The Future of Requirements for Safety Critical Systems



The "Future of Requirements for Safety-Critical Systems" seminar was held on the 22nd September 2022 at the Chartered Institute for IT, London and virtually online. The seminar looked at the importance of requirements and specifications for safety-critical systems, the state of the art in requirements management and what can go wrong if requirements are not correct in some way, e.g. missing, ambiguous or incorrect. Bernard Twomey provides a summary of the event.

Mike Parsons introduced the seminar and seminar speakers and pointed out that a large number of failures could be traced back to the requirements, which in some cases were non-existent.

Requirements – How Not to get Them Badly Wrong…



Jane Fenn, BAE Systems, provided an insightful journey into the world of requirements and how defects at the requirements phase could have a significant impact on the demonstration of safety to the relevant regulatory authorities.

Jane stated that "it is never too early to start thinking about safety and isn't something you do at the end of the program" and then explained the principles of 'Architectural Trade Off', Functional and non-Functional safety, Validation of Requirements, and the use of Tool Supported Validation of Requirements.

Key points from the presentation included:

automated testing could result in requirement errors not being identified'

- There is no perfect solution
- do not be overly seduced by tools
- significance of getting the requirements right is going up…significantly.

Aircraft Accidents that have resulted from Incorrect Requirements

Dewi Daniels, Software Safety Ltd, provided a very structured presentation which started with a positive statement that*:*

'Not a single hull-loss accident in passenger service has been ascribed to failure of software to meet its requirements'. His next slide however, stated that '*There have been several accidents where the software behaved exactly as specified in its requirements, but those requirements specified unsafe behaviour in some unforeseen circumstances.*



Dewi presented several accidents, many of which resulted in fatalities. His presentation clearly described the accident causes with one case study raising the question on *'who is responsible for providing accurate information to the pilot when landing?*

He stated that the number of lines of code is increasing, and the use of outsourcing to software houses is a potential risk, as they may not have sufficient design experience. He also raised the issue of a loss of independence in the verification process.

Key points from the presentation included a realisation that the avionics industry is (surprisingly) good at deploying correct software, accidents have been caused by requirements errors e.g. the Maneuvering Characteristics Augmentation System (MCAS) software attributed to the Boeing 737 MAX accidents of 2018/2019 (see Boeing 737 MAX Safe to Fly? – Safety Systems Newsletter, Volume 29, Number 1) behaved exactly as specified in its requirements, and in one case the accident report blamed the pilots.

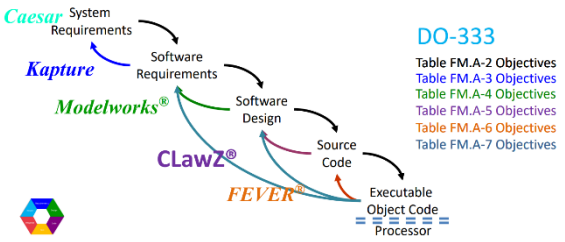
The presentation highlighted the necessity to get better at writing and validating requirements if we are to improve aircraft safety and prevent loss of lives.

Tackling the Uncertainties in Requirements

Nick Tudor, DRisQ Ltd, provided an insight into the development of automatic Formal Methods based verification technologies. His presentation identified the significant costs associated with verification activities which then led onto the need for requirements attributes to be 'Clear' and 'Precise', which were supported with a number of excellent examples.

Nick then went on to explain the problem in communicating our requirements to specialists and non-specialists alike. He stated that writing clear and precise requirements should be taught at university, as everything goes wrong from poor requirements capture. This was again, supported with excellent examples along with potential solutions, such as, adding 'Universal Quantification' that would help in defining a reasonably unambiguous statement of requirements.

Nick then explained the 'Tools' developed by DRisQ and how they could be used in the verification of requirements. Kapture – designed to automate compliance to a requirement standard. CAESAR provides a formal link between system requirements and software requirements, Modelworks verifies the design against Kapture, ClawZ automatically verifies the auto-code and FEVER provides automatic verification of the object code.

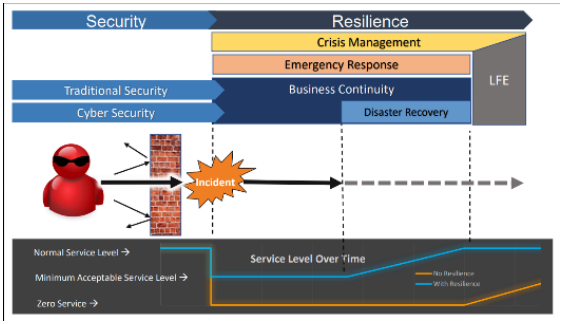


Key points from the presentation included a statement that '*nothing can tell you that you have the right requirements and there still needs to be a human review*', '*no tool can do everything for you*', and developing requirements is a social exercise and you should not expect them to be right first time.

One Million Monkeys or one Bored Teenager: The Challenge of Security Requirements

James Yolland, Ebeni Ltd, discussed the world of security and the development of security requirements. His presentation started with examples of security failures and how the threats are real and potentially harmful e.g. Vishing and virtual kidnappings. He then went onto explain the foundations for good security requirements relies on identifying your attackers capabilities and their motivations along with excellent knowledge of the system and the stakeholders through life.

James explained the goal of security is to protect the system and the stakeholders from unacceptable harm. He explained that following a successful general threat/hazard attack the risk largely remains unchanged with a clear recommendation that security requirements would need to be updated.



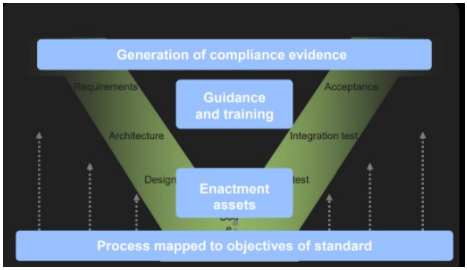
The presentation then proposed a recommendation for writing better security requirements, which included, aligning and engaging with the systems engineering lifecycle, understanding the systems and the stakeholders, identifying threats, vectors and critical assets and addressing the full lifecycle of the system.

Key points from the presentation included a realisation of the complexity of the systems domain and the necessity to have accurate knowledge and identification of the critical assets along with early engagement with the systems engineering lifecycle, preferably at the start of a project, not at the end.

Requirements Management for Safety Critical Systems

Steve Rooks, IBM, explained that poor requirements management has a significant impact on your business in the way of requirements rework, project delays and project impacts, citing a wide range of statistics to support his argument.

He explained that standards initially increase project costs due to 'inadequate level of detail and process for requirements, improper tool qualification (too much or too little), and weak process and checklist management.



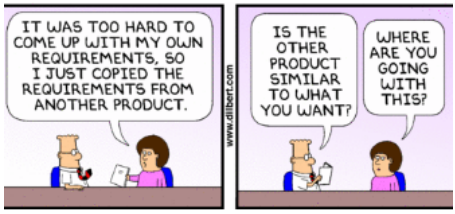
Steve then explained the principles of automating safety critical compliance, linking of data to support compliance, requirements management fundamentals for safety critical systems development and data models that include safety and cyber security.

Steve then explained the principles of IBM DOORS Next, FMEA and presented an excellent demonstration on how the tool has been used (e.g. Infusion Pump Requirements) and covered link validity, traceability and the use of an open link data set – Open Services for Lifecycle Collaboration (OSLC).

Configuration management was discussed which covered audit trails, variants, global configuration management, requirements validation and reporting.

Key points from the presentation included an acknowledgement that poor requirements capture is a risk that can be reduced by the use of IBM DOORS and management of requirements can be of benefit for many safety critical projects along with helping to improve quality and productivity.

Requirements Management



Sean Sullivan from 3DS Dassault Systemes, gave his presentation online and explained the principles of requirements engineering and the need to consider hardware and software together. He stated that the relationship between Elicitation (vision and scope, identified use cases), Analysis (data dictionary), Specification (derived requirements) and Verification (requirement verification) are extremely important.

The necessity for requirements to be under configuration management control during all steps of the lifecycle offers extremely valuable services to the end users. The message from Sean aligned with all the other presentations on the need for requirements capture and validation to be taken seriously by all relevant stakeholders.

The day finished with a lively panel discussion with a common theme of requirements that are clear and precise can help prevent accidents and could improve costs.

It was felt the day went very well with interesting and very useful presentations, whether the attendees will board an aircraft again is debatable!