An Overview of IEC 61508 on E/E/PE Functional Safety

Peter B. Ladkin
Causalis Limited and University of Bielefeld

What IEC 61508 is about, how it is standardised, how used

The International Electrotechnical Commission is the organisation which develops and sets international standards in electrotechnical engineering areas. In 1997 the IEC published the standard IEC 61508, *Functional safety of electrical/electronic/programmable electronic safety-related systems*. The phrase "*electrical/electronic/programmable electronic"* is cumbersome and is often shortened to *E/E/PE*, which some pronounce "*ee-ee-pee-ee"* and others such as myself "*eepee*".

The standard has been slowly making its way into the safety-critical and safety-related digital systems mainstream over the last decade. The British Health and Safety Executive (HSE) uses it as a measure of whether a safety-critical system has reduced risk "*as low as reasonably practicable*", a requirement of English law, otherwise known as ALARP. The terminology ALARP refers back to a judgment by Lord Asquith of Bishopstone in the English case Edwards vs. National Coal Board in 1949 in which Lord Asquith also issued guidance as to the meaning of the phrase "reasonably practicable". It is intended that there shall be sector-specific norms deriving from IEC 61508, which will take precedence in those sectors for which they exist. So, for example, the European Committee for Electrotechnical Standardisation (Europäisches Komitee für elektrotechnische Normung) CENELEC issues standards for railway automation, such as EN 50128, on software for railway control and monitoring systems which are based on, but which supercede, the requirements of IEC 61508.

Definitions of Basic Concepts

Definitions devised by engineers are unfortunately not generally of the same quality as those used by mathematicians, logicians or philosophers. However, they do give some idea of what concepts are being addressed by a standard, and IEC 61508 is no exception.

- **Harm** is physical injury or damage to the health of people either directly, or indirectly as a result of damage to property or to the environment

Harm is the basic notion of what you don't want, and the basis for explaining a notion of safety as, say, absence of harmful events, although, s we shall see, this is not how IEC 61508 does it. There is no notion of accident as a harmful event in IEC 61508, but rather

- **Hazardous event:** a hazardous situation which results in harm

The standard does not explain what is meant by a situation. Given we have a definition of *state*, it is tempting to identify a situation as a state, since this is how the word is used in normal everyday language. However, for us an *event* is a pair of states, so an event logically cannot be a state (pairs cannot be identical with their elements according to the usual set-theoretic logic) and it would follow that an event logically cannot be a situation, if situations were to be states. But the definition identifies hazardous *event* with a certain type of hazardous *situation*, so it seems as if situations are events here. The standard defines

- **Hazardous situation:** a circumstance in which a person is exposed to hazard(s)

We are not much nearer, though, since the term *circumstance* is not defined. Let us therefore not set too much store by the word "situation" or "circumstance", and take a *hazardous event* to be an event from which harm causally results. The question arises how long or convoluted the causal chain is to be between hazardous event and harm, and what other factors may come into play, and how much. For example, I ride my bicycle from my driveway onto the roadway, and as I do so a car comes around the corner on my side of the road and wipes me out. Riding my bicycle onto the roadway is certainly an event, and it did causally result in harm, by the Counterfactual Test, since if I hadn't done it I wouldn't have been wiped out. But do we really want to call it a hazardous event and make it the center of focus? Surely the event that should be at the center of focus as unwanted is the fact that car and bicycle met in the same physical place at the same time and harm directly resulted from the collision: what we would usually call an accident. And then you can ask how this might have happened causally, and trace precursor events. But we cannot do this with the IEC 61508 vocabulary, which allows us no accidents; only hazardous events, and provides no guidance as to how causally "near" an event must be to harm to count as a "hazardous event". In fact, the state of the usage is that most engineers of my acquaintance who are conversant with the standard would use the phrase "hazardous event" as equivalent to "accident or a direct precursor of an accident". So that is probably the best way to go.

As we have seen, hazardous events are hazardous situations and these are circumstances in which one is exposed to a hazard, which is

- **Hazard:** a potential source of harm

We know what harm is, but not what it is to be a *source* of harm, let alone a *potential source*. As the philosopher and logician W.V.O. Quine once asked, exactly how many potential fat men are standing in that empty doorway? There are long-standing difficulties with the coherence of notions of potential-this-and-thats. Let us look therefore at the notion of *source*. I guess a source is some kind of origin. One may guess further that what is meant is a causal origin, i.e., a cause. We can put this together with the common usage of the term "hazard" to refer to a state out of which sometimes harm occurs, but not inevitably, and guess that maybe this is what is meant by "potential": not inevitable. We are familiar from Chapter XXXX with a notion of hazard as some state from which an accident will result if the environment is unfavorable, and our current notion looks to be similar. We can imagine that a hazard might be a state, or an event, which contains a significant set of causal factors of an accident, but which causal set is not necessarily sufficient for an accident to occur, in other words that other causes have to participate. The point of the term "significant" is to indicate something like "a lot", or "almost all"; that is, that you have a lot of causal factors present in the hazard and it only requires one or two more, maybe under your control but maybe not, to "tip the scales" and result in an accident, that is, harm.

That is the core set of concepts in IEC 61508 which talk about what we would call accidents. We have not get seen a definition of safety, and that is because the standard defines safety not through accidents, but through risk.

- **Risk:** [the/a] combination of the probability of occurrence of harm and the severity of that harm

We have seen something like this before. The definition does not say how the probability and severity are "combined" but we have already discussed various proposals in technical detail and can imagine that something along those lines is meant here. Now, here comes a novelty of the IEC 61508 standard:

- **Tolerable risk:** risk which is acceptable in a given context, based on the current values of society

"Acceptable" means something which is possible to accept, or which is accepted. It is not suggested who does the accepting of the risk. Lawmakers? Politicians? Individuals exposed to harm? Other "stakeholders"? Neither is it suggested how "current values of society" might be determined, or even to what this phrase refers. 5,000+ people die in accidents involving automobiles on the road in Germany every year; 3,000+ in Britain. Is the risk of 1 in 15,000 of dying in any given year on the road in Germany "acceptable", based on the "current values of society"? The current value of road deaths is indeed 5000+, and people do drive, so it seems as if people in general in some sense do "accept" this risk. But consider the police and state campaigns to control errant driving and reduce accident totals. If 5000+ deaths per year represent the "current values of society" then it seems as if these campaigns are fighting against the "current values of society". Indeed so, and who would say it wouldn't be a good thing in this case to change the "current values of society" so that 5000+ deaths per year are no longer "acceptable?

So we can see there are lots of problems with this definition. But the basic idea is clear. The guidance as to what is an acceptable risk or not comes not from theory, but from social considerations, from asking the society at large. And however one does that, and whatever answer one receives, one works with that answer to determine "tolerable risk". And then one can define

- **Safety:** freedom from unacceptable risk

The concept "unacceptable risk" is not defined, but let us imagine that it means "risk that is not tolerable risk" (examples of such sloppiness should be trivial to correct, but it seems that in the process of deriving international standards such as this it can become almost impossibly hard). So safety is not defined in terms of absence of harm or other unwanted effects, but in terms of the absence of risk of a certain sort.

Let us reconsider the 5000+ road-deaths-per-year figure above, and the argument that this is what seems to be acceptable based on German society's current values. It then follows from the definition of safety that German roads are safe. And, *mutatis mutandis*, British roads and Greek roads and Spanish roads and Indonesian roads and everybody else's roads as well. Because the current levels of road accidents, whatever they may be, are what is - obviously - accepted in those societies. The reader may conclude for hisherself whether this notion of safety makes much sense. But it is what we have to work with when we work to IEC 61508.

Let us briefly compare with other definitions of unwanted events, say the definition of *aircraft accident* in the US Code of Federal Regulations, 49 CFR 830.2. This definition is similar to that in Annex 13 to the Convention on International Civil Aviation, usually known as ICAO Annex 13, which is longer, and also accounts for missing aircraft. 49 CFR 830.2 reads as follows:

- Aircraft accident means an occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight and all such persons have disembarked, and in which any person suffers death or serious injury, or in which the aircraft receives substantial damage.

There is here a specific definition of an event which is called an accident. In that specific event, a certain amount of harm (in the 61508 sense) occurs *or* substantial damage to the aircraft. This latter phenomenon is not covered by IEC 61508; this definition thus covers outcomes which may be fundamentally different from those covered by IEC 61508, although there are commonalities. For example, suppose after a long transatlantic flight you arrive at an airport with a shortish runway and a difficult approach over obstacles, land hard, comprising the landing gear so that some braking systems no longer function as intended, overrun the runway, and damage the aircraft enough so that it has to be scrapped, as happened to an Airbus A340 aircraft from Madrid to Quito, Ecuador on 11 November, 2007. You have had an accident according to 49 CFR 830.2, but not a "hazardous event" according to IEC 61508, since no harm resulted - very luckily. We leave it to the reader to decide whether this seems to be an appropriate selection of concepts.

Although we succeeded in interpreting the definition of "risk" in IEC 61508 in a manner consistent with other notions of risk, we were able to do so because we know, or think we know, what is meant. Let us briefly see what kind of difficulties we might have had, were we to have been ignorant of other notions of risk and have tried to derive our understanding from the IEC 61508 definition directly.

It is certain that I shall suffer some small degree of harm from using my bicycle regularly. Once every couple of months I scratch myself on something (that is harm) and that is going to continue happening. And, who knows but I hope not, I may be run over by a car sometime and killed. So the probability of occurrence of harm is 1, representing certainty (that scratch *will* occur). And according to the definition, I am supposed to combine that probability 1 with "severity of that harm". What severity? There are at least two to choose from: the trivial severity of the scratch, and the catastrophic severity of my life being lost. If I combine the probability 1 with the trivial severity (following the indication of the words "*that* harm", which suggest the harm to which the probability attaches), then my calculation of risk would lose the significant fact that I stand some chance of being killed. It would suggest that riding a bicycle on the road is similar to using a screwdriver in risk. But you don't die if you stop paying attention to the surroundings when using the screwdriver, and you well might on a bicycle. So combining the probability 1 with the trivial severity doesn't seem right. Maybe we should combine it with the catastrophe of the deadly accident? But then that would indicate a similarity to, say, jumping off a tall building on to a concrete sidewalk, high probability "combined" with catastrophic consequence. And I want to say the risk, whatever it may be, of riding my bicycle doesn't seem to me at all like the risk of jumping off a tall building onto a concrete sidewalk. So that can't be it either.

The conclusion is that we cannot take the IEC 61508 definition of risk as a good guide to what risk is. We need something like the modified de Moivre definition to make sense of the notion. Since that definition has been around, used and abused for 300 years and still going strong for financial people, it makes sense to continue to use it, even if one has to approximate it sometimes.

A good set of definitions

- defines terms before they are used
- tries to be precise
- tries to reduce or eliminate ambiguity
- tries to limit the number of concepts left undefined
- tries to be clear to the intended audience as to what is meant

From the discussion above, the reader might well conclude that IEC 61508 does not do very well on any of these criteria.

The Major Concepts of IEC 61508

IEC 61508 bases its approach on a number of fundamental, sometimes complex, concepts. It is important to understand that all systems are considered by the standard to be control systems. This follows from the observation that two specific subsystems are always present, namely

- the *Equipment under Control* (EUC). This is the subsystem consisting of the equipment providing some or all of the functions for which the system was designed. (The definition of EUC in IEC 61508 is somewhat unhelpful.)

- the *EUC Control System* (EUCCS). The control system of the equipment which is under control. The EUCCS is a "*system which responds to input signals from the process and/or from an operator and generates output signals causing the EUC to operate in the desired manner.*" The standard says further that "*the EUC control system is separate and distinct from the EUC.*"

The fundamental concepts of IEC 61508 are:

1. A detailed system life-cycle model, from initial development through decommissioning

2. The notion of *functional safety*. The standard requires the provision of so-called *safety functions* for the mitigation of risk associated with functions of the system in the cases in which this risk is deemed to be too high

3. A focus on risk *reduction*. The standard implicitly assumes that there is no such thing as zero risk - risk cannot be entirely eliminated. The safety functions are to perform the risk reduction where it is needed

4. A four-way classification of (sub)system types. Besides the EUC and EUCCS (the first two subsystem types, of which there is only one of each), there may be *Safety-Related Systems* (SRS). Besides this, there may be subsystems which are none of these.

   This classification may not be exclusive: the EUCCS may or may not be a SRS. The EUC is not an SRS.

5. The notion of *Safety Integrity Level*, or SIL, of a required function. There are four degrees of SIL, SIL 1 through SIL 4, ranging from moderately stringent to very stringent integrity

requirements (we shall see later exactly what *integrity* may be).

The System Life-Cycle

The safety life cycle of a safety-related system is defined in a task flow graph. The task graph associates 16 steps with the safety life cycle. The safety life cycle is that part of the life cycle of a system during which activities related to assuring the safety of the system take place. Other tasks appear if they are prerequisites for tasks associated with assuring the safety of the system.

The point of the safety life cycle flow graph and its tasks is to define a fixed structure within which safety-assurance activities associated with the system may fit, and thus be audited to ensure that every necessary activity or measure has been undertaken. The life cycle does not claim to be the only appropriate life cycle, as far as I know, but it is important to have precisely one defined such model to which all can adhere.

I take such lifecycle models seriously, in the sense that they are partial specifications of processes. They are fairly simple in that they subdivide development and give a precedence ordering to the parts. When such a model appears in a standard such as IEC 61508, then anyone adhering to the norm must be able to subdivide hisher development process in exactly the way specified by the Lifecycle model and exhibit it in documentation, or risk being judged as not having followed the standard, which in some cases in some countries can be a criminal offence.

Various features of the life cycle model are worth noting.

The first safety-assurance activity occurs at Stage 3, with Hazard and Risk Analysis, called by some in other standards the Preliminary Hazard Analysis. At this point, you have some idea from concept and scope definition what the system should do, functionally, albeit abstractly. It is generally agreed amongst critical-system engineers that at this point it makes sense to start figuring out all the ways in which things can go dangerously wrong. Not how it can go wrong, but how it can go *dangerously* wrong. Say we are designing a car. Then the engine may cut out on starting, and we may not want this, but this is not a dangerous failure. But if a wheel falls off at high speed, this would be a dangerous failure, as would a failure of the brakes to work when applied. Identifying all these types of dangerous failure is the activity at Stage 3.

Having performed the preliminary hazard analysis, you have some idea of how the system can go dangerously wrong. At this point you can start formulating constraints that it not go dangerously wrong in these identified ways. The wheels must be attached firmly enough that they cannot come loose, or the driver must be warned in good time of a problem. The brakes must be so designed that they cannot fail to operate moderately effectively when required, or the driver must be warned appropriately in advance. And so on. This is Stage 4 and these constraints form the *Overall Safety Requirements*.

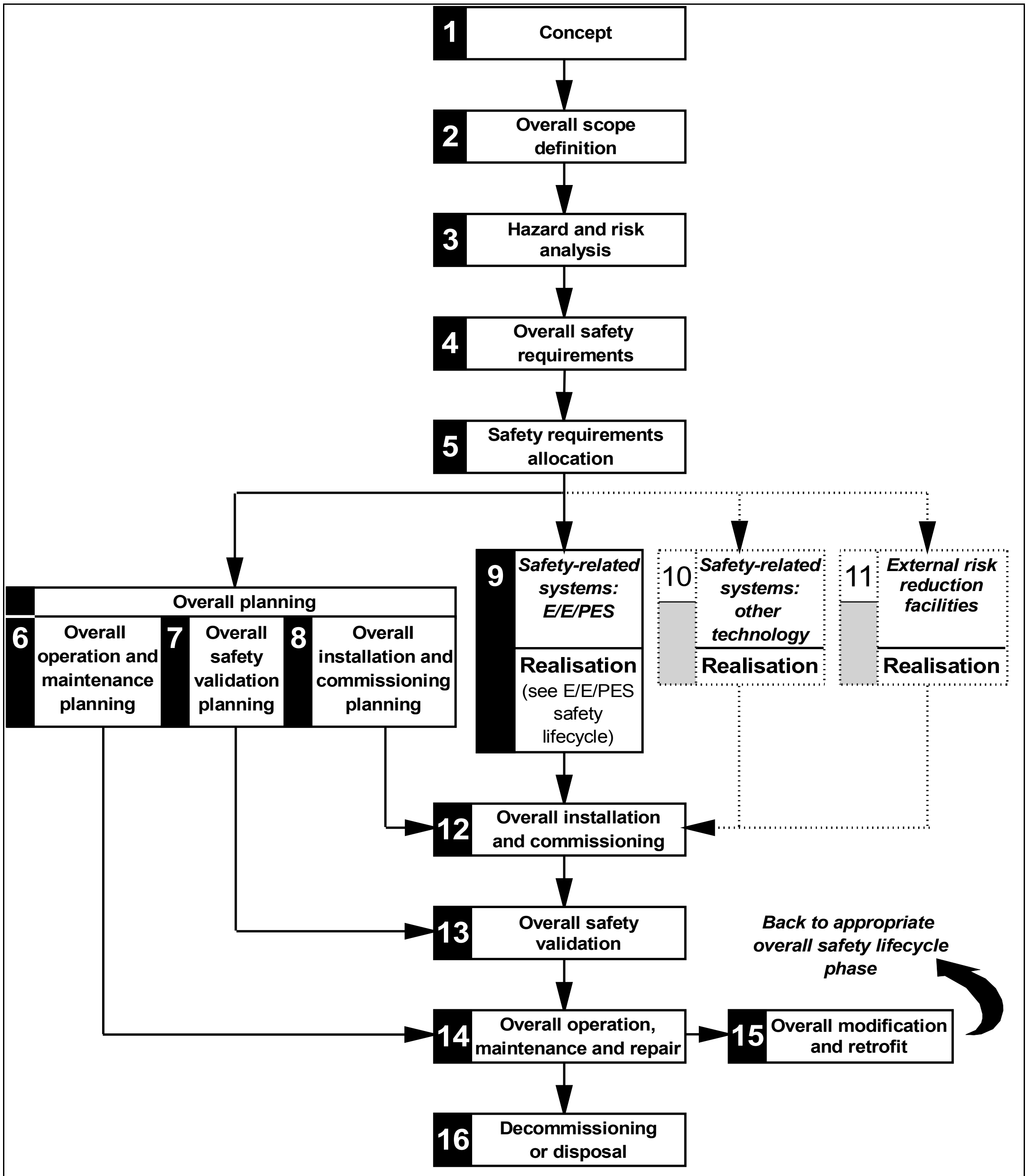
Given the Overall Safety Requirements, one may then begin to consider to which parts of the system each safety requirement must apply. The safety requirement that the wheels not fall off applies to the wheels themselves, the hubs, the axles and associated subsystems. It does not concern the fuel tank, or to the sunroof. This is Stage 5, the Safety Requirements Allocation of requirements to subsystems to which they apply.

At this point, it is understood that the requirements and requirements analysis part of system development is finished (in the very simple and abstract so-called Waterfall model of system development; in the Spiral Model of system development, which corresponds rather better to actual practice it will be revisited many times during system development). The system begins at this point to be designed in detail.

Stages 6-8 concern the overall planning. Stage 6: how will the system be installed, operated and maintained, and how will the various continuing safety requirements be assured during installation, operation and maintenance of the system? Are the wheel nuts to be installed with a locking pin? How often does the tightness of the wheel nuts need to be checked? Whether there is metal fatigue around the holes for the locking pins? How often the brake lines need to be checked for fluid leaks. And so on. Stage 7 concerns how the safety of the system is to be checked (validated). How you assure that the brake lines are sufficiently robust to resist rupture between the inspection intervals. How you assure that the wheel nuts will not work loose between inspections. Stage 8 concerns how the system is to be installed and commissioned. Here, an example of a chemical processing plant would be more appropriate than considering a car, and indeed it has been a concern of some that IEC 61508 was heavily influenced by concerns from the process industries rather than, say, the auto industry.

Stages 9-11 form the nub of the safety-design activities. We have briefly seen, above, that the standard assures safety through provision of safety functions in (sub)systems contributing to an activity which is insufficiently safe (whatever this might be; we have yet to discuss how this is determined). A (sub)system required to implement a safety function, or which contributes to the achievement of a required SIL, is called a *Safety-Related System* (SRS). Stages 9 and 10 concern the design, analysis and implementation of these SRSs, which are of two types: those involving E/E/PE components (Stage 9) and those which do not (Stage 10). IEC 61508 is concerned primarily with the E/E/PE components, so Stage 9 will be decomposed further, as we shall see. The standard acknowledges that there may be ways to reduce risk that do not involve the provision of safety functions, in our car example, say, by restricting the speed at which the car may travel, so that the wheel attachments are not subject to the kinds of forces that will work them loose. Such ways are pursued in Stage 11. The standard does not have anything further to say about Stages 10 and 11.

The further stages, Stages 12-16 are performed when the system has been built. It must be installed and commissioned (if it is a processing plant; for a car this may simply mean “sent out of the factory to a dealer”). This is Stage 12. Then the system must be checked to verify that all the required safety attributes have been identified and appropriately handled during building and installation, in Stage 13. (There is some systematic confusion amongst engineers as to what activities the terms “verify” and “validate” refer. Some have it one way, say, software verification specialists, and others use the terms exactly opposite to these meanings, say, telecommunications system engineers. I don't propose to resolve this issue of meaning here. IEC 61508 says this is “validation”, I said “verify”. So be it.) Then the system may be put into operation, and there are safety activities required during operation and maintenance. These form Stage 14. It is also foreseen that the system will be modified during operation, say to cope with an operating environment with different characteristics than foreseen at system-design time (say, the roads are more bumpy and slippery than they were at design time, and the wheel attachments and brakes have to cope), or to cope with extended functional requirements (say, the motor needed to be more powerful, or the brakes more powerful, to address the competitor's new model, and so these modifications need to be incorporated). This is then Stage 15. Finally, when the system is all clapped out, you need to dispose of it. There may be activities here concerning safety. For example, you cannot just put the complete car in a landfill. You need to remove the battery and other toxic items and dispose of them separately. And maybe you want to reuse rubber tires, and recycle the metal in some of the parts. This is the final stage, Stage 16.


1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 Safety-related systems: E/E/PES
Realisation (see E/E/PES safety lifecycle)
10 Safety-related systems: other technology
Realisation
11 External risk reduction facilities
Realisation
12 Overall installation and commissioning
13 Overall safety validation
Back to appropriate overall safety lifecycle phase
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
16 Decommissioning or disposal

NOTE 1 Activities relating to ***verification***, ***management of functional safety*** and ***functional safety assessment*** are not shown for reasons of clarity but are relevent to all overall, E/E/PES and software safety lifecycle phases.

NOTE 2 The phases represented by boxes 10 and 11 are outside the scope of this standard.

NOTE 3 Parts 2 and 3 deal with box 9 (realisation) but they also deal, where relevant, with the programmable electronic (hardware and software) aspects of boxes 13, 14 and 15.

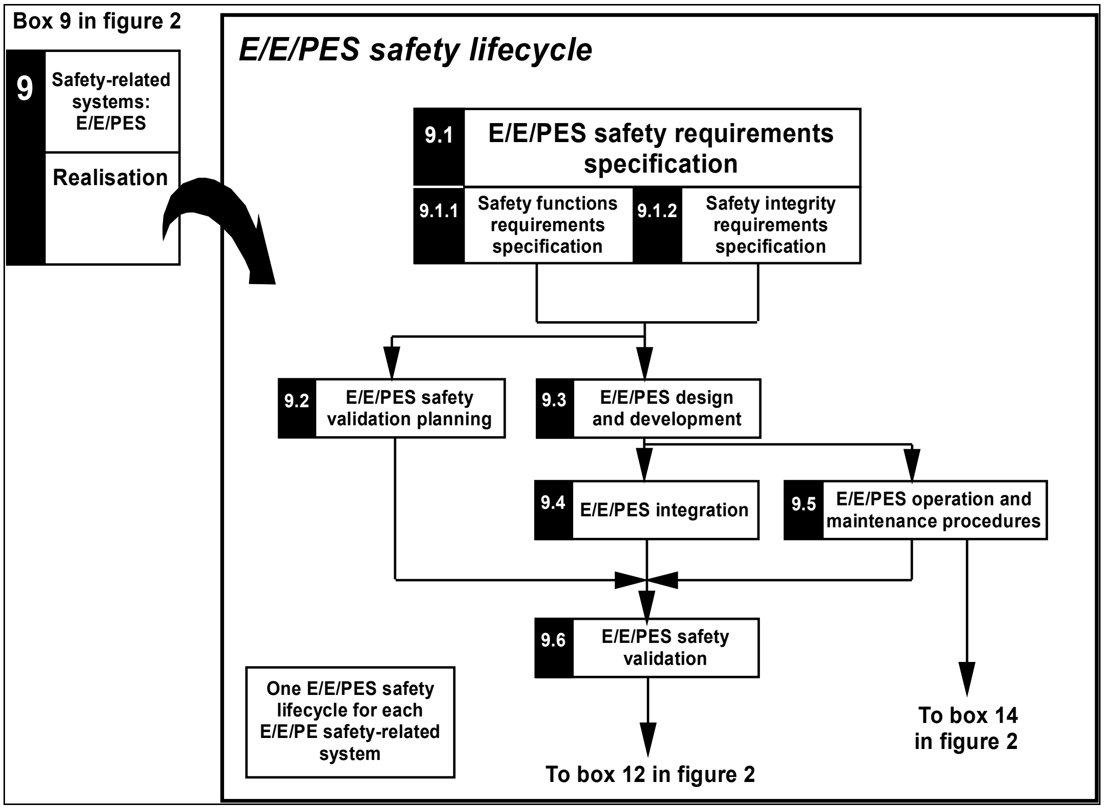
The E/E/PE and Software Safety Lifecycles

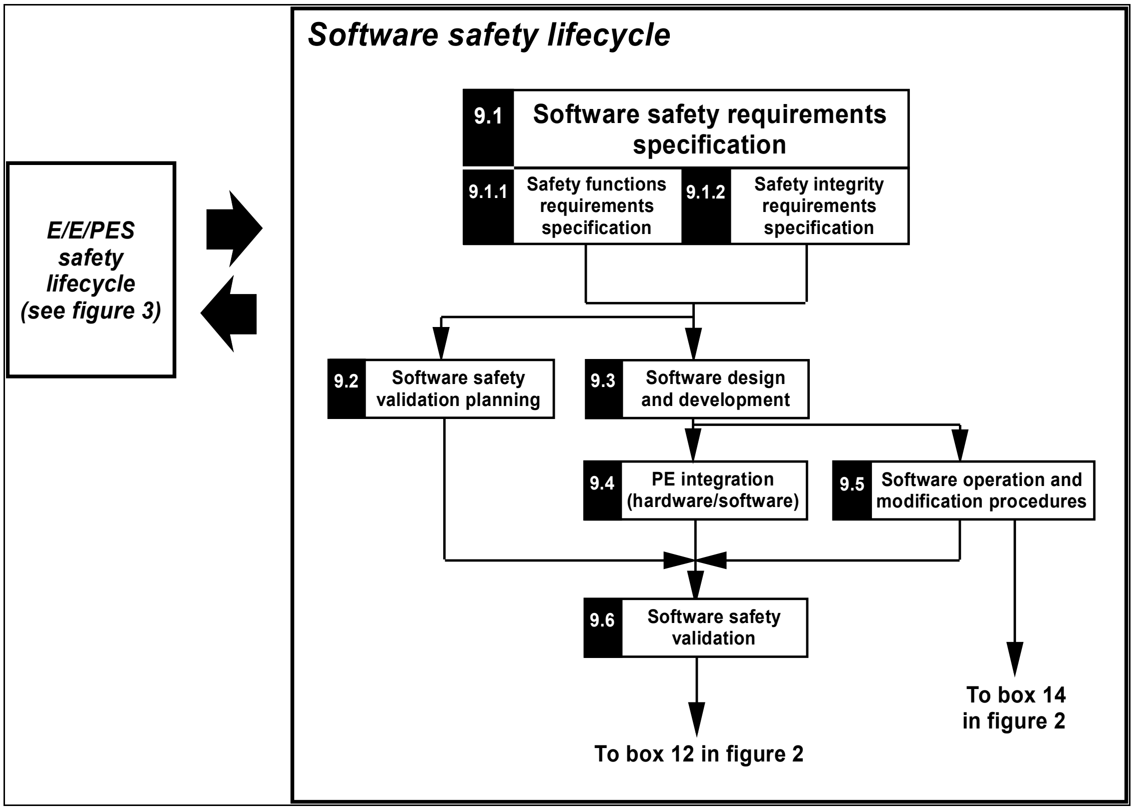
IEC 61508 defines more detailed, but still general, lifecycle stages for Stage 9 in the overall life cycle, the E/E/PE (sub)system development and the development of its software.

These mimic the overall life cycle tasks in miniature. They emphasise that along with development come specific tasks to plan for and execute the validation of the safety properties of the system or subsystem. Because this life cycle task structure is intended to apply to subsystems as well as the overall system, the integration of the subsystem in with other system parts is explicitly represented.

In the case of software, the integration activity includes the activity of "integrating" the software with the hardware, that is, of getting the code running on the target HW.

To repeat, these life cycle schemas are intended as a means of book keeping. Apart from guiding the development to ensure that no important safety-relevant tasks are inadvertently omitted, the schema plays a role in system justification. When one documents the safety activities undertaken and untertakes to demonstrate that they are sufficient to ensure the required level of safety, that is, when one prepares what is called the *safety case* for the system, then it is convenient to have such a standard task decomposition to help structure the safety case document.


Box 9 in figure 2
9
Safety-related systems: E/E/PES
Realisation
E/E/PES safety lifecycle
9.1
E/E/PES safety requirements specification
9.1.1
Safety functions requirements specification
9.1.2
Safety integrity requirements specification
9.2
E/E/PES safety validation planning
9.3
E/E/PES design and development
9.4
E/E/PES integration
9.5
E/E/PES operation and maintenance procedures
9.6
E/E/PES safety validation
One E/E/PES safety lifecycle for each E/E/PE safety-related system
To box 12 in figure 2
To box 14 in figure 2


Software safety lifecycle
E/E/PES safety lifecycle (see figure 3)
9.1 Software safety requirements specification
9.1.1 Safety functions requirements specification
9.1.2 Safety integrity requirements specification
9.2 Software safety validation planning
9.3 Software design and development
9.4 PE integration (hardware/software)
9.5 Software operation and modification procedures
9.6 Software safety validation
To box 12 in figure 2
To box 14 in figure 2

Common Software Lifecycle Stages and the IEC 61508 Lifecycle

Those developing software usually organise the activities into stages, which follow one another, yielding a so-called Waterfall Model of development.

A typical Waterfall Model might have

1. Requirements Specification and Analysis Phase. In this phase is determined what the system has to do, what functions it has to perform. And these requirements are analysed abstractly for completeness, consistency, feasibility and other desirable properties.

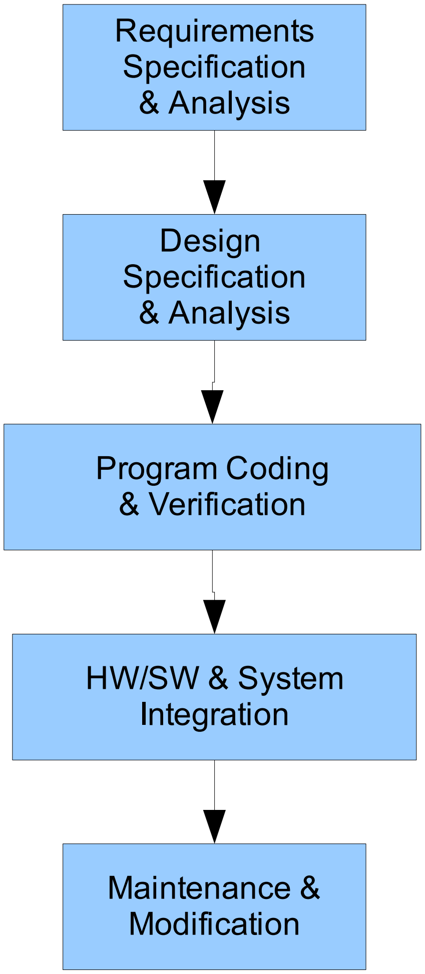
2. Design Phase. The detailed design of the system is performed. The result is a Design Specification. It is also shown in this phase how the design is to fulfil the Requirements Specification.

3. Implementation. The design is coded. This phase will usually also include verification, showing that the code indeed fulfils the Design Specification; and unit testing.

4. Integration. The code is loaded on to the hardware, and the hardware embedded in the larger system. This phase usually includes a testing task, called integration testing

5. Commissioning and Operational “Maintenance”. When the system is built and operating, modifications will usually need to be made. These are often referred to as “maintenance”, although “maintenance” is an activity that consumes most of the budget of any large project,

indeed may consume some 60-80% of it in largish projects.


Requirements Specification & Analysis
Design Specification & Analysis
Program Coding & Verification
HW/SW & System Integration
Maintenance & Modification

A Simple "Waterfall" Model of Software Development

The Waterfall Model is an idealisation. Typically, phenomena encountered in later stages will result in revisions undertaken at earlier stages. However, the later in the "Waterfall" a phenomenon is discovered, the more costly it is to fix in earlier stages (people may suggest a factor of 10 cost per backward stage, but this is only a very crude estimate, of course). The need for revisiting earlier stages led to the so-called Spiral Model of development, in which stages are explicitly revisited, one hopes with decreasing modifications, until one is finally done.

To date, there is no explicit integration of the IEC 61508 safety lifecycle model with even crude software development models such as a Waterfall or a simple Spiral Model. There is thus no guidance available for software developers how they might then integrate the IEC 61508 lifecycle model into their development processes.

Integration is not quite as easy a task as it might appear. For example, in typical system safety processes, a hazard analysis performed at requirements-specification time will be a preliminary hazard analysis. As the system becomes more concrete during the design phase, often a further hazard analysis will be performed as the interaction of the components of the design is specified. Hazards beyond those specified in the preliminary analysis almost inevitably crop up during design.

For example, suppose we are building a rail-road level-crossing (see the next section for a picture). The hazard analysis at the Requirements stage will concern functional deficits, for example

- A train approaches and the barrier stays up
- The barriers come down before the warning lights and bells activate
- A barrier comes half-way but not fully down

At the Design stage, new hazards may be identified that are not present at the Requirements stage.

- The train-sensors and the logic processor are physically separated and joined by a cable. What happens when the cable is severed or partially severed?
- Is the processor cabinet sufficiently protected against the ingress of rainwater?

At the Program Coding and Verification stage, further hazards may be identified that are not present at either Requirements or Design stages.

- The source code language is C. Does the source code contain any constructs that are ambiguous, or any that lead to non-deterministic behavior?
- The source code language is C. The C code is synthesised by a code generator from a state-machine-like "specification" language. Does the code generator always produce code which is deterministic and whose meaning exactly matches the meaning of the state machine specification from which it was generated?

At the HW/SW integration state, yet further hazards arise which were not expressible at earlier stages.

- We are using the compiler from company X. This has known weaknesses which we believe will not affect our code. Are we sure that our source code will be compiled in such a way as to preserve the meaning of our program?
- The target hardware is Y. Can the compiled code exhibit run-time errors?

So we see that at each stage of the Waterfall SW lifecycle we can identify hazards that could not be expressed in the language of the earlier stages. However, the IEC 61508 life cycle only allows for one hazard analysis stage, Stage 3. The need for many many hazard analyses as the design progresses into code and object code fits ill with this sole designated stage in the standard.

The standard does recognise the potential need for more than one hazard analysis:

- **7.4.2.1:** *A hazard and risk analysis shall be undertaken which shall take into account information from the overall scope definition phase (see 7.3). If decisions are taken at later stages in the overall, E/E/PES or software safety lifecycle phases which may change the basis on which the earlier decisions were taken, then a further hazard and risk analysis shall be undertaken. .... NOTE 2: It may be necessary for more than one hazard and risk analysis to be carried out.*

However, the standard does not explicitly recognise that such analyses are *inevitable* at each stage of the software Waterfall lifecycle (Note 2 above says "*may be necessary*"). And given that they will occur, there is no guidance on how the Safety Lifecycle model with its 13 stages needs to be modified to account for them.

To show that integrating the Safety Lifecycle with a software development model, even one as simplistic as the Waterfall model, will not be trivial, consider the following.

There is one stage for hazard and risk analysis, namely Stage 3. Hazard and risk analyses must be performed after Stages 9.3, E/E/PES design and development, and 9.4, E/E/PES integration, as we have just seen. It follows that in an integrated Safety Lifecycle/Waterfall development model an instance of Stage 3 must follow both Stages 9.3 and 9.4. Hence there must be shown a path from Stage 9.3, respectively 9.4, back to Stage 3. Let us indicate this with simple arrows, from Stage 9.3, respectively 9.4, back to Stage 3.

There are now two loops in the Safety Lifecycle graph: one loop passing from Stage 3 down to Stage 9.3 and back to Stage 3; the other passing from Stage 3 down to Stage 9.4 and back to Stage 3. These loops are nested. So when one has finished Stage 9.4, E/E/PES integration, and performs the hazard analysis (passing to Stage 3 again), then a further iteration of Stage 9.3, E/E/PES design and development, lies before us. Can this be right? Well, maybe if one is using the so-called Spiral Model of software development, in which the various stages are revisited again and again, but this is not the Waterfall model.

Both the Waterfall model and the Safety Lifecycle model were task-precedence diagrams without loops. Now, there is a loop in the Safety Lifecycle model. And the meaning of the arrows is no longer clear, for it seems that on looping back to Stage 3 from Stage 9.4, one may skip Stage 9.3 the second time through, for (let us assume) one has completed this stage successfully. So now it seems as if the tasks in the flow down the Safety Lifecycle may no longer be compulsory: one may skip certain tasks if one is, say, on the second time along a path which contains a loop. But then how do we determine which tasks we may skip on a path with a loop, and when? There is no guidance.

An alternative would be expressly to insert the additional necessary hazard and risk analyses in the Safety Lifecycle model at the points at which they must be performed. This would involve inserting a hazard and risk analysis step

- after Stage 9.1, before Stage 9.3,
- after Stage 9.3, before Stage 9.4,
- after Stage 9.4, before the confluence of the arrows from Stages 9.2 and 9.5,

and, presenting some difficulties,

- within the "inside" of Stage 9.3 itself.

The hazard and risk analyses within Stage 9.3 could be addressed by appending a note to the effect that between all recognised software development lifecycle stages within Stage 9.3, whatever the software development lifecycle model used, one may expect a hazard and risk analysis to be performed.

Both of these steps involve modification of the Safety Lifecycle diagram. We may conclude that the current diagram in the standard is an inaccurate portrayal of what must be performed.

Functional Safety

The IEC 61508 standard concerns itself with *functional safety*. This phrase refers to aspects of safety concerning the function of the system. There will typically be safety aspects of a system which do not concern the function of the system directly. These aspects are usually noted by the term *non-functional safety*.

Consider for example a level crossing (called a grade crossing in the U.S.A.) at which a road intersects a railway line. On moderately well-used roads, there will typically be movable barriers which, when raised, allow road traffic across the rails, and, when lowered, act as a barrier to passage of road vehicles. The barriers will be raised when no train is approaching, and lowered to halt road traffic when a train is approaching the crossing. There will also usually be an assortment of visual and aural warnings used when a train approaches. A typical level crossing, a grade crossing in Montana, U.S.A., is shown below. The view is taken from along the road. The barriers are shown raised. The rail lines, and thus trains, cross left-right between the barriers. When a train approaches, the barriers are lowered, and the red warning lights and the bells operate.

A level crossing in Montana, U.S.A., viewed from along the road.
(From Wikipedia, "Level Crossing", http://en.wikipedia.org/wiki/Level_crossing)

Functional safety aspects of this system would concern, say, the operation of the barriers and lights. Is it possible for the barriers to remain raised when a train approaches? Do the lights and bells always operate when a train approaches? How do operators know when a light or bell or barrier is no longer functional? Are the barriers always visible to road traffic when lowered? IEC 61508 concerns these aspects of a system.

Non-functional safety aspects might concern, say, the toxicity of materials used in the construction. Say, whether the paint on the barriers is lead-based and whether this poses any danger to humans and animals in the vicinity. Whether the lubrication oil or grease is toxic, and, if so, whether it is guaranteed to be contained or whether some can leak out. In other words, safety aspects of the system which are secondary to its intended function. Of course, oil, grease and paint are essential to the correct functioning of the system, but the function of the system concerns solely when barriers and warnings operate, and when they do not, and when they should (and maybe when they should not). IEC 61508 is not concerned with the non-functional aspects such as toxicity of system components.

Assuring Functional Safety and Safety Functions

If it is found during the hazard and risk analysis of the system functions that a particular aspect is too risky (we discuss below how this is determined), then the specific risk must be mitigated or eliminated.

One way to do this is to redesign the system such that the specific risky aspect is no longer present. For example, at a road-rail crossing, if it cannot be determined to a satisfactory degree that barriers will always come down and lights flash and bells sound when a train approaches a level crossing, maybe one would build a road bridge over the railroad instead.

Another way, emphasised by IEC 61508, is to provide additional functions whose purpose is to intervene to mitigate the identified risk so that it becomes acceptable. Such a function is called a *safety function* in the standard. The definition is somewhat convoluted:

- **Safety function:** Function to be implemented .... which is intended to achieve or maintain a safe state for the EUC, in respect of a specific hazardous event

Here, the mitigation of the risk is phrased in terms of a "safe state" for the Equipment under Control. But safety is not necessarily solely concerned with states, divided into "safe" and "non-safe", but also concerned with behavior. Although, indeed in theoretical computer science a "safety property" of a computational system is a partition of the states of the system into two (which may be called "safe" and "unsafe").

Consider, for example, a car collision in which airbags are deployed and save the car occupants from injury. Call this Case 1. Consider as well the exact same collision, with the same car and people at the same place, and the same dynamics, but this time without airbag deployment, and the occupants are seriously injured. Call this Case 2. Now, it would be reasonable to think that the airbags execute a safety function. They certainly have a function: to deploy and cushion in case of a sudden deceleration. This function has also, obviously, a lot to do with avoiding harm, since the human body cannot tolerate decelerations of the sort that occur in substantial car collisions without airbags.

But the airbag deployment does not return the EUC to a safe state. Indeed, the EUC remains just as thoroughly totalled in Case 1 as is does in Case 2. So it looks as if airbag deployment is not a safety function in the sense of IEC 61508.

Let us look again at my above explanation of a safety function as one whose purpose is to intervene to mitigate the identified risk so that it becomes acceptable. We can certainly say that Case 2 is a hazardous event, because it was a hazardous situation which resulted in harm: people were injured. This hazardous event has a certain risk associated with it, namely its likelihood of occurrence combined with its severity. The airbag deployment serves to reduce the severity of this hazardous event considerably, indeed in Case 1 to the point at which it is barely still a hazardous event according to IEC 61508 (let us suppose the occupants are badly shaken, so this still counts as harm). Having functioning airbags thus serves not to alter the likelihood of any hazardous events that are collisions, but it serves to alter the severity of each of those events. Thus it is a risk reduction measure (one factor in the risk calculation remains the same; the other is reduced). It mitigates the risk so that it becomes acceptable, and does so through affecting the second component of the risk calculation rather than the first.

So as a function to reduce risk to an acceptable level, the airbag deployment works. It may make sense to consider it and its brethren an “honorary safety function” in an IEC 61508-conforming development.

Risk and Risk Reduction

The entire assessment of safety in IEC 61508 occurs through risk assessment and risk reduction. In the hazard and risk analysis, hazardous events are identified and the necessary risk reduction for these events determined. I discuss here how this is envisaged to happen.

Apart from the definition of “risk” which we have discussed, IEC 61508 says that “[r]isk is a measure of the probability and consequence of a specified hazardous event occurring” (Part 5, Annex A: Risk and safety integrity - general concepts, Paragraph A5. The definitions explicitly refer to Part 5, Annex A for “discussion”, so we may assume this is intended to be definitive). This suggests that the overall risk of using a system, the de-Moivre-type risk, is not a concept to which IEC 61508 explicitly gives much credence. Consider the definition of “EUC risk”:

- **EUC risk:** risk arising from the EUC or its interaction with the EUC control system

If we are to put together EUC risk with its explanation as a measure of the “probability and consequence of a specific hazardous event”, we would say that EUC risk is a measure of some features of a specific hazardous event. Assuming there will be many sorts of hazardous events, this does suggest that we are being led to consider each of them individually, and not aggregate as in a de-Moivre-type risk assessment. Indeed, the requirement in Part 1, Paragraph 7.4.2.7 is that

- **7.4.2.7** The EUC risk shall be evaluated, or estimated, for each determined hazardous event

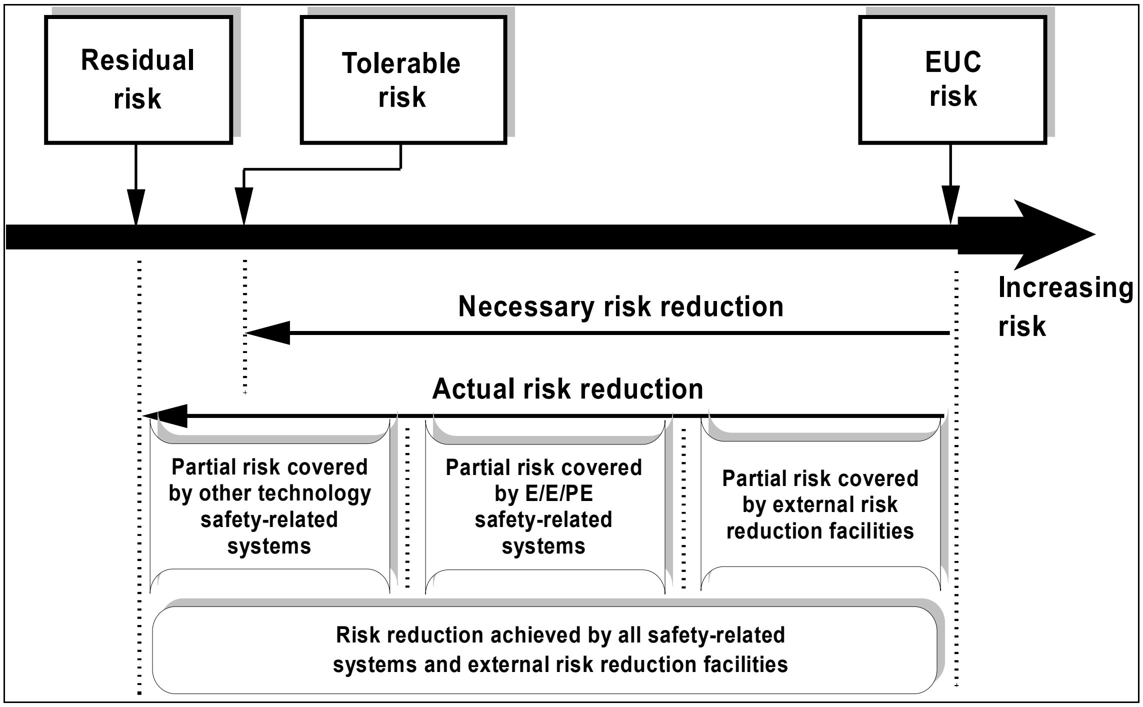
Given that the EUC risk has been determined for a specific hazardous event, there is now a requirement that

- **7.5.2.2** The necessary risk reduction shall be determined for each determined hazardous event. The necessary risk reduction may be determined in a quantitative and/or qualitative manner.

This refers to the notion of “necessary risk reduction”:

- **Necessary risk reduction:** risk reduction to be achieved by the E/E/PE safety-related systems, other technology safety-related systems and external risk-reduction facilities in order to ensure that the tolerable risk is not exceeded

This process of determining risk reduction is summarised in the following diagram:


Residual risk
Tolerable risk
EUC risk
Increasing risk
Necessary risk reduction
Actual risk reduction
Partial risk covered by other technology safety-related systems
Partial risk covered by E/E/PE safety-related systems
Partial risk covered by external risk reduction facilities
Risk reduction achieved by all safety-related systems and external risk reduction facilities

Let us consider a specific hazardous event, E, and suppose one has determined the EUC risk of E and the tolerable risk of E (in other words, what risk "society accepts" of E). Suppose further than the EUC risk of E is higher than the tolerable risk of E. Then one must take steps to ensure that the risk of E in the overall system S is reduced to at most the tolerable risk of E. The means envisaged by IEC 61508 for the risk reduction in the E/E/PE part is the introduction of functions which specifically reduce the risk of E, so that the risk of E in the operation of the system S', where

- S' = EUC enhanced with the introduced functions

is at or below the tolerable risk of E. The risk of E in the operation of S' is called

- **Residual risk:** risk remaining after protective measures have been taken

The difference between the EUC risk and the residual risk is denoted in the diagram as the "actual risk reduction". This phrase has no formal definition, probably because it is clear what it means.

This analysis and prophylaxis is supposed to take place for each specific hazardous event.

Determination of Risk Reduction: Balancing the Options

Suppose one has performed a hazard analysis and identified a set of hazardous events and their likelihoods and severities, as required in 7.4.2.

One of the ways in which one can deal with an identified hazard is to redesign the system to eliminate it. Indeed, IEC 61508 requires

- **7.4.2.2** Consideration should be given to elimination of the hazards

But suppose eliminating one hazard increases the risk of another hazardous event not associated with that hazard?

For example, a couple of decades ago, experiments were performed on an anti-misting kerosene (AMK) to be used in jet aircraft. The idea of the kerosene was to inhibit ignition of the fuel in the case of an aircraft accident. Indeed, the FAA believed it had identified 32 accidents between 1960 and 1984 in which AMK could have saved lives[1]. A test was performed using a remote-piloted Boeing B720, which was landed in the desert into a set of large steel "knives", placed vertically in the ground, whose purpose was to slice into the tanks and release fuel into the atmosphere, and see what happened. The fuel ignited, but burned at a lower temperature than otherwise expected, and less of it burned, thus giving potential passengers more time to evacuate away from the aircraft. Nevertheless, the television coverage showed a fireball and developments of AMK were effectively discontinued. Suppose indeed that development of AMK had been continued. The fuel had different physical characteristics from the usual jet fuels, and new means of delivery, both from fuel tankers to aircraft, and from aircraft fuel tanks to engines, would have been required. So, suppose one had decided to eliminate the hazard of immediate conflagration in an aircraft accident by using AMK. How does one balance this against the increased risk of engine problems during flight, which would have been engendered by the change of delivery technology required?

Consider another example. There has been a general trend over the last thirty years in new commercial aircraft design to replace mechanical control systems with electrical/electronic control systems. One of the reasons for this has been to reduce overall aircraft weight and to increase the ease of building the aircraft. However, it has also been argued that electrical/digital-technology control systems are actually more reliable than their forbears, inter alia because of the added ease of implementing fault-tolerance in such systems. However, putting more electrical wires into airplanes, bound together in wire bundles, increases the chances of insulation faults and arcing within and between electrical wires, so called "arc faults", a phenomenon given increased prominence after the accident to TWA 800 in 1996 and to Swissair 111 in 1998, and the ensuing investigations into wiring quality on commercial aircraft. How may one balance the potentially reduced risk of functional faults with electrical control systems against the increased risk of arc faults and ensuing aircraft fires due to the increase in required electrical wiring?

This example is particularly poignant because it involves exchanging a risk in function (operational reliability of a control system) with a non-functional risk (a wiring fire is a non-functional safety hazard, under the definition of functional/non-functional we have seen above). So in this example the second risk, that of a wiring fire, is "invisible" to IEC 61508 and therefore so is any trade-off of risk. IEC 61508 would suggest to increase control-system reliability through moving to electrics and would not necessarily recognise the increased wiring-fire hazard which it might engender.

IEC 61508 gives no guidance on how one may proceed in such cases, nor indeed is there any explicit recognition in the standard that such circumstances could arise.

There are a number of possible ways in which the convenors of IEC 61508 could reply to the above queries. The most trivial would be to say that IEC 61508 does not cover commercial aviation; that

[1] For a discussion of the experiments around AMK, see Chapter 3 of Harry Collins and Trevor Pinch, The Golem at Large: what you should know about technology, Cambridge University Press, 1998.

would just be avoiding the question, but this author has nevertheless heard such kinds of reply. A second reply would be to say that the standard allows recognition of the situation, but neither explicitly recognises the situation nor offers guidance because there is no generally agreed way to proceed. It may be the case that there is no generally-agreed way to proceed, but this is so with lots of other issues, such as when and how to apply so-called formal methods in development and analysis, on which the standard *does* offer guidance: at least an explicit recognition of the situation seems appropriate.

A third answer would be that the standard in fact *does* offer guidance, but implicitly, and that the guidance it does offer is not necessarily appropriate! As follows.

Let us suppose we have two hazards, H1 and H2. Say, H1 is the chance of immediate, deadly conflagration of jet fuel in the case of a tank rupture; and H2 is all engines cutting out in flight. IEC 61508 says to give consideration to eliminating the hazard. Suppose we do so, and eliminate hazard H1 (say, by introducing AMK into daily commercial flight operations), and thereby increase the risk associated with H2 (engines failing during cruise flight because of interactions because the fuel delivery systems and the physical properties of the fuel argue with each other). The IEC 61508 tells us we must look at hazardous events arising from H2, whose risk has been increased (through increasing the probability of occurrence of H2), and reduce the risk of those hazardous events to a tolerable level (to the "tolerable risk" associated with the event). So it seems as if IEC 61508 might be telling us to eliminate hazards where possible, and then work harder to reduce the increased risk of others down to their defined "tolerable level".

But what about considering the joint risk of H1-events and H2-events? Although one has eliminated the risk of H1-events, maybe the "tolerable level" associated with H2 events is such that the risk of (H2-events with AMK) is higher than the joint risk of (H1-events without AMK) together with (H2-events without AMK). A risk reduction approach associated with a de-Moivre-type risk would say that the joint risk is what is most important and advise to reduce the joint risk as far as possible. This seems then to be an approach which is contra-indicated by IEC 61508 as written.

The possibility of reading IEC 61508 contrary to established reasonable approaches, such as the de-Moivre type, suggests that explicit guidance in IEC 61508 is necessary. And it is not there at the moment.

The core of this issue lies with the focus in IEC 61508 on risks of individual "hazardous events", rather than on overall risk, as considered by a de-Moivre-type calculation.

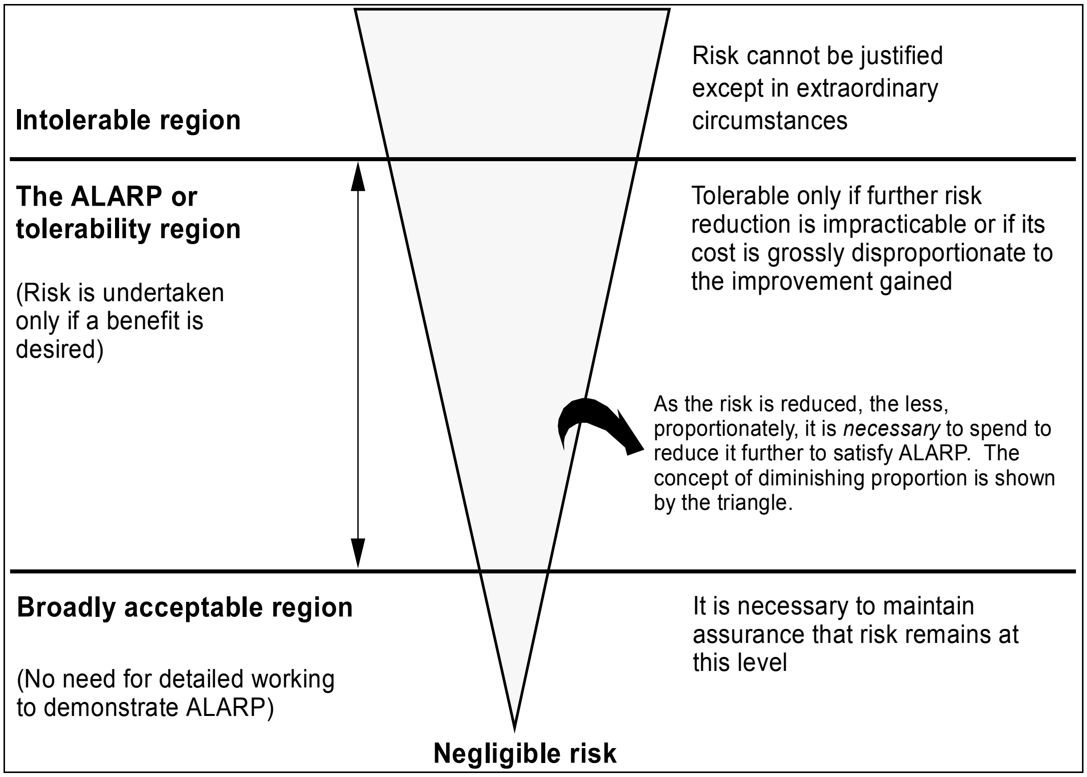
Risk Reduction and ALARP

IEC 61508 considers ALARP explicitly, in Part 5 Annex B, *ALARP and tolerable risk concepts*.

The first observation to make is that, while IEC 61508 concerns itself with reducing risk of infividual events, the ALARP principle encompasses overall risk. In the quandary over how jointly to handle H1 and H2 above, ALARP would say to reduce the joint risk (if it is "reasonably practicable" so to do) rather than eliminated H1 and incurring thereby an increased risk of H2-events above that of the previous joint risk.

There have been various attempts to turn ALARP into a principle of engineering, and IEC 61508 Part 5 Annex B is just one of them. It seems wise, therefore, to reiterate that ALARP is a principle of English law, not an engineering principle. It may be that engineering principles can be devised that, say, would have avoided the cases in which English law decided that ALARP had been violated, but there is no a priori guarantee that these principles exactly encompass all that case law has and will decide about ALARP. It is with this caveat that we may consider how ALARP can guide engineers.

The guidance given in IEC 61508 Part 5 Annex B uses a diagram:


Intolerable region
Risk cannot be justified except in extraordinary circumstances
The ALARP or tolerability region
(Risk is undertaken only if a benefit is desired)
Tolerable only if further risk reduction is impracticable or if its cost is grossly disproportionate to the improvement gained
As the risk is reduced, the less, proportionately, it is *necessary* to spend to reduce it further to satisfy ALARP. The concept of diminishing proportion is shown by the triangle.
Broadly acceptable region
(No need for detailed working to demonstrate ALARP)
It is necessary to maintain assurance that risk remains at this level
Negligible risk

This diagram, on the vertical axis, suggests the following

1. There are cases in which the risk is unacceptable, no matter what you do. The engineering advice is: don't go there except in “extraordinary circumstances”.
2. There are cases in which the risk is (socially) acceptable anyway. As an engineer, if you are in this region then you don't need to perform “detailed working” to demonstrate adherence to ALARP.
3. There is a region in which ALARP affects your work as an engineer. You have no need to reduce risk further than you have if:
   - either it is impracticable to do so, or
   - the cost of doing so is “grossly disproportionate” to the “improvement gained”.

Let us consider first Case 2. Suppose the risk is acceptable. But suppose for a certain amount of available resources you could have reduced the risk further. The legal principle of ALARP seems to require that you do so. However, the IEC 61508 guidance says “no need to perform detailed working to demonstrate” this. This seems to be, at best, questionable advice. A lawyer might well advise “whatever you think of the magnitude of the risk, make sure your documents prove you've done everything reasonable practicable to reduce it”. This suggests that “detailed working” might indeed be helpful, in contrast to Part 5 Annex B's suggestion that there is “no need”. All kinds of random assertions are made in court trials: my view is that it is best to be well prepared.

Case 3 is an interpretation of case law as it presently is in England. If it is impracticable to reduce risk further (and you can demonstrate to the satisfaction of the court that it is impracticable!) then indeed you have no need to reduce risk further. This advice relies explicitly upon the finding of Lord Asquith of Bishopstone in Edwards vs. National Coal Board 1949, in which he said

> *“Reasonably practicable is a narrower term than ‘Physically possible’ and implies that a computation must be made […] in which the quantum of risk is placed in one scale and the sacrifice involved in the measures necessary for averting the risk (whether in time, trouble or money) is placed in the other and that, if it be shown that there is a great disproportion between them – the risk being insignificant in relation to the sacrifice – the person upon whom the obligation is imposed discharges the onus which is upon him.”*

This may be taken broadly to substantiate the advice given in the diagram above. But note there is also further advice in the diagram, that “*as the risk is reduced, the less, proportionately, it is necessary to spend to reduce it further to satisfy ALARP*”. I do not know of a legal basis for this advice; neither does it follow from the finding of Lord Asquith.

The U.K. Health and Safety Executive explains its role thus: “*HSE's job is to protect people against risks to health or safety arising out of work activities.”* It was set up as a regulatory body by the Health and Safety at Work Act of 1974. It acts as the de facto regulator for complex safety-critical systems that are not otherwise regulated (for example, commercial aviation is otherwise regulated). HSE brings legal cases against companies deemed to have violated HSW. In deciding whether to bring such cases, one factor HSE considers is, of course, whether a company has fulfilled its duty to reduce risk ALARP, and HSE often uses IEC 61508 as a reference standard to test this duty[2]. Thus is a strong connection made between ALARP and IEC 61508 in the U.K. through regulatory behavior.

Subsystem Types

There are three types of system and subsystem to which IEC 61508 continuously refers:

- Equipment under control (EUC).

- EUC control system (EUCCS).

- Safety-related system (SRS). A system which “*both implements the required safety functions needed to achieve or maintain a safe state for the EUC and is intended to achieve, on its own or*

[2] Thanks to Mark Bowell, HSE, for suggesting this way of explaining HSE's deliberations (personal communication, 11. August 2008). Mark has explained HSE's policies more thoroughly in notes to a mailing-list on safety-critical systems, for example http://www.cs.york.ac.uk/hise/safety-critical-archive/2004/0473.html . Mark suggests that “the most important part of this [note] from HSE's perspective is the reproduction of the statement from the HSE board, at the end of the post.” Readers may also want to refer to HSE's enforcement policy statement at http://www.hse.gov.uk/pubns/hsc15.pdf .

with other E/E/PE safety-related systems, other technology safety-related systems or external risk reduction facilities, the necessary safety integrity for the required safety functions." So an SRS implements the safety functions and is which includes amongst its purposes not only implementing safety functions but also achieving the required safety integrity.

There is just one of EUC and EUCCS. There may be many SRSs. An SRS is required to mitigate any hazardous event for which the tolerable risk is lower than the EUC risk, and there may be many such hazardous events.

Besides these three subsystem types, there is the fourth type, of

- Subsystems which are none of the above

Safety Integrity and Safety Integrity Levels (SILs)

Safety integrity is a key concept in IEC 61508 and is defined as the reliability of an SRS:

- **Safety integrity:** probability of a safety-related system satisfactorily performing the required safety functions under all the state conditions within a stated period of time

E/E/PE SRSs are assigned a safety integrity level:

- **safety integrity level (SIL):** discrete level (one out of a possible four) for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, ...

The four safety integrity levels for so-called high demand or continuous-mode systems are logarithmically related, according to the following table:

Safety integrity level	**High demand or continuous mode of operation** (Probability of a dangerous failure per hour)
4	≥ 10^{-9} to < 10^{-8}
3	≥ 10^{-8} to < 10^{-7}
2	≥ 10^{-7} to < 10^{-6}
1	≥ 10^{-6} to < 10^{-5}
NOTE See notes 3 to 9 below for details on interpreting this table.	

The safety integrity refers to not to failure but to *dangerous failure*:

- **dangerous failure:** [a] failure which has the potential to put the safety-related system in a hazardous or fail-to-function state

We recall that there is no definition of what a "hazardous state" might be, but it might well be a SRS state as a consequence of which hazardous situation ensues. There is likewise no definition of what a "fail-to-function state" is, and here it is harder to guess. Is a fail-to-function state one in which the SRS engages in no dynamic behavior at all, or can it be a state in which the behavior of the SRS is different from that specified or required? If it is a state in which the SRS engages in no behavior (it is stopped), then it is easy to see how hazards arise. The SRS is required to implement one or more safety functions, and a safety function is specified because it has been determined that the EUC risk of a specific hazardous event is not tolerable. The safety function is there to reduce this risk to a tolerable risk. If the safety function is no longer active, then the EUC risk of this specific hazardous event is not tolerable, so it is easy to understand how a failure which might lead to a SRS failure to function (under this interpretation) could be termed "dangerous".

I emphasise that the notion of SIL applies to the E/E/PE parts of a system. There has been considerable discussion amongst safety-critical digital system engineers about the adequacy of this notion as defined in IEC 61508, and many have considerable reservations about it. There are other notions of SIL used in other standards which do not necessarily suffer from the same disadvantages.

First, a SIL is an attempt to designate the criticality of an SRS implementing a safety function. It is an attempt to do so which retains a certain amount of simplicity, namely there are just a few "boxes" into which any SRS function can be put. The simplicity is advantageous in that the categories are necessarily broad, so that it is not necessary to try to estimate the exact reliability with which a safety function is performed, but just to fit it into a general region. To have few categories while covering all the possibilities, one is drawn most obviously to some kind of logarithmic scale, which is how the categories are indeed defined in the table.

However, as Martyn Thomas has pointed out, the categories are incomplete. What if you need a safety function F, but it only needs to be reliable to the point of one dangerous failure in ten thousand operational hours? That is a critical figure, and one that is not necessarily easy to achieve, but yet it accrues to no SIL. IEC 61508 proffers guidance on what techniques are appropriate for developing an SRS with a specified SIL, but since the requirement for F corresponds to no SIL, there is correspondingly no guidance. But there is no reason to think that the safety function F is any less critical to safety than any other safety function.

There may also appear to be an incompleteness at the other end. No failure rate lower than one dangerous failure in 10^9 hours of operation occurs in the SIL table. One reason for this is explained in Note 6 accompanying the table. The standard sets a lower limit of one dangerous failure in 10^9 hours of operation on the any dangerous-failure measure that can be claimed. So it does not allow anyone to claim lower than this in any circumstance. On the other hand, to achieve SIL 4, the table suggests one has to show not only that a failure rate is lower than one dangerous failure in 10^8 hours of operation, but also that *more than or equal to one dangerous failure must occur every 10^9 hours of operation*, which seems perverse (as well as likely indemonstrable). It would have been preferable simply to specify a probability of dangerous failure per hour of "< *10^(-8)*". We may hope this anomaly will be corrected in future revisions of the standard.

A fundamental disquiet with the notion of SIL used in the standard is the association of a SIL with a set of recommended development techniques, for example, whether the use of formal methods is or is not recommended. So, for example, the use of *formal methods such as CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z* is "*recommended*", but "*only exceptionally, for some very basic components only*" for SIL 3 (Part 6, Table E.14). One can query the wisdom of this sort of advice, as follows.

A SIL of 3 requires a system attain a very, very low dangerous failure rate, between one in ten million and one in one hundred million hours of operation. Now, typical safety-critical code quality is about one error per thousand lines of source code (per KLOC: "Lines of code" = LOC, thousand LOC =

KLOC) (John McDermid, personal communication). Some organisations are able to attain measured quality of one error in 25 KLOC by using highly-integrated formal methods while writing the programs (for example, the SPARK Ada system of the U.K. company Praxis High-Integrity Systems). Others are able to attain one error per 4 KLOC on smallish systems, say a few tens of KLOC (see for example Les Hatton, Safer C, Addison-Wesley 1989). As is well-known, the consequences of these errors in software can hardly be functionally bounded in advance. It is usually not possible to say with any degree of confidence how the system will ultimately behave if a certain type of error occurs. It follows that a worst-case assumption is that each error in SRS software may result in a dangerous failure, and it is rare to see how to do better than this worst-case assumption. So there is a wide gap between measured software quality, even using integrated formal methods throughout development, and a requirement for dangerous failure such as contained in SIL 3. Use of formal methods such as by Praxis seem to improve the quality of delivered software by one to two orders of magnitude, but there is no known evidence that one can achieve SIL 3 reliability *without using rigorous development techniques*, as the standard suggests.

Indeed, even if there were such evidence, what could be a reason for not wishing software to be as reliable as possible, even exceeding its SIL requirement?

One suspects that behind such advice lies the old canard about formal methods consume resources in great quantity when compared with the gain in quality that they enable. Against, one may observe that certain companies use formal methods in order to produce software of higher quality with lower resource consumption. Since this can be done, it sets the state of the art and there is no reason not to recommend, indeed require, state-of-the-art quality-attainment methods in *every* critical software development, from SIL 0 (if it existed) through SIL 4.

In question here are two aspects of the association in IEC 61508 of development methods (as “not recommended”, “recommended” or “highly recommended”) with software quality as represented in the SILs.

The first aspect is that, although one may reasonably expect particularly careful development processes to result in software of higher quality, no association has ever been shown between the use of a specific development technique and a specific quality level of the resulting software, let alone for a range of development techniques and the entire SIL quality-specification range.

The second aspect is that any association of specific methods with reliability ranges that are powers of ten must be somewhat arbitrary. Even if such an association were to be shown between, say, requirements analysis and specification in Z and rates of dangerous failure in the range of less than 10^(-7) per operational hour, what reason would we have to expect that such a boundary would lie at 10^(-7) rather than, say, 12^(-7)?

More obviously reasonable advice on association between development techniques and SILs would surely be that one be free to use whatever development techniques one thinks suitable, but that the *results*, in terms of the quality of delivered software, must be demonstrated to a high degree of confidence based on objectively good evidence.

However, this runs up against the difficulty of what might constitute good evidence. In 1993, Littlewood and Strigini pointed out, for a Bayesian approach to evidence, and Butler and Finelli for a frequentist approach, that assuring any failure rate (dangerous failure or not) of below 10^(-5) per hour through statistical testing of a software system was infeasible. Indeed, Littlewood and Strigini observed that if you needed the result of your testing to confirm a failure rate of 10^(-6) per hour or below, you had to go into testing with a very good reason to claim (a prior probability in Bayesian-theory terms) that you had achieved that rate already. And if you have that good reason in any case, why do all that testing?

There are serious issues here of evidence and achievement that cannot satisfactorily be summarised in a few paragraphs, but constitute rather a research program. It seems likely, though, that any anchoring of evidence to failure rates will have to take place at much higher failure rates than those envisaged in the SIL table in IEC 61508. It seems certain that that table will have to be significantly modified. For these reasons and others, many thoughtful engineers consider the requirements concerning SILs in IEC 61508 to be irrecoverably flawed.

Lowest Recognised Risk Reduction

We have noted that the standard sets a lower limit of one dangerous failure in 10^9 hours of operation on the any dangerous-failure measure that can be claimed. There are at least two areas in which this lower limite may or must be exceeded.

1. In commercial airplane certification, critical subsystems must be free of any single failure that causes a catastrophic event (loss of the airplane, loss of life) to a level of *at most* one in 10^9 operational hours.
2. In the automobile industry, it is well possible that some critical electronic components attain of the order of 10^(10) lifetime operational hours[3]. Suppose they could be analysed and guaranteed down to one dangerous failure 10^9 operational hours, the lowest level recognised by IEC 61508.Then, as far as one could tell, there would still be of the order of 10 critical events during the system life.

One may well query whether it is wise to have a standard that does not allow us even to *express* a measure of one dangerous failure, or more generally discriminate this case from that of ten dangerous failures, in the operational life of a particular system.

Relationship between SILs and ALARP

Felix Redmill has elucidated the relationship between SILs and the ALARP principle for U.K. engineers (Proceedings of the 8th Safety-Critical Systems Symposium, Southampton, U.K., Spinger-Verlag London, 2000).

Redmill pointed out that a SIL is an a priori requirement, which is defined and allocated in the Safety-

[3] A car is said to travel on average 300 hours per year, and popular model cars sell of the order of a million examples per year. In just over three years, one may expect 10^9 lifetime operational hours on just this one model year. But a device may be used on cars over many model years, and critical components especially on popular cars are also supplied to competitors. So if a component is used on three popular models over three model years, we are in the realm of 10^(10) operational hours.

Requirements-Analysis task during development. However, ALARP is a dynamic requirement that is assigned and handled in the task of Design, when designing and building the system is underway. He concludes that in some particular case the ALARP principle could well require a reduction in risk beyond that required by the SIL for a function.

This observation is astute, but to date no examples of this phenomenon have appeared in the literature. One may also further observe that, since the HSE consider ALARP to be satisfied if IEC 61508 has been followed during system development, that no further evidence would be required by the regulator that ALARP has been followed than by exhibiting the derived SIL and the evidence for its attainment.

Appropriate and Inappropriate Applications of the IEC 61508 Approach

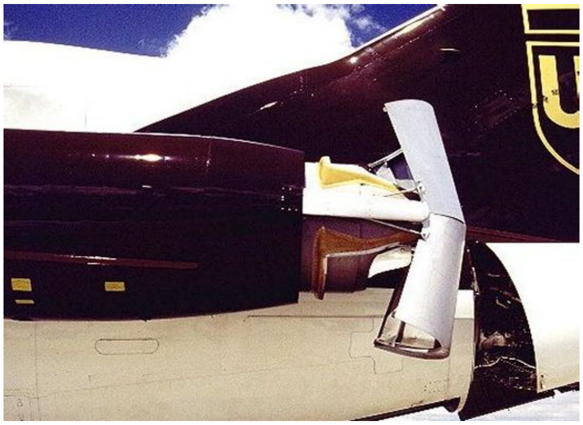
With some understanding of the basic concepts behind IEC 61508, we may now look at some possible applications and see how and whether the standard fits those applications conceptually.

Safety Functions

We have seen that achieving risk reduction through safety functions may not always be the most appropriate way to reduce risk.

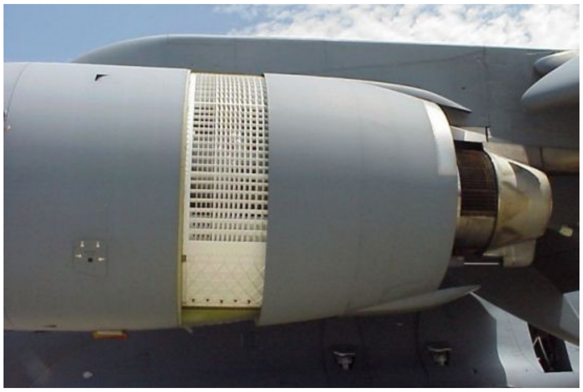
An appropriate application

An example of an appropriate application of a safety function is the interlock on Boeing B767 aircraft (as well as many others) which prevents in-flight deployment of reverse thrust. The normal thrust on a jet aircraft comes from the air and other gases expelled out of the rear of the engine. Reverse thrust is a mechanism to help deceleration on the runway during landing. A thrust reverse mechanism can be one of two types, “clamshell” or “cascade”



A deployed clamshell thrust reverser

Two parts of the cowling rotate into the thrust stream and mechanically deflect it upwards/downwards and, most importantly, somewhat forwards. This forwards-directed component of thrust helps brake the aircraft on the runway during landing. A thrust reverse mechanism of the "cascade" type is not quite as visible, since the thrust stream is reversed this time inside the cowling and exits midway. A cascade reverser works on the thrust stream essentially the same way as a cascade reverser.



A deployed cascade reverser

On many if not most modern jet aircraft, thrust reverser deployment in flight almost inevitably leads to loss of control of the aircraft, which is defined as a "catastrophic" event in commercial aviation certification terms. So such an event would count as a hazard, and it can be eliminated through inhibitory mechanisms. On the Boeing B767 aircraft, there is a hydromechanical mechanism which physically prevents the thrust reverse mechanism from operating when the landing gear is not deployed and there is not weight on the main landing gear (known as a "weight on wheels" or WoW criterion). WoW may be determined in a number of ways, most of which involve sensing main landing gear strut movement as the aircraft weight compresses the shock absorbers.

In May of 1991, a Lauda Air Boeing B767 crashed over Thailand. It was determined that thrust reverse had been electrically commanded (the crew spoke of the deployment warning light illuminating on the cockpit voice recorder) and the subsequent radar track showed a loss-of-control descent consistent with thrust reverse having actually been deployed on the affected engine. However, there was no failure mode known of the interlock mechanism, the safety function as IEC 61508 would have called it. Subsequent tests discovered such a failure mode. The rubber-compound seals on the interlock hydraulics became brittle with time, and could partly disintegrate, leaving hard rubber particles in the hydraulic fluid. These particles could become lodged in a valve in the interlock mechanism, preventing it from closing fully when it needed to, and thus inhibiting the interlock function. Since at least one failure mode of the interlock mechanism had been demonstrated, it was thereby known that the interlock could fail and thus that in-flight thrust reverse, while unlikely, was possible. Deployment of the left engine thrust reverser in flight, leading to loss of control, was designated as the probable cause by the accident investigators.

An inappropriate application

The ground spoilers and thrust reversers on the Airbus A320 aircraft are used, along with wheel brakes, to brake the aircraft on the runway. We have discussed thrust reverse, which is actually a minor braking effect (but which becomes significant when the runway is slippery, say when very wet or icy). Ground spoilers dump the lift on the wings, putting the aircraft weight on the wheels and enabling effective wheel braking. Ground spoilers are deployed and thrust reverse enabled on landing through digital logic involving the WoW criterion.

Ground spoilers deployed on an Airbus A321
(from http://en.wikipedia.org/wiki/Spoiler_(aeronautics))

In September 1993, an Airbus A320 landed fast on a very wet runway in Warsaw just after passage of a thunderstorm with accompanying wind shear (which the previously landing aircraft had experienced a minute or so earlier). Upon touchdown, ground spoilers should have deployed automatically, and reverse thrust was also commanded. However, neither functioned for some 9 seconds (and the wheel brakes not for 13 seconds). The aircraft ran off the end of the runway, hit an earth bank, and burned. Two people lost their lives, but all others, passengers and crew, escaped without injury.

The inhibition of ground spoilers and reverse thrust during flight has the same justification on the A320 as on the Boeing B767. As we have seen, reverse-thrust inhibition proceeds on the B767 through an explicit safety function, namely an interlock mechanism. On the Airbus A320 family, this inhibition is accomplished through digital logical conditions, which signals also have other functions and effects. It seems inappropriate to speak of implementing a “safety function” here. Rather, it seems preferable to speak of a safety requirement being satisfied through the design of the digital logic.

Risk Reduction

We have seen that risk reduction is the main mechanism through which IEC 61508 enhances safety.

An appropriate application

Consider an anti-lock braking system, an ABS, on a road car. In terms of IEC 61508 concepts, the system components are as follows.

- the EUC is the brakes
- the EUCCS is the brake activation mechanism, from pedal to brake pads
- the SRS is the wheel-rotation sensors and the responsive brake-release-and-reapply mechanism

The EUC risk we can take as known, similarly the tolerable risk, so one can calculate the required risk reduction and transform this requirement into a SIL according to the SIL table. Then one demonstrates that the ABS fulfils the SIL.

Note that IEC 61508 is not applied in this way. Designers wish to allow the car to be driven also when the ABS does not function. So ABS is not designated as an SRS (which must always be active) but rather as a functional enhancement which is not formally safety-related. So this example is conceptual. I chose it because it is rather clear.

An inappropriate application

This section owes a considerable debt to the discussion in Derek Fowler's paper *Application of IEC 61508 to Air Traffic Management and Similar Complex Critical Systems*, published in *Lessons in System Safety*, Proceedings of the Eighth Safety Critical Systems Symposium, ed. Felix Redmill and Tom Anderson, Springer-Verlag, London 2000.

Consider air traffic control. The purpose of air traffic control is to avoid mid-air collisions between participating aircraft by maintaining a specified amount separation between the aircraft, while enabling each aircraft to continue towards its destination and landing. The IEC 61508 subsystems in air traffic control would be as follows.

- the EUC is the air traffic itself: all participating aircraft
- the EUCCS is Air Traffic Management (ATM), that is, the procedures contained in the laws of flight plus the centralised planning that results in the Flight Plan for each participating aircraft
- the SRS is the dynamic control exercised by the controllers with whom the aircraft crews talk when under way: Air Traffic Control (ATC). The controllers modify the planned flights of the participating aircraft dynamically, to ensure that separation is maintained.

But now we run into some difficulty concerning the necessary estimates of risk according to IEC 61508.

- EUC risk. This would be the probability of collision with ATM, but without ATC. But ATC historically has been present since there has been a significant risk of collision amongst air traffic, certainly under all modern traffic and ATM conditions. So there are no statistics at all by which to judge EUC risk
- Tolerable risk would be given by the Target Level of Safety (TLS), a certain probability of collision. This can be and has been set by ATC specialist organisations when needed

The question then arises of how much risk reduction needs to be achieved by ATC over that provided by ATM? (The “required risk reduction” of IEC 61508.) This question likely cannot be answered; indeed it may not seem appropriate even to ask it. One has no idea of the relative contributions of ATM and ATC to the TLS.

So it seems as if air traffic management and control is not a suitable area in which to apply the risk-reduction concepts of IEC 61508 without considerable alteration in the foreseen approach.

Safety Integrity Levels

An appropriate application

Consider what is called on British trains the Driver's Safety Device, or “dead man's handle”. This is a device which must be continually activated by a train driver, and may be a pedal or a lever, or some other such device

A “dead man's handle” on a suburban train

The device on British trains is a lever with a red knob, which must be kept continually depressed by the driver's hand when the train is running. If it is released, after a second or two emergency brakes are automatically applied. The device is there to ensure that, if the driver becomes incapacitated for any reason (stroke or heart attack, say), the train will be brought immediately to a stop.

In January 2003, near Waterfall, New South Wales, Australia, a train driver suffered a heart attack but the “dead man's brake” did not activate. The train derailed, killing seven people on board including the driver. A brief description of the accident may be found at http://en.wikipedia.org/wiki/Waterfall_train_disaster

The statistics on train drivers being incapacitated are, obviously, known in detail by every rail authority. These authorities also set tolerable risk, or Target Levels of Safety. The required risk reduction can therefore be determined, and from this the SIL of the Driver Safety Device may be read off.

Indeed, the dead man's handle implements an on-demand function which is triggered less than once a year, system-wide. A designation of SIL 4 for an on-demand function requires that it be more reliable than one failure in 10,000 applications (the SIL table for on-demand safety functions is different from that for continuous functions which we discussed above). This would translate, then, to a condition that the safety function provided could fail once every 10,000 years. That is obviously unnecessarily stringent. The device could be happily designed to an on-demand SIL 2 or SIL 3 requirement.

More details on the device may be found at http://en.wikipedia.org/wiki/Dead_man's_switch

An inappropriate application

An inappropriate application is represented somewhat trivially by the thrust-reverser interlock mechanism discussed above. Certification requires a failure probability of *less than* 10^(-9) per operating hour, whereas the most stringent SIL allowed by the IEC 61508 standard is a failure probability of between 10^(-8) and 10^(-9) per hour.

Another inappropriate application is represented by the air traffic control system discussed above. There are many independent ATC subsystems, say of the order of some hundred. These subsystems can be classified, at the highest possible, at SIL 4. However, the entire system, consisting of 100 or more independent subsystems, can then only be shown to satisfy at most SIL 2 (*100 x 10^(-9) = 10^(-7)*). However, typical TLS's for air traffic control are in the range of a probability of dangerous failure per hour of 10^(-8) or lower. Again, I am indebted to Derek Fowler, *op. cit.*, for this observation.

Difficulties with Applying ALARP

To show that IEC 61508 is not alone in its conceptual difficulties, we can consider how ALARP may fare.

An appropriate application

Consider a SIL 1 programmable electronic subsystem, say implemented on a Field Programmable Gate Array (FPGA). Its functional-dangerous-failure probability requirement lies in the region of one every 10,000 operating hours, which is about once per year of continual operation (a year is typically taken by system engineers to be 10,000 hours, more or less. It is actually some 8,760 hours, or 8,784 hours in a leap year). Say one took 4,000 of these FPGAs and ran them continuously with inputs spanning the entire projected operating range for some 3 months, without a failure. Then one has accumulated failure-free experience of the order of some ten million operating hours, and Bayesian statistical calculations from this statistical testing allow one to form a high degree of confidence that the dangerous-failure requirement of one failure every 10,000 hours is fulfilled.

An inappropriate application

Consider now a digital-logic-based flight control system (DFCS) for an aircraft. Since failure of flight controls can lead to loss of control, considered a “catastrophic” event, this system must be required to be reliable to the order of one failure in 10^9 operating hours, a failure probability of 10^(-9) per operating hour. The quantity of 10^9 operating hours has never been exceeded by any fleet of commercial aircraft of a single model, so this says that such a failure may not be expected in the fleet life of the aircraft.

But there is no corresponding requirement on maintenance procedures. In March 2001, an Airbus A320 aircraft departing Frankfurt was found to be cross-controlled, that is, when the pilot flying applied right bank to correct a left wing dip on take-off, he exacerbated the left wing dip and the wing came within a couple of meters of hitting the ground. Luckily, the non-flying pilot noticed what was happening and his control was not crossed, so he was able to recover, by deploying the take-over command button and flying the aircraft himself.

In other words, although risk had been reduced as far as reasonably practicable on the *design* of the DFCS, it had certainly not been reduced as far as practicable on the maintenance, which cross-wired the captain's control for bank, and failed to detect the error during post-maintenance control checks. the ALARP principle applies to design, and not necessarily to maintenance, although of course there are other legal principles, such as *duty of care* which occur under the various concepts of negligence, that would apply should an accident occur through maintenance mistakes and should it be decided to process such an event through the legal system.