Success and Failure: Human as Hero – Human as Hazard

Carl Sandom

iSys Integrity
2 Fairfield Heights
Sherborne, Dorset, DT9 4HH, England

carl@iSys-Integrity.com

"About the only problem with success is that it does not teach you how to deal with failure." Tommy Lasorda

Abstract

Human Factors are often cited as the cause of hazards within safety-related systems (human as hazard); yet system safety cases often contain no mention of them. Conversely, system operators often provide substantial mitigation between hazards and their associated accidents (human as hero); yet this is also often overlooked. If the human factors risks are not considered a system will not achieve the required level of integrity. If the human factors mitigations are not considered the technical components may be over engineered at additional cost to achieve a safety integrity target. This paper explores the positive and negative contributions that humans make to system safety.

This paper deals with problems associated with Human Factors throughout the development of safety-related programmable systems that typically rely on people, procedures and equipment to function safely within a specific operational environment. Typical examples of such systems are found in Air Traffic Control and Railway Control Rooms.

The paper begins by highlighting the problematic relationship between Human Factors and System Safety Engineering before briefly examining several reasons for this difficult relationship. The paper then examines some important safety-related concepts related to risk and barriers before introducing the concept of success and failure cases. Finally, the paper describes a Human Factors process, based upon Critical Task Analysis and Human Error Analysis, used to generate evidence to support human success and failure cases.[·]

Keywords: human factors, barriers, risk, critical task analysis, human error analysis, success, failure.

1 Introduction

Broadly speaking, system safety analyses can take either a top-down or a bottom-up approach. A top-down approach will address safety from a holistic systems perspective and must include the specific context of use of each system when considering credible accidents, accident sequences and their associated hazards. Inevitably, an analysis of the accident sequences will require an evaluation of many different human factors. Conversely, a bottom-up approach will start with an analysis of the errors and failures within a systems boundary and will end at the hazards (or hazardous events) at the system boundary. A bottom-up analysis of any system should include an evaluation of the contributing human errors and failures. Despite this, many system or software safety case do not consider either the internal or external human factors within or external to the system boundary.

A comprehensive and compelling system safety case must be made to enable systems developers to convince regulators that a system is tolerably safe before use. To do this, systems developers must consider both internal and external human factors. Systems are not merely collections of hardware and software components. Systems do not exist in isolation from human interactions and inputs from designers, developers, operators, maintainers and other third parties. Software on its own cannot cause anyone harm as it must be hosted on hardware to function and there must be some form of human input which can be a combination of direct human interaction and indirect inputs from human designers. Despite this, many system or software safety cases do not consider either the either internal or external human factors associated with human interaction or indirect human input.

Some safety standards do advocate the use of a combination of top-down and bottom-up approaches. However, in practice many systems, or subsystem hardware or software components, are analysed using only a bottom-up approach with little or no consideration of either the context of use or human factors in general. For example, neither context nor human factors are considered when applying guidelines from DO-178B, Software Considerations in Airborne Systems and Equipment Certification (1992), to the production and certification of Aerospace software.

In recent years human error has been established as a major cause of accidents in all safety-critical domains. While the statistics vary, in one study Khandpur (2000) found that human error has been attributed to 90% of nuclear facility emergencies; 65% of all airline accidents; 90% of all auto accidents. Despite this, industry often concentrates the majority of safety assurance effort upon technical issues (i.e. hardware, software) while neglecting the extensive human contributions. The human components of safety-critical systems are often not considered safety-critical and are not therefore subject to hazard analysis and risk assessment to the same degree as any other safety-critical system component. The conclusion to be drawn from this is that in many instances, industry produces safety cases that, at best, provide only limited safety assurance as the prevalent errors are related to issues associated with the human factors.

2 Human Factors Challenges

In the previous section it was argued that problems exist with the assurance of systems safety due to the neglect or omission of human factors. This section examines some of the reasons for these problems.

2.1 Complexity

Human Factors is a discipline broadly concerned with the need to match technology with humans operating within a particular environment; this requires appropriate job and task design, suitable physical environments and workspaces and human-machine interfaces based upon ergonomic principles. Systems must demonstrate how their human-computer interfaces can foster the safe, efficient and quick transmission of information between the human and machine, in a form suitable for the task demands.

Human Factors can generally be characterized as dealing with both anthropometrics and cognitive issues related to systems. Anthropometrics are concerned with the physical aspects of system design related to human sizes, colours perception, auditory capabilities etc. and these are relatively straightforward to analyse and evaluate (see Pheasant (1996) for a detailed discussion). In contrast, cognitive human factors are concerned with what is occurring inside the human mind and how this affects human behaviour. Cognitive issues are much more difficult to address than the anthropometrics due to their non-physical, abstract nature and their analysis and evaluation are much more difficult and much less objective. The difficulties associated with the complexity of cognitive human factors issues are one of the major reasons why human factors are often neglected by systems developers.

2.2 Scope

Another reason that human factors are neglected is the problem of scope. Figure 1 shows a representation of a typical system and its boundaries. In this representation the core-system represents the various sub-systems implemented in hardware, software or allocated to human users or operators. The operational level represents the system operators, operational procedures and other organisational factors. Finally, the external environment represents the wider application domain within which the system resides and typically contains people, procedures and equipment.

As discussed in the previous section, systems developers must consider both internal (core and operational) and external (environmental) human factors associated with human interaction and indirect human input. Analyses of human factors issues in safety-related systems consistently reveal a complex set of attributes relating to the people, procedures and equipment interacting at each system level within a complex environment. These attributes are normally very tightly coupled and each of these attributes can interact with the other. Put simply, the scope of a comprehensive human factors analysis is vast and must range from a consideration of relatively intangible issues (e.g. organizational culture) to more tangible issues at the operational interfaces (e.g. human errors).

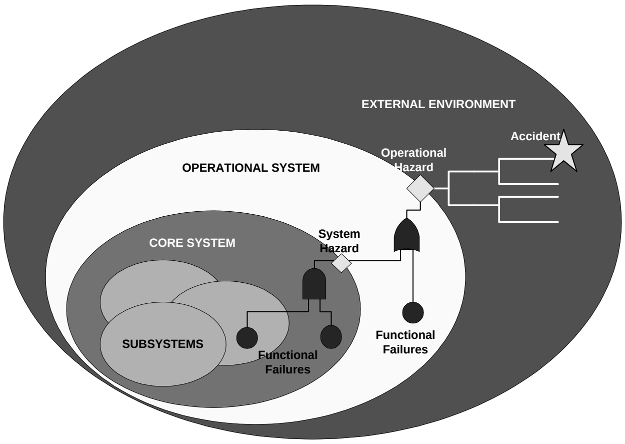

Figure 1: System Boundaries

2.3 Cost-Benefit Justification

Throughout this paper it has been argued that human factors are an important element in the design of effective and safe systems. However, before any organisation can be expected to fund a human factors capability, a sound business case must be made to show the potential benefits. In practice, this is not easy as the costs and benefits associated with typical human factors activities are difficult to quantify. A detailed explanation of making a human factors business case can be found in Trenner and Bawa (1998). However, the key point is that only early integration of human factors in the design and system life-cycle promotes solutions that take account of human capabilities and limitations. The safety-related benefits include: enhanced usability, reduced error rates and improved in-service performance. Early integration of human factors into the design process helps reduce the number of design changes and associated costs throughout the whole product life-cycle.

From this brief discussion on human factors it can be concluded that the difficulties associated with the analysis and evaluation of cognitive human factors, scope and the cost-benefit quantification are just three reasons why systems safety-related systems develops may neglect human factors when making safety cases.

Having established that a problem exists with human factors and safety assurance and having explored some of the reasons why it exists, this paper will outline a process of integrating human factors with System Safety Engineering. However, before that can be done, some safety-related concepts need to be described here to provide a foundation upon which the human factors method may be presented.

3 Safety Foundations

This section will start with a brief description of Reason's popular Barrier Model (1997). The Barrier Model will then be related to Barrier Risk Model which will introduce the concept of success and failure cases. The concept of success and failure cases provides a foundation for the human factors process introduced later in the paper.

3.1 Barriers and Risk

A useful safety model for systems developers is the popular Barrier Model adapted from Reason (1997) and illustrated in Figure 2.

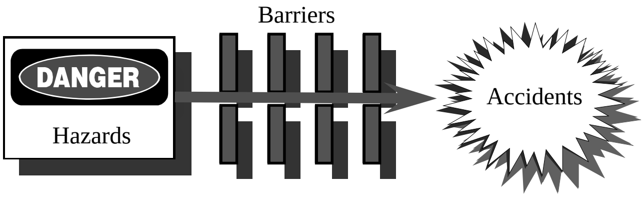

Figure 2: Barrier Model

Figure 2 shows the relationship between hazards, barriers and accidents for systems with a primary aim of controlling risk. Figure 2 shows that these systems have a number of barriers to prevent hazards from becoming accidents. In practice these barriers can be provided by any combination of people, procedures or technical equipment. More importantly perhaps, the model shows that barriers are not infallible. Barriers can be breached and when they do accidents will occur. Reason (1997) suggests that human, technical and organisational factors are all likely to be implicated in barriers failing.

When considering risk, and the forthcoming risk model involving success and failure, the most important attribute that barriers have in common is that none of them (either singly or in combination) is 100% effective even when working to full specification. This leads to some important conclusions regarding safety and provides the basis for the Barrier Risk Model derived from Sandom and Fowler (2006) and shown in Figure 3.

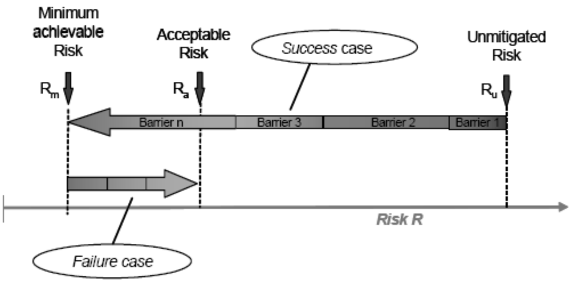

Figure 3: Barrier Risk Model

An excellent discussion on the role of barriers in accident prevention can be found in Hollnagel (2004). However, the Barrier Risk Model in Figure 3 is used to show a relationship between risks and barriers which are provided for safety-related systems designed primarily to provide risk reduction. The Barrier Risk Model in Figure 3 provides the basis for introducing the concept of success and failure cases and the risks levels shown on the model are explained as follows:

1. **Unmitigated Risk (R_u)**. Barriers need to be provided for systems where inherent risk exists at an unacceptable (unmitigated) level (R_u) in order to mitigate that risk. For example, aircraft flying in crowded airspace without Air Traffic Control (ATC) pose an unacceptable risk to society; therefore, the design of an ATC system provides various barriers.
2. **Minimum Risk (R_m)**. Risk cannot be eliminated totally unless the threat is removed entirely (i.e. stop flying) so the minimum level to which risk can be reduced (R_m) is determined by the properties of the barriers (e.g functionality, accuracy, capacity, speed of response etc) in the absence of failures.
3. **Acceptable Risk (R_a)**. Barriers are fallible and therefore the risk mitigation provided by each barrier is itself reduced by the undesired properties of each barrier (e.g. reliability, availability) causing risk to increase in the presence of failures. Clearly the net risk for any system must lie at or below the acceptable level (R_a).

This discussion on the risks associated with the fallibility of barriers within safety-related systems leads to the concept of success and failure cases.

3.2 Success and Failure Cases

Figure 3 shows that for any system that includes barriers any safety assessment of that system must address two key issues:

1. How safe it is when the barriers are working to specification, in the absence of failure. This is known as the *success case*.
2. How much less safe it is in the event of total or partial failure of a barrier. This is known as the *failure case*.

There is a widespread view (reinforced by some safety standards) that safety is largely a matter of reliability despite the fact that theory and experience have shown this to be far too narrow a view of safety (see Sandom and Fowler 2003). What the success case tells us is that one of the first considerations in assessing system safety must be whether the functionality and performance properties of the system are adequate to achieve substantially better than an acceptable level of risk.

Once the success case is established, only then is it worthwhile considering the failure case and the increase in risk associated with the failure-related properties of the system. This leads directly to the conclusion that Safety Requirements must take two forms:

1. Those relating to the required function and performance of the barriers – the *functional safety requirements*.
2. Those relating to the required reliability and availability of the barriers – the *safety integrity requirements*.

The prevalence of human mitigations and human failures in complex, safety-critical systems has already been discussed. If we accept that human factors can contribute significantly to the safety risks in these systems, then a credible safety case must explicitly address human success and human failure associated with human task, performance and integrity requirements.

Having established that a problem exists with human factors, and having also explored some of the reasons for the problem, the discussion in this section provided a foundation for the introduction of a means of specifying, validating and verifying human task, performance and integrity requirements within a systems safety engineering context.

4 Tackling Human Factors

There are a range of different Human Factors methods that can be used in system development. This paper is primarily focused upon the integration of Human Factors methods with System Safety Engineering activities during the systems development lifecycle. This section will introduce a typical high-level process for the specification of both technical and human safety requirements before looking in detail at the validation and verification of those requirements.

4.1 Safety Requirements Specification

A high-level design process for a safety-critical system requires the allocation and apportionment of system safety functions and system safety targets to sub-system functions and eventually to specific hardware, software or human components. Systems designers sometimes do the initial allocation of function in a haphazard manner allocating functions to humans or machines without fully considering the technical and human capabilities.

Human Factors support is typically requested once a design is complete and problems begin to emerge from the users. Human Factors analyses are then carried out and recommendations made to address shortcomings in the design. Ironically these recommendations are often ignored due to the high costs associated with redesign. The challenge for safety-related systems developers is to specify safety requirements (task or function, performance and integrity) for both technical and human functions in a timely manner during design taking into consideration both negative and positive aspects of technology and humans.

Figure 4 shows a diagram of a general process for the specification of technical and human safety requirements for a typical safety-related system requiring both technical and human subsystems. Following the high-level design activity an initial Allocation of Function activity should be undertaken by system developers to allocate primary safety requirements (task or function, performance and integrity) to both technical and human subsystems. To address the human factors in a timely manner, rather than as an afterthought, the allocation of function should be done during the high-level design in a systematic manner using suitable methods or techniques.

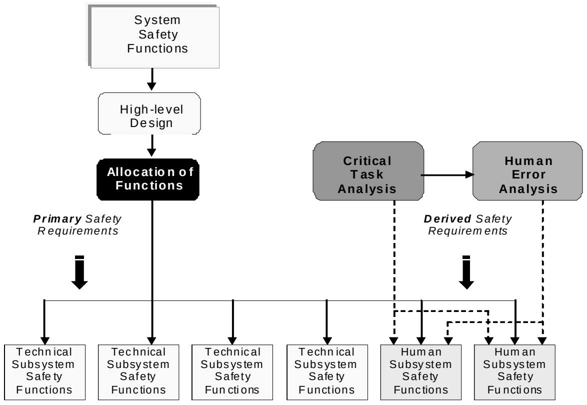
System Safety Functions
High-level Design
Allocation of Functions
Critical Task Analysis
Human Error Analysis
Primary Safety Requirements
Derived Safety Requirements
Technical Subsystem Safety Functions
Technical Subsystem Safety Functions
Technical Subsystem Safety Functions
Technical Subsystem Safety Functions
Human Subsystem Safety Functions
Human Subsystem Safety Functions

Figure 4: Simplified Specification Process

It is highly desirable for systems engineers to validate both human and technical requirements and to verify that they have been properly implemented throughout the phases of the development lifecycle. However, the methods and techniques used to validate and verify human and technical requirements will be different. Figure 4 shows that a combination of Human Factors methods described as Critical Task Analysis (CTA) and Human Error Analysis (HEA) may be used to validate an initial allocation of function to the human components and in particular the subsequent human performance and integrity requirements. Figure 4 also suggests that derived safety requirements may be generated from these CTA and HEA activities undertaken primarily for validation and verification of the primary safety requirements.

For the purpose of this paper validation is defined here as confirmation that the behaviour of the system meets the user needs (i.e. correct specification) while verification is defined as confirmation that the *product* of a system development *process* meets its specification.

Before examining these CTA and HEA activities it is first useful to look in more detail at the types of human tasks typically required for safety-related systems and to relate these to the concept of success and failure discussed previously.

4.2 Human Tasks and Technical Functions

The allocation of functions between humans and machines, and defining the extent of operator involvement in the control of the system, is a critical activity in safety-related systems. An important feature is that the high-level design must take into consideration the human factors in the initial allocation of Safety Functions. Too often, this decision is based upon technical capability and the human is allocated whatever functionality can't be implemented in hardware or software, regardless of the suitability of the human to undertake the resultant tasks.

The production of a high-level architectural design requires initial decisions to be made on the allocation of functions to human or equipment sub-systems, in full knowledge of the safety risks involved. Functional allocation decisions need to be informed by good human factors principles.

The Barrier Risk Model in Figure 3 makes a clear distinction between success and failure and relates that to the acceptable level of risk at the overall system level using people, procedures and equipment to implement system functionality. Likewise, a clear distinction needs to be made between the human success and failure cases as follows:

1. **Human Success Case**. The intention of the human success case is to assess whether the tasks allocated to the human can be undertaken safely and to identify all the support (e.g technical, procedures, tools etc.) that the human would require while undertaking those tasks.
2. **Human Failure Case**. The intention of the human failure case is to identify human error potential and assess reliability when specifically related to the dangerous human errors of commission or omission. In addition, the failure case must identify any human tasks arising from the need to mitigate machine failures.

Figure 5 shows a classification of the different human tasks and technical functions associated with a typical safety-related system that would contribute to the success and failure cases.

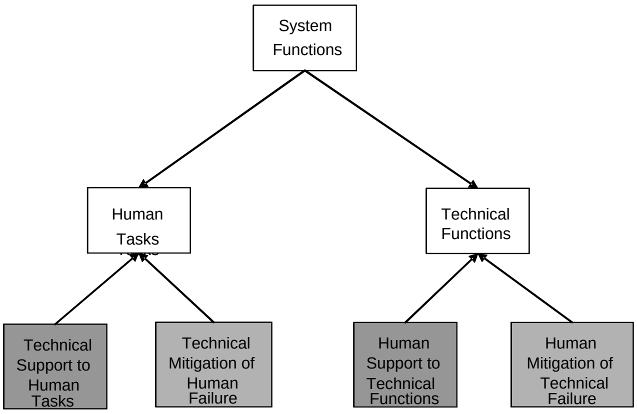
System Functions
Human Tasks
Technical Functions
Technical Support to Human Tasks
Technical Mitigation of Human Failure
Human Support to Technical Functions
Human Mitigation of Technical Failure

Figure 5: Human Tasks and Technical Functions

Figure 5 shows the system-level human *success case* requirements for tasks and functions, which are typically as follows:

1. Identify what additional human tasks are needed to support the technical functions (e.g. operation, insertion of data etc.).
2. Identify what additional technical functions are required to support the human tasks at a specified level of performance (e.g. information, computation etc).

In addition, Figure 5 shows the high-level *failure case* requirements for tasks and functions, which can be summarised as follows:

1. Technical mitigations of potential human errors.
2. Human mitigation of potential technical failures.

A summary of success and failure from different perspectives is given in Table 1 and it can be seen that a human success case requires the specification of achievable human tasks to include the successful provision of human mitigation for technical failures where possible.

Case Type	**Human View**	**Technical View**	**System View**
SUCCESS	Human Tasks	Technical Functions	Absence of failure
FAILURE	Human Error (of success tasks and tasks for mitigation of technical failures)	Technical Failure (of main functions and functions for mitigation of human errors)	Failure (of Tasks, Functions and mitigations)

Table 1: Success and Failure Case Summary

Focusing on the human view from Table 1, it can be seen that a process is required for the validation and verification of human tasks, human errors and human mitigations (of technical failures).

4.3 Human Safety Assurance

As mentioned previously, human operators can have both positive and negative influences on system safety and humans can alternatively be considered as 'hazard' or 'hero' depending upon the circumstances of the specific system interaction. Figure 6 introduces an approach to the integration of safety engineering activities and human factors analyses that has been used successfully on various safety-related programmable systems. This approach has been undertaken for the specification, validation and verification of human safety requirements (task, performance and integrity) and their contribution (both success and failure) to safety assurance provided by a system safety case.

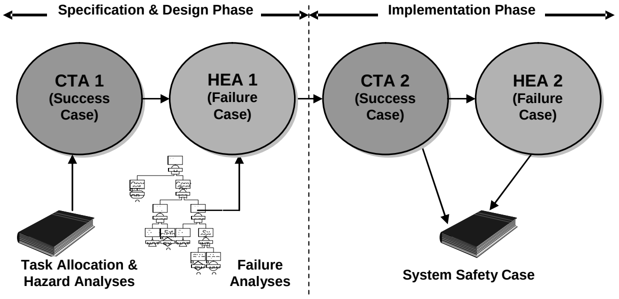

Figure 6: Human Safety Assurance Process

Figure 6 shows a high-level human safety assurance process comprising of two different safety-related human factors analyses described as CTA and HEA. CTA and HEA are high-level descriptions of analyses, which may be undertaken using single, or multiple combinations of the many human factors Task Analysis, Human Error Identification and Human Reliability Analysis methods and techniques available (see Sandom and Harvey (2004), Stanton *et. al*. (2005) or Kirwan and Ainsworth (1992) for detailed discussions on different methods and techniques).

It is important to note that CTA deals only with the safety-critical tasks and likewise HEA deals only with safety-critical human errors. Other human factors analyses may have a wider scope and they may address wider usability issues that are not directly safety-related. Both CTA and HEA analyses should therefore be planned to ensure that there is no unwanted (and costly) overlap with any human factors programme should one exist.

These human factors activities are entirely complementary as CTA and HEA are bottom-up and top-down analysis techniques respectively (from a hazard to human event perspective). As discussed previously, this combination of top-down and bottom-up analyses significantly increases the probability of identifying inconsistencies in the individual techniques and thus enhances safety assurance. Typically, a minimum of two iterations of each analysis should be undertaken to cover both system design and implementation phases and, as the analyses become more focused, the results from each one will inform and focus the other.

The remainder of this paper will examine the broad issues relating to specific human factors methods and techniques that can be used to undertake CTA and HEA. Such techniques aim to generate detailed evidence to support the human success and human failure cases and contribute to the overall system safety assurance.

5 Human as Hero (Success Case)

A human success case needs to be made to address the high-level human tasks including human support to technical functions, which together make up the primary human safety requirements. These tasks contribute to or constitute the system barriers that are required to reduce an unacceptable level of risk (R_u) to a minimum level of risk (R_m) *in the absence of failure*. It can be seen from Figure 6 that the human success case argument is based upon CTA activities undertaken during the design phase for the initial specification and validation of human safety requirements and also during the implementation phase for the subsequent validation and verification of those requirements.

Before looking in more detail at the CTA process it is useful to examine the characteristics and aims of the underlying analysis techniques used.

5.1 Task Analysis

Task Analysis (TA) is a general term applied to the process that identifies and examines tasks performed by humans, or groups of humans, as they interact with systems. Broadly, TA seeks to promote appropriate job and task design, suitable physical environments and workspaces, human-machine interfaces and the appropriate selection, training and motivation of the humans involved. At the detailed level TA examines how the design of human-computer interactions can foster the efficient transmission of information between the human and machine, in a form suitable for the task demands and human physical and cognitive capabilities (i.e. performance).

TA is a method supported by a number of specific techniques to collect and organize information to build a detailed picture of the system from the human perspective. TA activities can be characterized as being undertaken for one or more of the following broadly defined purposes:

1. Interface design or assessment.
2. Task and procedure design or assessment.
3. Personnel selection.
4. Operability and workload assessment.
5. Training requirements or assessment.

TA can be undertaken to identify and analyse the human performance issues in critical tasks. TA is a bottom-up technique used broadly to analyse the relationships between system hazards and operational tasks and the HMI design. The analysis works in a bottom-up fashion from operational tasks, related to base events, to identified service-level hazards.

TA will also typically look for opportunities for hazard mitigation through identification of human error potential and improved information presentation by comparing the TA with HMI design guidelines from appropriate sectors. In summary, the TA will enable the safety-related system developer to:

1. Define the allocated safety functions in terms of human operator tasks, including potential mitigations to be provided by the Operator in the event of failure of technical subsystems.
2. Capture the interactions and interfaces between the human and equipment subsystems.
3. Determine task skills, knowledge and procedure requirements and record these as additional functional safety requirements.
4. Confirm feasibility regarding human capabilities performance and reallocate inappropriate tasks to equipment (i.e tools, automation etc) as functional safety requirements.
5. Identify training requirements and record these as functional safety requirements.
6. Determine human information requirements and human-machine interaction requirements and record these as functional safety requirements.

5.2 Critical Task Analysis Process

CTA is used to focus various TA methods on specific safety issues rather then examining the system as a whole. With reference to Figure 6, CTA is undertaken initially during the Specification & Design Phase primarily to take the high-level allocated human functions (primary safety requirements) and to produce a low-level specification of human safety requirements (derived safety requirements) using two distinct CTA analyses to focus on tasks and performance separately. During the implementation phase, another CTA is undertaken to provide assurance that the human task and performance safety requirements have been achieved by the implemented system.

5.2.1 Specification and Design Phase

The first CTA activity is undertaken at the end of the requirements specification phase and starts with the high-level allocated human tasks and human support tasks to apply appropriate Task Analysis (TA) methods to produce a detailed representation of these high-level tasks. Various TA methods can be used for understanding the required human-machine and human-human interactions and for breaking down tasks into component task steps or physical operations. The resulting task descriptions will describe the tasks in terms of the individual steps required, the technology used to complete the task (controls, displays, input devices etc.) and the sequence of the task steps involved.

The task description will almost certainly contain a mixture of tasks from which some will be more important to safety than others. The CTA should concentrate initially on the identification and analysis of the relationships between system hazards and safety-related tasks. Consequently, the task descriptions must be analysed using input from functional hazard analyses, usually undertaken by system safety engineers, to identify the safety-related subset of task steps, operations and interactions.

This CTA analysis will enable both the functional hazard analyses and task analyses to be checked for consistency, providing confidence in subsequent safety assurance claims. Any deficiencies, such as hazards with no related tasks or safety-related tasks with no relationship to identified hazards, can be highlighted. The resulting task descriptions resulting from CTA are then typically used as the input to the specification and design phase HEA activity.

Once the task descriptions exist it is possible to undertake Human Factors analyses on them to identify the human performance requirements. The initial TA should focus on human performance aspects relating to the design of the human tasks including high-function cognitive functions such as: attention; vigilance; situation awareness etc.

These specification and design phase CTA activities will result in human performance requirements associated with each safety-related task and together these constitute the derived human safety requirements.

5.2.2 Implementation Phase

A CTA is undertaken during the implementation phase to generate evidence for the success case and provide verification that the human safety requirements have been implemented as specified. The implementation phase CTA will take the output from the specification and design phase CTA and HEA to focus analyses on the most critical aspects of the design.

The first CTA activity is to revisit the task descriptions generated during the specification and design phase to verify that they are consistent with the actual implementation of the system. In practice, modifications will need to be made. Once the task descriptions are consistent various task requirements evaluation methods can be applied including detailed assessment of the system interface analysis methods such as checklists, link analysis or layout analysis.

The output from the implementation phase CTA activities should be success case evidence relating specifically to the specification and implementation of the human safety requirements and contributing to the overall system safety case.

6 Human as Hazard (Failure Case)

A human failure case must be made to address the human errors and human mitigations of technical failures associated with the primary human safety requirements. There are two important points regarding human failures and mitigations related to the Barrier Risk Model in Figure 3 as follows:

1. Human failures are latent conditions within system barriers (after all, to err is human). Systems designers must ensure that all human failures associated with system barriers increase risk to an acceptable level (R_a).
2. Human mitigations of technical failures provide a link between the success and failure cases. These mitigations contribute to or constitute the system barriers, which are used to reduce an unacceptable level of risk (R_u) to a minimum level of risk (R_m).

It can be seen from Figure 6 that the human failure case argument is based upon HEA activities undertaken during the specification and design phase for the initial validation of human safety requirements and also during the implementation phase for the subsequent verification of those requirements.

Before looking in more detail at the HEA process it is useful to consider the nature of human error and the limitations associated with the analysis of human behaviour and in particular human reliability prediction.

6.1 Human Error

Human Error is defined by Reason (1990) as all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to the intervention of some chance agency. Human error analysis methods can be used both during the design process to identify potential design induced errors and to identify error potential once a system is implemented.

Error analysis methods can be broadly classified as either qualitative or quantitative. Qualitative approaches are used to identify potential errors and to determine the types of errors that might occur. Quantitative approaches are used to provide a numerical probability of error occurrence. The analysis of human errors should be a systematic process, integrated with the wider safety engineering activities, that includes both qualitative and quantitative methods for the analysis of safety-related systems.

A qualitative analysis is required to identify potential errors and to determine the types of errors that can arise prior to any attempt at quantification. A systematic identification of all possible human errors must be undertaken before any quantitative analysis can commence. If the qualitative error identification process is sufficiently comprehensive, valuable insights will emerge with regard to the sources of risk, and where limited resources should be most cost effectively applied in minimising these risks.

In addition, Performance Shaping Factors (sometimes referred to as Performance Influencing Factors or Error Producing Conditions) must also be identified and later quantified as these are direct and indirect factors that influence the likelihood that a task will be performed successfully.

Once human errors have been identified, a quantitative analysis needs to be undertaken to enable Human Integrity Targets (HIT) to be specified. However, the derivation of quantitative human integrity targets is not a simple task and Human Reliability Analysis (HRA) techniques have attempted to address this issue (see Kirwan 1994).

The difficulties arise because much of the HRA research has been dominated by assumptions that apply to technical systems and arguably these do not translate well to human systems. While the failure probability of hardware can be largely predicted by its basic design and its level of use, human error probabilities are influenced by a much wider range of contextual factors, such as the quality of the training, the design of the equipment and the level of distractions.

6.2 Human Error Analysis Process

HEA is used to focus the various Human Factors methods associated with human error on the safety-related human tasks. With reference to Figure 6, HEA is undertaken initially during the Specification & Design Phase primarily to take the safety-related task descriptions and performance criteria from the initial CTA and reconcile these with other safety-related analyses such as Fault Trees to ensure that all hazardous human errors have been identified. In addition, the initial HEA is used to specify human integrity requirements (derived requirements).

During the implementation phase, another HEA is undertaken to provide assurance that the human task and performance safety requirements have been achieved by the implemented system.

6.2.1 Specification and Design Phase

The first HEA activity is undertaken during the specification and design phase and has the CTA and other functional failure analyses (e.g. Fault Tree Analysis) as its inputs. This initial HEA is undertaken to achieve the following:

1. The specification of Human Integrity Targets relating to the success-case human tasks.
2. The specification of Human Integrity Targets relating to the human tasks required to mitigate technical failures.
3. The identification of additional technical mitigations required to mitigate human failures.

A pragmatic method of specifying a HIT is to undertake a HRA focused specifically on the basic human events identified by the system safety analyses such as the system Fault Tree Analyses. For systems that typically have a high degree of operator interaction, many basic FTA events will be identified as human interactions.

Once each fault tree is modelled, predictive, quantitative failure data can be input from hardware and software availability and reliability data for all hardware and software base events. By subtracting these values from the associated overall hazard target, a quantitative HIT can then be calculated for each critical human event. The HEA should then focus on developing individual safety arguments for each basic human event to provide evidence that the HIT can be achieved.

For critical areas, where the HEA reveals that a HIT is unrealistic, mitigations can be re-assessed and recommendations developed for further action. In this way, no predictions are being made about the human error rates; rather, a HIT is derived from the remaining integrity requirements once the hardware and software failure data is input and an analysis is undertaken to ascertain if the remaining human integrity requirements are realistic.

Finally, the initial HEA process may reveal derived safety requirements in the form of additional technical mitigations required to mitigate unachievable integrity targets.

6.2.2 Implementation Phase

An HEA is undertaken during the implementation phase to generate evidence for the failure case and provide verification that each HIT has been achieved by the implemented system as specified. The implementation phase HEA should focus on the most critical tasks as directed by previous CTA and HEA activities and these should be reconciled against the actual human-machine implementation to ensure that each HIT has been achieved.

In practice, the implementation phase HEA can use some of the many existing human factors methods for the verification of the system against the task descriptions and other safety-related analyses.

Finally, the output from the implementation phase HEA activities should be failure case evidence relating specifically to the specification and implementation of the human integrity requirements and contributing to the overall system safety case.

7 Conclusions

This aim of this paper was not simply to raise awareness of Human Factors issues related to safety-related systems but, perhaps more importantly, to introduce a practical process to address Human Factors issues within the context of the System Safety Engineering life-cycle.

This paper outlined how the specification, validation and verification of human safety requirements can be integrated with a typical System Safety Engineering life-cycle. The paper then described how specialist human factors methods can be used to generate the evidence required to support human success and failure cases as part of an overall system safety case.

Given the massive scope and complexity of the human issues which must be addressed for dynamic and unpredictable systems, Human Factors issues in safety-related systems are not easy to deal with. Human operators can have both positive and negative influences on system safety and these must both be addressed in system safety cases.

System safety engineering is still a maturing discipline; however, many of the safety assurance issues related to hardware and software are now relatively well understood. In contrast, the same cannot be said for the Human Factors.

There should be no doubt that adequately addressing Human Factors is the major challenge facing system safety engineers in the future in the ongoing struggle to make significant improvements to system safety.

8 Acknowledgements

The author would like to acknowledge Derek Fowler for his significant contribution to the concept of success and failure cases and Tony Cant for his helpful comments on a draft of this paper.

9 References

Hollnagel, E. (2004): *Barriers and Accident Prevention*, Ashgate, UK.

Khandpur, R. (2000): Human Factors in Ship Design, Technical Report, Ship Technical Operations Committee, Panel O-38, Human Factors (HF) and Manning, http://www.sname.org/technical_committees/tech_ops/panel038_reports.htm. Accessed June 2007.

Kirwan, B., and Ainsworth, L. (1992): *A Guide to Task Analysis*. Taylor and Francis, London.

Kirwan, B. (1994): *A Guide to Practical Human Reliability Assessment*. Taylor and Francis, London.

Pheasant, S. (1996): *Bodyspace - Anthropometry, Ergonomics and the Design of Work*. 2nd Ed., Taylor & Francis, UK.

Reason, J. (1990): *Human Error*. Cambridge University Press, UK.

Reason, J. (1997): *Managing the Risks of Organizational Accidents*. Ashgate, England.

RTCA/DO-178B. (1992): *Software Considerations in Airborne Systems and Equipment Certification*.

Sandom, C., and Fowler, D. (2003): Hitting the Target - Realising Safety in Human Subsystems, *Proc. 21st International System Safety Conference*, Ottawa, Canada.

Sandom, C. and Fowler, D. (2006): People and Systems: Striking a Safe Balance Between Human and Machine. in Redmill, F. and Anderson, T. Developments in Risk-based Approaches to Safety, *Proc. 14th Safety-Critical Systems Symposium*, Bristol, UK, 7 – 9 February 2006, Springer-Verlag.

Sandom, C., and Harvey, R. S. (2004): *Human Factors for Engineers*, The Institution of Electrical Engineers, UK.

Stanton, N., Salmon, P., Walker, G., Baber, C., and Jenkins, D. (2005): *Human Factors Methods: A Practical Guide for Engineering and Design*. Ashgate, England.

Trenner, L., and Bawa, J. (1998): *The Politics of Usability*, Springer-Verlag, Berlin.