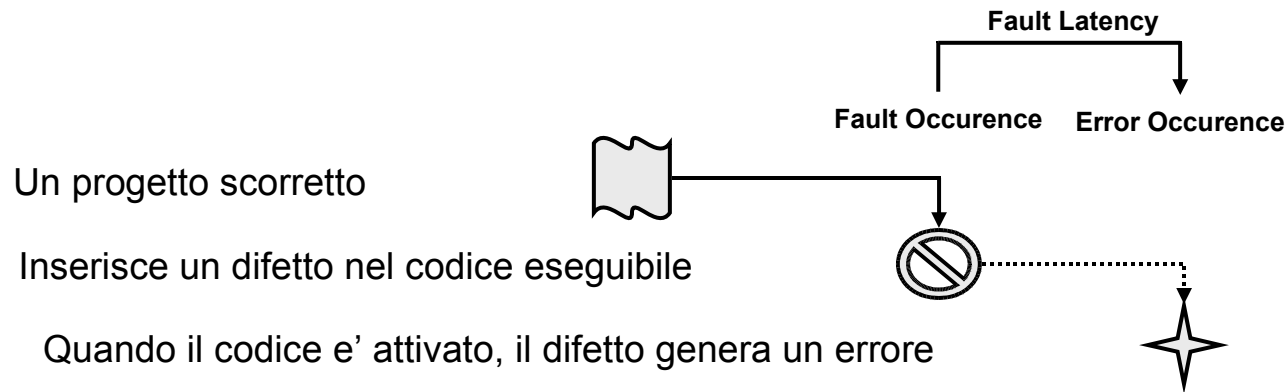
Level D: 28 obiettivi

Con gradi di indipendenza  
fra progettista e verificatore  
via via crescenti

# La Norma DO-178B

## Tool di Sviluppo

**Negli ambienti di sviluppo moderni moltissime attività di progetto per software sono attualmente eseguite per mezzo di strumenti informatici che coprono dal disegno concettuale fino alla sintesi di codice sorgente o alla simulazione comportamentale del sistema.**



**Diventa quindi critico accertarsi che gli strumenti automatici non siano essi stessi affetti da errori nella produzione dei lavorati su cui si basera' tutta l'implementazione**

# Tool di Sviluppo: Valutazione E Qualifica

**Rischio da evitare: gli output di questi strumenti contengano un errore che si rifletta in un difetto implementativo che possa causare un avaria**

**La qualifica dei tool e' richiesta quando dei processi descritti dalla norma sono**

- **eliminati,**
- **ridotti o**
- **automatizzati**

**ricorrendo all'uso di strumenti e senza che il loro output sia sottoposto a verifica come descritto dalla sezione 6**

**Obiettivo del processo di valutazione e qualifica dei tool e' di assicurarsi che lo strumento sia capace di fornire la stessa confidenza dei processi eliminati,ridotti o automatizzati**

# La Norma DO-178B

## La Valutazione e Qualifica Dei Tool

***Software tool*** possono essere classificati come:

Tool di sviluppo software

tool il cui prodotto in uscita entra a far parte del sistema avionico e che possono introdurre difetti (fault)

Esempio: un generatore di codice.

Il tool deve essere qualificato se il codice generato non e' sottoposto a verifica

Tool di verifica software

tool che non possono introdurre difetti. Ma che possono fallire il loro rilevamento

Esempio: un instrumentatore di codice per la copertura strutturale

Il tool deve essere qualificato se l'output generato non e' sottoposto a verifica da un'altra attivita'

**Solo tool deterministici possono essere sottoposti a qualifica allo stesso livello del software prodotto attraverso l'emissione di:**

- un ***Tool Qualification Plan***
- un documento di ***Tool Operational Requirements***
- evidenze di conformita' dello strumento con il TOR
- coperture strutturali e di requisito nonche' test di robustezza appropriati per il livello identificato
- un ***Tool Accomplishment Summary*** che riporti le evidenze del raggiungimento degli stessi obiettivi del Design Assurance Level del software prodotto

# Domande

# Sistemi Safety Critical

## Safety & Affidabilita'

Il concetto di **Sicurezza/Safety** viene sempre accompagnato dal concetto di **Affidabilita'**.

Tuttavia, esiste una differenza significativa fra i due concetti:

- **l'affidabilita'** e' la probabilita' che il prodotto si comporti come progettato nelle condizioni previste
- **la safety** e' la probabilita' che il prodotto non devii significativamente nel suo comportamento in condizioni complementari a quelle previste

Potremmo esemplificare come segue:

- la nitroglicerina e' un prodotto altamente affidabile perche', progettata per esplodere, sicuramente lo fa in un ambito di condizioni largamente verificabile
- la dinamite e' un prodotto sicuro (safe)
- il plastico e' ancora piu' sicuro perche', progettato per esplodere, lo fa solo in un cono di condizioni ristretto

⇒ Deviazioni anche significative dallo scenario operativo di riferimento non innescano comportamenti indesiderati (mishaps).

# Sistemi Safety Critical

## Analisi di Safety secondo Def-Stan 00-56

Formulation of the Safety Analysis Tables

Table 1

Example Equivalent Numerical Probabilities

<b>Probability</b>	<b>Numerical Equivalent</b>
Frequent	$10000 \times 10^{-6}/\text{operating hour}$
Probable	$100 \times 10^{-6}/\text{operating hour}$
Occasional	$1 \times 10^{-6}/\text{operating hour}$
Remote	$0.01 \times 10^{-6}/\text{operating hour}$
Improbable	$0.0001 \times 10^{-6}/\text{operating hour}$
Incredible	$0.000001 \times 10^{-6}/\text{operating hour}$

Table 3

Probability Ranges

<b>Accident Frequency</b>	<b>Occurrence</b> during <i>operational life</i> <i>considering all instances of the system</i>
Frequent	Likely to be continually experienced
Probable	Likely to occur often
Occasional	Likely to occur several times
Remote	Likely to occur some time
Improbable	Unlikely, but may exceptionally occur
Incredible	Extremely unlikely that the event will occur at all, given the assumptions recorded about the domain and the system

# Sistemi Safety Critical

## Analisi di Safety secondo Def-Stan 00-56

### Formulation of the Safety Analysis Tables

Table 2  
Accident severity categories

<b>Category</b>	<b>Definition</b>
Catastrophic	Multiple deaths
Critical	A single death; and/or multiple severe injuries or severe occupational illnesses
Marginal	A single severe injury or occupational illness; and/or multiple minor injuries or minor occupational illnesses
Negligible	At most a single minor injury or minor occupational illness

Table 4  
Risk Class Definitions

<b>Risk class</b>	<b>Interpretation</b>
Class A	intolerable
Class B	undesirable, and shall only be accepted when risk reduction is impracticable
Class C	tolerable with the endorsement of the Project Safety Review Committee
Class D	tolerable with the endorsement of the normal project reviews

# Sistemi Safety Critical

## Analisi di Safety secondo Def-Stan 00-56

### Formulation of the Safety Analysis Tables

Table 5  
Example Risk Classification Scheme

	<b>Catastrophic</b>	<b>Critical</b>	<b>Marginal</b>	<b>Negligible</b>
<b>Frequent</b>	A	A	A	B
<b>Probable</b>	A	A	B	C
<b>Occasional</b>	A	B	C	C
<b>Remote</b>	B	C	C	D
<b>Improbable</b>	C	C	D	D
<b>Incredible</b>	C	D	D	D

Table 6  
Claim Limits

Safety Integrity Level	Minimum failure rate
S4	Remote
S3	Occasional
S2	Probable
S1	Frequent