

Experiences in Auditing Critical System Suppliers and Integrators in the Process Chemical Sector against IEC61508

Paul Lucas & Eric Gilchrist ABB Eutech

Over the past three years ABB Eutech has performed many gap analyses and audits of safety critical system suppliers and integrators against the requirements of IEC61508. All of the systems have contained software, although many of the suppliers will state that they configure, rather than program the systems. We will discuss how standard software engineering activities, such as requirements capture, design, review, test and configuration control are being addressed, how software tool support is being used and what evidence is produced to support the development process.

1. INTRODUCTION

Within the UK process chemical sector, many system suppliers and integrators have embraced the philosophy and guidance of IEC61508 [1] for developing and supplying software based protective systems. IEC61508 is a generic standard designed to cover all sectors and act as a base for sector specific variants, for the process chemical sector IEC61511 [2] has been developed and is close to full publication. However, since the publication of IEC61508, purchasers have been quoting the standard in tender documents and regulators referring to it as a basis for good practice and expectation. This has resulted in system suppliers and integrators having to interpret the relevant clauses in the standard to their scope of work and demonstrate knowledge and compliance to the purchaser. Within this paper, we will consider some of the successes and difficulties experienced by the system suppliers and integrators in interpreting and applying the generic standard to their software development processes.

2. LIFECYCLES

One of the fundamental principles of the IEC61508 standard is the use of a defined lifecycle of activities. The Overall safety lifecycle is shown in figure 1. The majority of activities for system suppliers and integrators occur within phase 9, the Realisation phase, however there is much discussion as to the extent of involvement in other phases. It seems sensible for a system integrator to be involved with installation activities for example, but we have experienced system integrators providing significant input to the safety requirement allocation and overall safety requirement phases, mainly due to the knowledge and experience of the system suppliers and integrators. System integrators have stated that the requirements that they receive from the client can demonstrate a lack of understanding of the principles of IEC61508.

They have received requests for a 'SIL2 system' or an 'IEC61508 compliant system'. Against this background, integrators feel that they can assist their clients by offering advice on the earlier stages of the lifecycle.

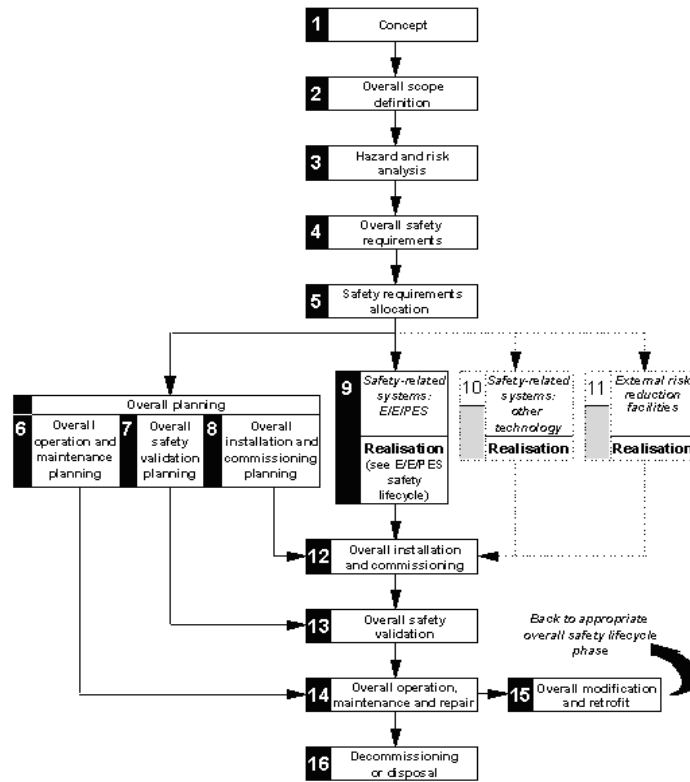


Figure 1 - Overall safety lifecycle

Within phase 9, there are two lifecycles defined for the software requirements. These are contained in Part3 and describe the software safety lifecycle and the software development lifecycle, more commonly known as the 'V' model.

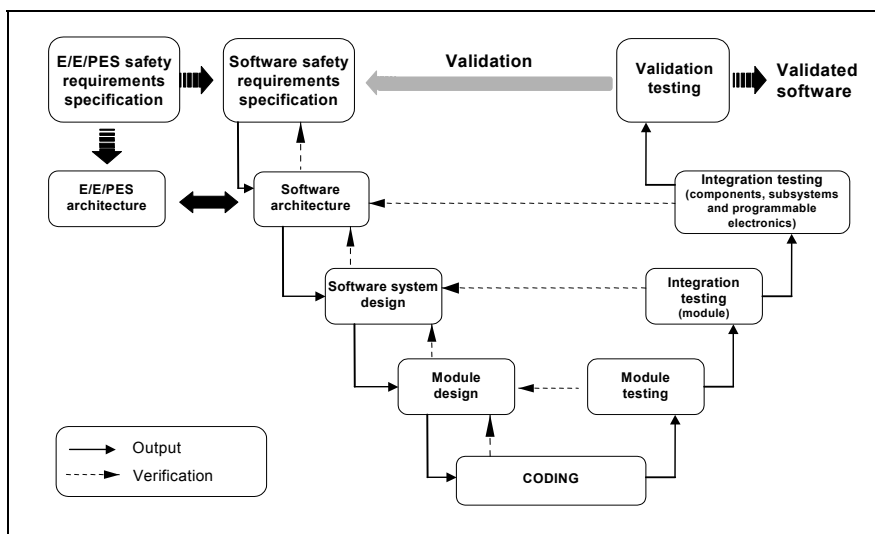


Figure 2 - Software safety integrity and the development lifecycle (the V-model)

All of the companies that we have visited have incorporated the requirements of the two lifecycles within their business processes to form a defined set of activities and checks to control their development process. The scope of their activities has seen them compress the 'V' model, combining activities and documents to be more in line with their current working practices. As indicated in the Part 3 of the standard in the clauses on requirements for software architecture, where system integrators are utilising safety platforms from another supplier several of the lower levels of the 'V' model are performed by the platform supplier. Supporting evidence and activities to ensure the suitability of the hardware will be discussed in the section on platform selection.

The activities defined in Part 3 are underpinned by associated sets of tables, which detail techniques and measures to be used during the phases of the development process for different target integrity levels. The interpretation of these tables and the selection of the appropriate techniques for a target integrity level has probably been the most challenging aspect of implementing the standard for many system integrators and suppliers. Part 6 Annex E contains two worked examples on interpreting the tables for hardware typically used in the process chemical sector. More explanation in these tables would have been helpful.

In order to simplify their procedures, some system suppliers have decided to perform the same set of activities no matter what target SIL is required. The techniques and measures selected are therefore set at a high integrity level (normally SIL 3) and could, in theory; result in over-engineering and extended effort for a system where the target integrity level is lower.

The first phase of the IEC61508 software safety lifecycle is the software safety requirement specification. In nearly all cases, the use of cause & effect diagrams to represent the required functionality is used. One company always created a cause and effect diagram, even if the user did not specify their requirements in this form. This is a form of matrix that uniquely defines actions for every initiating event and is often quoted as semi-formal method for requirement definition. This method of specification is strong in functionality description, and leads to an emphasis on black box testing later in the lifecycle. This method does have some drawbacks in that it hides the essential safety requirements; it gives you bare logic without supporting rational and relative importance of the actions.

3. PLATFORM SELECTION AND SOFTWARE ARCHITECTURE

Early in the design phase, IEC61508 introduces the requirement for consideration of the software architecture and the recommended techniques for internal diagnostics that the platform should implement to reach higher levels of integrity. For all of the system integrators, this set of activities posed many questions. Within this sector, many of the integrators offer only one or two system solutions; these are third party

certified 'safety' PLC or TMR technologies. Quoted by one system integrator as the 'Gold Standard', certification by TÜV is the most common method of assuring that the platform meets the integrity requirements of the safety functions.

AK
2
3
4
5
6
7
8

SIL
1
2
3
4

Until recently, the certification was against other, generally German, standards DIN V VDE 0801 [4] and DIN V 19250 [5], which generated an AK rating rather than a Safety Integrity Level. There is no basis for deriving a direct relationship between the two scales; one is based on rigour of build, the other on reliability. However, many integrators claim a SIL level capability based upon an AK rating.

Certification is now offered directly against IEC61508 and SIL levels will be quoted.

The certification will also cover the compilers, translators and development tools that support the product. The certification report and scope are substantial documents providing details of the tests that have been performed, and the constraints for the use of the system. It is tempting to take the view that the certification provides the assurance of the performance of the system, the more knowledgeable companies also take due regard of the limitations in scope provided by the certification report.

4. COMPETENCIES

There are examples of suppliers spending a significant amount of time and effort in interpreting the IEE/BCS Guidelines [3] for safety practitioners within their organisation. They have developed their own set of competencies as part of their internal staff appraisal process. This enables them to specify in their generic project process a job role as being appropriate for a task, and allocate staff to tasks based upon their assessed job title. This avoids repeated assessments for each new project. Others have adopted a more simplistic approach based upon Part1 annex B. This has involved recording not only qualifications and training, but also experience gained from project to project. The key requirement is to have sufficient information available to the project manager so that gaps in skill and competency requirements against the needs of the tasks can be identified. Some companies place a copy of the team competency records in the project file as evidence of the gap analysis and competencies. The most common failing that we have found in this area was that companies fail to capture evidence of relevant experience, there is plenty of detail on training and qualifications, but the guidance in the standard also states experience in the technology and sector as being an important competency.

5. AUTHORISATION

There has been a tendency for authorisation to proceed for an activity to be signed off by staff because they are in a senior position. The development of competency processes has led to a more defined methodology for authorisation. Companies are defining the competencies required for these sign-offs and maintaining a list of authorisers, although a common fault has been deputies not being on the list of authorised signatories.

There has been considerable improvement in the use of guidance to assist the authorisation activities. Companies have been using the tables specified in the standard to develop guidance checklists to assist the checker and provide evidence of the activity happening.

However, the concept of formal check and approve is not so advanced for software development as for hardware development. It is universally accepted by engineers to check, authorise and sign off engineering drawings, so why not apply the same discipline for software development?

6. PROGRAMMING AND CODING STANDARDS

The interpretation of the software guidance in IEC61508 has posed some difficulties for most software developers in the process industry. Many developers have stated that they don't program software. They state that they merely configure pre-defined logic blocks. This distinction is realised in IEC61511. However, some of the 'knitting' together of the function blocks involves complicated logic, and the development of new 'standard' blocks requires a degree of programming greater than just 'drag and drop'.

Most of the certified hardware platforms are supplied with a safety manual describing the correct and safe usage of the standard, 'certified' function blocks. We have seen examples where the copies of the appropriate page from the safety manual has been included in the project documentation, and completed as a method of ensuring that the standard block has been used in the correct manner.

IEC61508 recommends the use of certified or trusted function blocks. This is generally taken to mean in the industry to be blocks that the user has created and has developed a body of experience in its performance. In theory, the same level of documentation detailing the safe method of implementing the function as for the certified functions should be provided, including specimen test scripts. However, this is not generally the case.

Both for certified and 'trusted' blocks, reviews of the software should verify that the blocks are being used in the correct manner. At present, this seems to be done by hand

and part of the code review. Configuration control of these certified or 'trusted' blocks has been variable. Ideally some control must be implemented to ensure that the blocks are used as designed. Facilities to 'lock' both certified and 'trusted' blocks would assist with this requirement, however we have seen occasions where this had not been the case and blocks can be modified.

Other areas of configuration control have had a greater spread of interpretation. The best examples we have witnessed have put platform and development software as well as design documents and evidence of verification, validation and testing under configuration control with the application software.

Suppliers are developing coding standards that guide the engineer into standard ways of implementing common functionality, and define safe constructs. Coding standards have been developed for each platform and language used. Some suppliers have further developed these guidelines into checklists to prompt them through code reviews, and kept the completed lists as evidence of review. Compliance with coding standards is enforced during manual code review; we have seen no evidence of automated tool support of these activities.

7. VALIDATION, VERIFICATION AND TESTING

The industry seems very comfortable and capable in performing black box testing. This is supported by the common requirement method of cause and effect drawings that lend themselves to the black box method. Some of the tools that support the platforms give the capability for examining intermediate values during testing to give greater visibility to the testing process. However, the ability to automatically run pre-determined test scripts would assist in this area, but we have not come across any use, or knowledge of this type of support.

Having set up guidelines and standards to assist engineers in developing the software, how do you check that people follow the rules? There have been real difficulties in understanding the differences and requirements of verification, validation and function safety assessments.

Verification is consistent in that end of phase checks are being performed supported by guidance documents and generating evidence. The feedback from system suppliers and integrators is that they find these to be beneficial activities to detect and remove problems.

There has been much discussion about whether validation is really applicable for software when it is so far from the original safety requirements. The majority of suppliers and integrators perform this task on the integrated system.

Even more varied is the understanding and implementation of functional safety assessments. Can a system supplier or a logic box supplier who is not providing the complete safety function perform functional safety assessments or is this the

responsibility of the end user? IEC61511 clarifies the situation with its five-stage process, stage 2 functional safety assessment occurring after design and engineering of the safety-instrumented system, and before installation and commissioning.

8. CONCLUSIONS

In general, we have found the system suppliers and integrators in the process chemical sector to be knowledgeable engineers who participate in many industry forums and associations and probably understand the principles of the IEC61508 standard better than many of their clients.

Competency assessment processes have been introduced into nearly all the organisations that we have seen; many of these based upon the IEE/BCS competency guidelines. This has not been a trivial task. However, we have seen other interpretations based upon guidance in IEC61508 that have achieved a similar objective.

The difference between programming and configuring has made the interpretation of some of the software techniques and measures supporting the IEC61508 difficult, but the forthcoming IEC61511 will assist in this area. The reliance on certified hardware platforms reduces the software lifecycle, and is seen to provide confidence in the safe performance of the hardware. However, all parties do not explicitly perform investigation and understanding of the scope and limitations of the certificate.

There may be opportunities for improvements in tool support to lock both certified and user defined blocks, enforce safe constructs and assist with the automation of some of the testing. More advanced software measurement and testing techniques are not evident in the sector, probably influenced by the popularity of extensive black box testing derived from the semi-formal cause-effect method or requirement definition.

Several companies have reported benefits in the work methods from adopting the standard by developing template documents, phase specific guidance and checklists, improved configuration control and the re-use of certified and trusted program blocks.

9. REFERENCES

[1] IEC61508 - Functional Safety of electrical/ electronic/ programmable electronic safety-related systems

[2] IEC61511 - Functional Safety - Safety Instrumented Systems for the Process Industry Sector

[3] Safety, Competency and Commitment: Competency Guidelines for Safety-Related System Practitioners

[4] DIN V VDE 0801 - Principles for Computers in Safety Related Systems

[5] DIN V 19250 - Measurement and Control. Fundamental Safety Aspects for Measuring and Control Protective Equipment.