

Appendix E

Techniques

This appendix provides additional guidance on the execution of the following techniques

- 1 Failure Mode and Effects Analysis (FMEA) (see chapter 8)
- 2 Hazard and Operability Studies (HAZOP) (see chapter 8)
- 3 Fault Tree Analysis (see chapter 8)
- 4 Cause Consequence Diagramming (see chapter 8)
- 5 Data Recording and Corrective Action System (DRACAS) (see chapter 12)

E.1 Failure Mode and Effects Analysis (FMEA)

FMEA should be carried out in compliance with established standards such as BS 5760 [F.20].

Note that users of this standard should ensure that they use a common set of units, if they wish their risk ratings to be comparable

The analyst should consider components at a detailed level of indenture and record their failure modes along with causes and effects. The failure effects of these sub-components then become failure modes of components at the next higher level of indenture. The process is repeated up the functional hierarchy to yield the individual failure modes of the whole system.

The depth of the analysis should be adjusted according to the preliminary appraisal of the hazards. The components which contribute to more severe hazards should be analysed in greater detail.

Checklists, HAZOP or other techniques may be used to identify basic failure modes.

The analysis is recorded on a worksheet which has at least the following columns:

Item Ref	The unique identifier of the sub-component being considered.
Description	A description of this sub-component.
Failure Ref	A unique identifier for the failure mode entered.
Mode	A description of the failure mode.
Causes	For this failure.
Effect	Of this failure (local and system-wide).
Compensating Provisions	Which may cause the effects of this failure not to be promulgated.
How detected	The means by which the failure may be detected.
Remarks	Any other notes made by the analyst.

This conforms to the British Standard for FMECA [F.20 §2.2.3] except that there is no column for 'Severity of effects'. Criticality is considered instead during the later stages of Risk Assessment although note that FMECA may be more appropriate for some applications.

E.2 Hazard and Operability Studies (HAZOP)

Where detailed design information is available and a high level of assurance is required a Hazard and Operability Study or HAZOP can be carried out.

HAZOP is a systematic, creative examination of a design by a multi-disciplinary team.

HAZOP is recommended for systems with potential catastrophic accidents, novel features or for systems that span several engineering disciplines.

HAZOP is an analysis technique developed originally for the chemical industry and described in the Reference Guide [F.21] and Interim DEF-STAN 00-58 [F.19]. The technique should be carried out as described in these documents.

The principal difference between application of HAZOP in the chemical industry and application in other engineering fields is in the way in which the design documentation is examined. In the chemical industry, examination is guided by traversing the flowchart, a schematic showing the connection of vessels, pipes and valves. In engineering applications an alternative representation of the parts and their interactions, such as a mechanical drawing, circuit schematic or data flow diagram should be used. The same technique can be applied at a number of levels within the design.

If no convenient form of the design exists then the analyst should construct a **Functional Block Diagram**. At each level of indenture this shows the components of the system or a sub-system as blocks with lines drawn between each pair of boxes that directly interacts.

The team collects the design documentation, including a full functional breakdown of the system. Each component, including the interfaces, of the system is inspected in turn. The team considers the **intention** of the system and by applying a list of **guide words** attempts to reveal plausible **deviations** from the design intention.

The guide words for hardware systems typically are as follows. Alternative guide words for Programmable Electronic Systems are considered in MOD Interim Def Stan 00-58 [F.19].

- | | |
|---------------|---|
| a) NO or NOT | No part of the intention is achieved but nothing else happens |
| b) MORE | Some quantitative increase over what was intended |
| c) LESS | Some quantitative decrease over what was intended |
| d) AS WELL AS | Some qualitative increase over what was intended |
| e) PART OF | Some qualitative decrease over what was intended |
| f) REVERSE | The logical opposite of the intention happens |
| g) OTHER THAN | Something quite different happens |

The team should be constituted to cover the areas of expertise required to fully understand the system. For example, the examination of a signalling system may require a safety process expert, a hardware engineer, a software engineer, an expert in signalling principles and potential users and maintainers.

It is quite likely that the team will be unable to establish immediately whether a possible deviation can occur or what its effect can be. In that case an action can be recorded to establish this outside the meeting.

E.3 Fault Tree Analysis

Fault Tree Analysis (FTA) is a widely known and accepted top-down or deductive system failure analysis technique. The Fault Tree Handbook, NUREG-0492 [F.22], is a comprehensive reference document for FTA, and may be used in conjunction with other FTA standards.

FTA begins with a single undesired top event and provides a method for determining all the possible causes of that event.

A correctly constructed Fault Tree is a graphical and logical model of the various parallel and sequential combinations of events that will result in the occurrence of the Top Event.

FTA can be used for both qualitative as well as quantitative analysis. The graphical nature of the technique aids the qualitative identification of potential sources of single-point failures and safety critical failure combinations.

The precise definition of the top event is critical to the success of the analysis, since an incorrect top event will, in most cases, invalidate the whole analysis.

The system is analysed, from the identified top events, in the context of its environment, and modes of operation, to find all credible causal events.

The fault tree is made up of gates, which serve to permit or inhibit the flow of fault logic up the tree. The gates show the relationship of lower events - the inputs to the gate - needed for the occurrence of a higher event - the output of the gate. The gate symbol denotes the relationship of the input events required for the output event.

The fault tree is used to produce the minimal cut sets - the minimum combination of independent base events which, if they occur or exist at the same time, will cause the top-event to occur. The minimal cut sets provide the basis for both the qualitative and quantitative analysis of the system.

Fault Trees are relatively simple in concept, but can be very difficult in practice. This is particularly true when quantitative analysis is required. Chapter V of NUREG-0492 [F.22] provides a detailed explanation of the technique. The following key concepts and rules from that document are given here to guide the analyst in the approach required to the construction of the tree.

In determining the causes of an event in a fault tree, the analyst should identify the **immediate, necessary and sufficient** causes for the occurrence of that event. The temptation to jump directly to the **basic** causes should be resisted, even if these may appear obvious.

The dependence between base events within a minimal cut set should be identified during FTA. This is achieved by performing Common Cause Failure Analysis on the Minimal Cut Sets to identify potential dependencies.

The following basic rules should be applied when constructing a fault tree:

- a) Write the statements that are entered into the event boxes as faults: state precisely **what** the fault is and **when** it occurs.
- b) If the answer to the question 'Can this fault consist of a component failure?' is 'Yes', classify the event as a '**State of Component Fault**'. If the answer is 'No', classify the event as a '**State of System Fault**'. If an event is classified as 'State of Component Fault', add an OR-gate below the event and look for primary, secondary and command faults that may cause the event. If an event is classified as a 'State of System Fault', an AND-gate, OR-gate, INHIBIT-gate, or possibly no gate at all may

be required, and the minimum, necessary and sufficient causes should be determined.

- c) If the normal functioning of a component propagates a fault sequence, then it is assumed that the component functions normally.
- d) All inputs to a particular gate should be completely defined before further analysis of any one of them is undertaken.
- e) Gate inputs are to be properly defined fault events. Gates are not to be connected directly to other gates.
- f) Identify fixed probabilities ie non-failure conditions, with inhibit gates.

E.4 Cause Consequence Diagramming

Cause Consequence Diagramming (or Cause Consequence Analysis) is a technique that embodies both causal and consequence analysis. However, in the context of the Yellow Book it is useful primarily as a consequence analysis tool.

The technique provides a diagrammatic notation for expressing the potential consequences of an event (normally a hazard) and the factors that influence the outcome.

The basic notation is introduced in the context of the example in figure Figure E-1. In this diagram the hazard is Ignition. The final outcomes (or 'significant consequences') are shown in octagons and vary from no incident to a major fire. The major factors that influence the outcomes are shown in 'condition vertices'.

The diagram shows that a major fire will only occur as a result of the ignition hazard if both the sprinkler and alarm system fail. If we can estimate the frequency with which the hazard will occur and the probability that the sprinkler and alarm systems will fail on demand (and, importantly, we know to what degree these failures are correlated) then we can estimate the frequency with which the hazard will give rise to this accident. This is an essential step on the way to estimating the risk arising from the hazard.

There are variations in notation. Railtrack have procured a tool, ACCA, which produces output in a slightly different format. There is an example of this output in appendix D.

The notation allows further symbols. For a slightly fuller exposition refer to 'Safeware: System Safety and Computers' [F.23], pages 332-335.

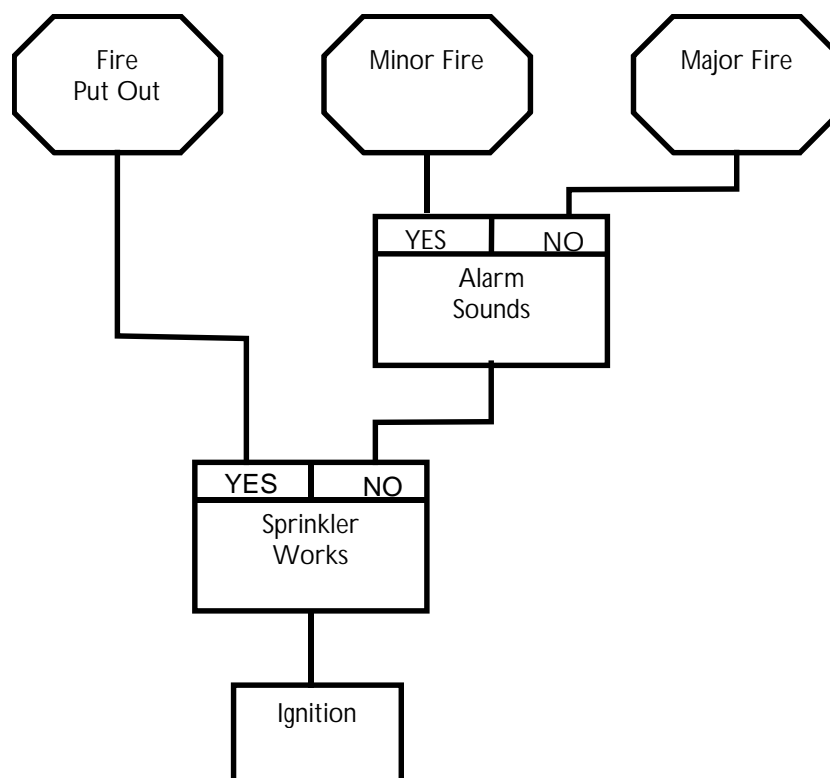


Figure E-1 - Example Cause-Consequence Diagram

E.5 Data Reporting Analysis and Corrective Action System (DRACAS)

The Data Reporting Analysis and Corrective Action System (DRACAS) is a closed loop data reporting and analysis system. The aim of the system is to aid design, to identify corrective action tasks and to evaluate test results, in order to provide confidence in the results of the safety analysis activities and in the correct operation of the safety features.

Its effectiveness depends on accurate input data in the form of reports documenting incidents. These reports should therefore document all the conditions relating to the incident.

The Project Manager or Project Safety Manager should be part of the team that reviews the incidents, in order that their impact on the safety characteristics of the system can be quickly assessed and any corrective actions requiring design changes quickly approved.

The DRACAS process is illustrated in Figure E-2 and may be summarised as follows:

1. The incident is raised and recorded on a database.
2. A data search is carried out for related events.
3. The incident is reviewed.
4. If the incident is a new hazard it is recorded as such in the Hazard Log.
5. Corrective actions are recommended as necessary.
6. If no corrective action is required the database is updated and the process ends.
7. The corrective action is authorised and implemented and assessed for success.
8. If the corrective action is successful the database is updated and the process ends.
9. If the corrective action is unsuccessful the incident is re-reviewed (the process returns to step 4).

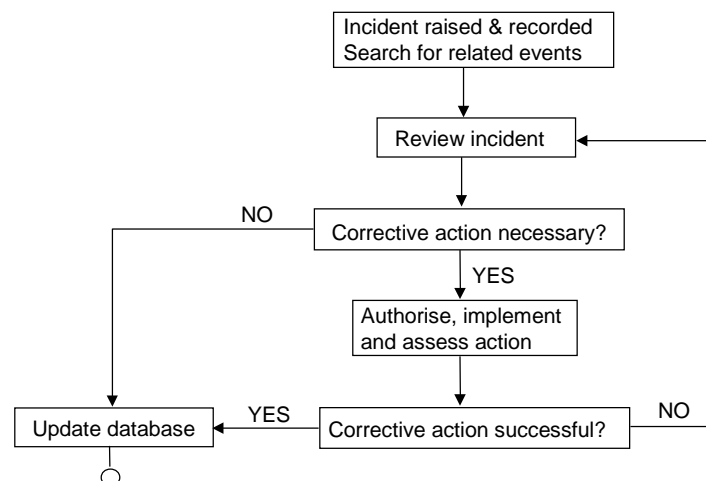


Figure E-2 - The DRACAS process

This page left intentionally blank