SAFETY-CRITICAL VALIDATION
ABSTRACT

Software is now being used to perform critical functions in increasing numbers of systems and industries, such as the energy, aerospace, automotive, space and defence industries and the railway and healthcare sectors. This growth has come about because of the flexibility software can offer, the complexity it can handle, the high levels of functionality, complexity, configuration and adaptation it can offer and the rapidly growing and changing needs of modern society.

However, moving more and more critical functions to software is not a panacea. Mechanical, difficult to reproduce, human-dependent production processes, combined with the lack of maturity of software engineering disciplines, make it difficult to guarantee the safety and high integrity of systems.

We believe there is a light at the end of the tunnel for high integrity (safety-critical or mission-critical) system developers. To ensure high integrity, special attention must be given to the non-functional requirements related to safety, verification, validation, qualification and certification.

CRITICAL Software has provided professional engineering products and services in the field of high integrity systems for more than fifteen years. Our skills have been developed within the context of our diverse experience in safety/mission-critical systems in different industries. The services provided by our experts in dependability and safety systems allow our customers to focus on their core business by outsourcing all or part of the work related to non-functional aspects of safety, verification, validation, qualification and certification.

This paper presents CRITICAL Software’s vision for safety-critical validation in the form of three methods: Reliability, Availability, Maintainability and Safety (RAMS), Verification and Validation (V&V) including fault injection, formal methods and security assessment techniques and Qualification and Certification support.

For the remainder of this paper, these methods will be referred to as ‘safety-critical validation methods’; the term ‘safety-critical systems’ will refer to safety/mission-critical systems.
1. CRITICAL SYSTEMS IN CONTEXT

Software is now being used to perform critical functions in increasing numbers of systems and industries, such as the energy, aerospace, space and defence industries and the automotive, railway and healthcare sectors. [RD-4].

The increase in the use of software is due to its flexibility when compared to hardware and mechanical systems. The characteristics of a piece of software are easily modified and can be re-used without extra cost. This flexibility supports the provision of a significant number of critical functions, configurations and adaptations that can cope with the increasing level of complexity required by systems that must respond to the rapidly-growing and changing needs of modern society.

1.1 SYSTEMS EVOLUTION

Advances in computer systems mean that to perform competently, safety/mission-critical systems and applications must increasingly rely on embedded software. However, moving critical functions to software is no panacea and indeed, can significantly increase the criticality and complexity of software.

Software engineering disciplines are less mature in comparison to other engineering disciplines; this, combined with the intangible nature of software, means clients do not always take the impact of software changes into consideration, assuming the software can be changed at any time with minimum impact, a concept known as the ‘impact of hidden complexity’. Glass [RD-5] estimated that software complexity increased by 100% for every 25% increase in related business.
1.2 WHAT IS THE PROBLEM?

If software complexity grows four times more quickly than business complexity [RD-5] and the cost of remedying an error increases with time and the point in its lifecycle that the software has reached [RD-4], we can assume that increased complexity will create more errors that cost more to fix. This affects development costs and schedules: it can cost as much as 100 times more to remedy an error that is detected late in the development lifecycle.

Two important concepts describe how errors occur during the development lifecycle of software: error propagation and error distribution. It makes sense to invest in methods that avoid the negative effects of error propagation and distribution and also contribute to early error detection and effective defect prevention. This paper presents some of these methods, which we believe can address such problems throughout the system development lifecycle, with positive impacts on schedule control, error reduction and cost containment.

| BUGS | STAGE FOUND |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Stage Introduced | Requirements | Coding/Unit Testing | Integration Testing | Beta Testing | Post Release | Total |
| Requirements | 5.0 | 8.0 | 2.3 | 0.2 | 0.2 | 15.6 |
| Coding/Unit Testing | - | 32.0 | 40.5 | 4.5 | 4.5 | 81.5 |
| Integration Testing | - | - | 2.3 | 0.4 | 0.4 | 3.0 |
| Total | 5.0 | 40.0 | 45.0 | 5.0 | 5.0 | 100 |
| Cost per bug | 1X | - | 10X | - | 100X | - |

Table 1 - Cost per bug through the lifecycle. [RD-4]

ERROR DISTRIBUTION

Typically, the number of errors uncovered in system development is higher in earlier phases of a project. This, together with the higher costs of removing errors discovered in later phases, suggests that without a strategy to prevent the discovery and propagation of errors in the initial phases of development, costs will increase and schedules become out of control at later stages.

ERROR PROPAGATION

If not corrected, errors discovered in early stages (e.g. requirements or design) will propagate and give rise to more errors in later stages (e.g. coding or implementation) of the development process. This leads to an exponential increase in both the total number of errors and the effort needed to remedy them. Error removal in the early stages, using error detection and defect prevention techniques, offers improvement and optimisation of software development processes.
1.3 ENGINEERING METHODOLOGIES

A system is considered “safety-critical” when its failure can cause death or injury to a human being and “mission-critical” when its failure can cause the failure of operations critical to the organisation’s mission, equipment or facilities. Safety-critical systems include for example those used to control aircraft or trains, railway signal systems, critical automotive systems (such as brakes or airbags) and nuclear reactor control systems. Examples of mission-critical systems include satellite systems, telecommunications and financial systems. To ensure high dependability and integrity, functional - and particularly non-functional -requirements, such as those related to safety, verification, validation, qualification and certification, require special attention.

When building a house or a bridge, regardless of the functions and requirements of the project, no one argues against the basic assumption: the structure should stay up. Thus, all the requirements (and costs) related to that assumption should be treated seriously, best practice be followed and regulations strictly adhered to. In some cases, this practice is not followed when developing software engineering projects with high integrity levels; high integrity requirements are typically not at the top of the list for discussion.

Despite efforts to standardise systems and software engineering practices, the systems and software production process is still very mechanical, difficult to reproduce and dependent on people, and is therefore prone to errors. To minimise failure, stringent systems and software development standards must be adhered to. Rigorous and dedicated processes, methods and techniques must be used throughout the system and software lifecycle to ensure dependability and safety functions. These include:

- **RAMS ANALYSIS** - Hazard analysis, fault tree analysis, failure modes effects and criticality analysis, common cause analysis, hardware-software interaction analysis
- **V&V** - Verification support at different phases of the lifecycle (requirements, high-level design, detailed design, code) and testing (unit, integration, system) using tools usually applied in the development of safety-mission-critical systems, including the application of fault injection, formal methods and safety/security assessment methods to verify robustness, error handling capabilities and system-level diagnostics
- **Qualification & Certification Support** - Definition of the processes, methods and certification needed; support for the analysis, selection and qualification of tools; production of safety cases and evidence for certifying authorities

![Figure 4 - Development flow of a safety-critical system focused on the methodologies and techniques that address the safety and dependability requirements](image-url)
To ensure unbiased, reliable and secure conduct of these activities, a level of independence from the development team is usually recommended. The use of such high integrity techniques adds value to systems and software development even when certification by a third-party is not mandatory. The most commonly-used techniques are based on V&V methods. These are important for detecting and removing existing errors and can be seen as a “reactive” activity, removing errors after they have been found in the system.

To ensure the high integrity of the system being developed, other, more “proactive” activities, (such as dependability, safety and RAMS analyses, and qualification), may be applied at earlier phases of the development, to prevent defects being introduced into the system.

1.4 APPLICABLE DOMAINS

The Safety-Critical Validation techniques described here have a direct application in safety-critical systems, and are often required by specific domain regulations. Users are normally concerned with the dependability and safety properties of their products or systems, especially if human safety or mission success can be adversely affected by the failure of the system in operation. Safety-Critical Validation techniques are now considered an essential part of core engineering activities and are used in:

- **Aerospace** - Avionics Control and Navigation
- **Space** - Satellites, Launch Vehicles and Ground and Mission Control Systems
- **Defence** - Embedded Computers and Communications Systems
- **Transportation** - Automotive and Railway

The application of Safety-Critical Validation services has expanded into other industries at the same time as critical, often embedded, systems have become core to their functions. Industries such as the nuclear, medical, banking/finance, semiconductor, telecommunications, energy and home technologies demand not only safety but also mission criticality.

The trend to use Safety-Critical Validation services is due to new regulations imposed by federal or regional authorities, such as the European regulations approved by the Committee on the Internal Market and Consumer Protection, which hold Original Equipment Manufacturers (OEM) responsible for software defects in their products. Techniques such as dependability, safety and RAMS analysis, certification and qualification support are being applied at early phases of development to try to detect defects before they are introduced into the system.

The cost of remedying an error increases with time and the software’s lifecycle stage. As shown in Table 1, applying and anticipating these techniques early in the development lifecycle can provide significant early error detection and have a dramatic impact on development costs and production time. The cost of error-remedying can be as much as 100 times greater in the final phases of the development lifecycle.
2. SAFETY AND VALIDATION METHODOLOGIES

As shown in Figure 4, development teams must guarantee the system’s ability to address the desired functionality, including safety. System development typically allocates functional requirements to hardware and software. These components have their own implementation and together create the system’s design. The combined system (hardware and software) must be assessed for safety and dependability, which introduces RAMS and V&V techniques into the development process.

Figure 5 shows the safety-critical validation activities that, to cover all dependability and safety requirements, must be executed throughout the lifecycle. As shown in Figure 5, the major Safety-Critical Validation methods are RAMS, V&V and Qualification & Certification. In the following sections we will look in more detail at the Safety-Critical Validation methods and their respective activities.

2.1 RAMS METHODOLOGIES

Reliability, Availability, Maintainability and Safety (RAMS) comprises the techniques and analysis used to assess the safety and dependability of a system. RAMS techniques, if applied early enough in the development lifecycle, can have a major impact on decisions related to the design of the sub-systems that contribute to a more dependable and safer system that can be designed and developed at a lower cost.

A set of RAMS-related assessment techniques can be applied during a project’s lifecycle. Depending on various aspects of the system (criticality, system requirements, etc.) different combinations of techniques are used. A major distinction among analyses can be made on the basis of the cause-effect relation of faults and failures within the system and between the hardware/software interactions with the overall system. These include:
RAMS analysis is an excellent engineering support activity, especially during the requirements and architecture phases. It provides useful inputs for requirements completeness and coherence, especially regarding safety and fail-safe issues. Moreover, RAMS is a very useful instrument for assessing the safety properties of system architecture. It also contributes to identifying the effects of failures on the overall system, by considering the possibilities and probabilities of errors. System architecture improvements, operation manuals and FAQs can also benefit from RAMS assessments to make the system safer.
DEPENDABILITY ANALYSIS

Dependability is a collective term used to describe availability performance and its influencing factors: reliability, maintainability and maintenance.

Dependability analysis typically starts by describing the system functionality, using a function analysis (FA) followed by a Criticality Analysis (CA) to determine system criticality. Reliability analysis focuses on ensuring the ability of a system or component to perform the required functions in defined conditions for a specified period of time. The most common analysis conducted is Reliability Block Diagrams (RBD).

Availability represents the percentage of time within the period, usually the system’s functional life, that a system is functional. Mathematically, availability is the Mean Time Between Failures (MTBF) divided by the sum of MTBF and Mean Time To Repair (MTTR).

Maintainability is defined as those characteristics of design and installation that determine the probability that equipment, machines or systems that have failed can be restored to normal operation within a given time, using prescribed practices and procedures. It is sometimes sub-divided into serviceability – ease of conducting scheduled inspections and servicing and reparability – ease of restoring service after a failure.

Other topics, such as security, supportability, testability and operability can be covered by the dependability programme.

SAFETY ANALYSIS

Safety is generally considered to be “freedom from unacceptable risk of harm”. It is also defined as a state in which, with respect to the following factors, an acceptable level of risk is not exceeded:

- Injury or occupational illness
- Damage to launcher hardware or launch site facilities
- Damage to an element of an interfacing manned flight system
- The main functions of a flight system itself
- Pollution of the environment, atmosphere or outer space
- Damage to public or private property

“There is no such thing as zero risk.” [RD-3]. This offers some justification of the need for safety analysis in projects that, in some way, may jeopardise human life or have a considerable financial or environmental impact. Unfortunately, life has provided us with evidence for this; all systems, including the most demanding safety-critical systems such as aircraft, trains, cars and nuclear power plants, are susceptible to failure.

Safety analysis is integrated within a safety programme and usually encompasses a set of activities that identify risks and causes for system failures which, in either a single or chained set of events, may lead to system breakdown. The system components’ criticality levels are one of the outputs of the analysis, and will affect both the way the system is developed (lifecycle processes) and the way the system is validated.
Because of these effects, safety analysis includes a set of activities that must be considered as early as possible in the system development lifecycle. Avoiding these activities at the start is known to increase project costs in later stages, as systems must be forced to reach the safety levels required. Safety is very often related to certification and qualification.

The type of analysis to be performed, and its depth, varies with the system and domain but Hazard Analysis (HA) is the most common.

**BENEFITS AND COSTS OF RAMS SERVICES**

RAMS methodologies can offer advantages such as:

- Revealing expected system failures caused by hardware/software or use errors
- Contributing to an overall increase of confidence at system level, for example through the systematic use of techniques such as FMEA at the software/hardware level
- Supporting design decisions and/or justifications on the basis of the results of RAMS analysis
- Focussing verification and validation in the most critical system or sub-system hardware or software components
- Supporting upgrades or re-qualification of systems under development or in production on the basis of the results of RAMS analysis

**2.2 V&V SERVICES**

Verification and Validation (V&V) activities involve system, hardware and software development and are conducted throughout the system development lifecycle, from planning to acceptance. These activities guarantee that the product or system is free from faults and performs according to its system/hardware/software specifications. V&V also guarantees that the results from safety and dependability analyses are not flawed due to incorrect implementation and that the system is doing what it was designed to do.

Verification and Validation (V&V) activities can be applied at all levels of a system, in either hardware or software. CRITICAL Software has the experience to create V&V plans for every phase of the development lifecycle and to assess the most efficient strategies (plans, methods, tools and strategies, risk assessment and management solutions) to meet the customer’s needs and deadlines.

Verification consists of ensuring that the system complies with requirements through all phases of the lifecycle. This is accomplished by analysis, inspections, and formal evaluations of intermediate and final system artefacts. The artefacts to be analysed are selected according to previously performed criticality analysis, increasing the activity’s “value for money”. Validation demonstrates that the system accomplishes its intended purpose. It is achieved by testing the product in either real or simulated environments.
The purpose of validation is to assess error-handling behaviours, safety and security issues and areas where a failure might cause undesirable effects. Validation can also be performed using a set of techniques that will test the system from different angles to ensure it is fit for its intended use.

Examples of these techniques are Fault Injection, Formal Methods and Security Assessment.

### FAULT INJECTION

Included within verification and validation, fault injection capabilities are essential to verify robustness, error-handling capability and system level diagnostics. Manual fault injection (lifting and shorting pins, destroying hardware, and so on) is costly and time-consuming and moreover, only a small set of faults can be created.

Fault injection is considered a valuable complement in nominal tests of system and to RAMS analysis. Fault injection on the tests is performed to:

- Verify that the system under test can perform with minimal disruption during a hardware fault
- Verify that correct error reporting processes and the correct recovery actions for that error occur
- Verify the capability of the diagnostics software / implemented architecture to identify and isolate hardware and software defects

### FORMAL METHODS VERIFICATION SERVICES

Formal verification uses mathematical techniques to ensure that a design conforms to a notion of functionality. To achieve verification, formal methods, namely a set of mathematically rigorous techniques and tools for the specification, design and verification of software and hardware systems, are used [RD-6].

To provide formal proofing, having as input the designed models, a set of tools and techniques that operate on the models for extracting formal evidence of correctness (or incorrectness) of the model can be used, such as:
All types of formal verification tool sets (including design modelling tools, formal proofing tools and model checking tools) are the subject of research and development.

SECURITY ASSESSMENTS

For safety-critical systems to operate safely, they must be secure. As technologies advance, and networking and computer systems become more accessible and their use more widespread, security becomes a concern for most organisations.

CRITICAL Software has defined an optimum methodology in which the system can either be addressed as a whole or as individual components. Our methodology includes techniques to address:

- **System Security assessments** - independent assessment which can take place at the application or infrastructure level, and take either an architectural, technological or process perspective

- **Security testing** - this focuses on testing the system from an intrusion perspective, with the objective of discovering vulnerabilities. They can be executed as blackbox, graybox or whitebox, depending on the level of knowledge of the system being tested

- **Vulnerability impact analysis** - this concentrates on analysing possible vulnerabilities from a broad perspective and at a high (architectural) level, with the objective of identifying the overall system impact of a single vulnerability
INDEPENDENT VERIFICATION AND VALIDATION (IVV)

IVV follows the same objectives as V&V. It is designed to improve the quality and reduce the costs of a system by detecting problems early in the development lifecycle. The difference between regular V&V and IVV activities relates to the independence of the team that develops the system and the team that performs V&V.

The outcomes of applying IVV to a system are diverse and can contribute to the project’s success in multiple ways. While increasing the system’s dependability, the recommendations provided by an IVV team allow the simplification of system and software architectures, resulting in increased maintainability and usability and higher reuse capability, which facilitates future developments, provides greater independence from providers and supports better knowledge and operation of the system. IVV results can also be used at the program management level to reduce development risks, as verification and validation are performed by an organisation independent of the system or software developer, providing information to support decision-making. Importantly, IVV activities are not a substitute for, but a complement to, V&V activities performed by the development team. Whenever possible, IVV uses alternative methods, tools and more non-nominal test cases, to increase the “added value” of its results.

Figure 10 - IVV activities and the typical development lifecycle.

BENEFITS AND COSTS OF VERIFICATION AND VALIDATION METHODOLOGIES

Studies indicate that the return on investment (ROI) of introducing V&V and IVV activities earlier in the lifecycle leads to efficiency increases of up to 75% during testing activities, which compensates for the costs of IVV.

This ROI estimate does not take into consideration other important and positive factors of V&V activities, such as resource allocation or project schedule constraints, minimising unexpected project delays, reducing the need for extra resources and offering improvements in design and quality.
2.3 CERTIFICATION AND QUALIFICATION SERVICES

Certification is the confirmation that a fact or statement is true and supported by documentary evidence, which increases confidence in a product or system. Certification is usually a combination of a strict V&V programme and a RAMS programme (Safety and Dependability analysis), guided by a set of mandatory standards. During the programme, the lifecycle evidence is audited by an external regulatory organisation legally empowered to approve or withhold permission for the system’s deployment and use.

Achieving an unbiased certification usually involves a certification authority and is mandatory in certain domains, including aviation, nuclear, railways and medicine. It typically involves:

- A process or set of guidelines that specify which objectives must be met to achieve certification
- Collection of artefacts in a safety case used as evidence/records/traceability that the objectives were achieved
- Providing artefact visibility to a certification authority through a liaison process during the whole life cycle of the product

Qualification is the formal process of ensuring the suitability of a given product or service for its specific context and application. The context and environmental conditions provide the requirements for achieving qualification. Qualification can be considered as a set of objectives that must be met before the product can be considered fit for purpose, according to a specific standard or guidelines. In some cases, a qualification authority is involved.

Qualification can be seen as a subset of a certification programme. Although it follows a set of standards and requirements, it is usually limited to an enhanced V&V programme that covers non-functional system requirements such as testability or maintainability.

Certification and Qualification services vary depending on the domain, as different standards apply in different domains, such as RTCA/DO-178B/C, RTCA/DO-254 or RTCA/DO-200a for airborne systems, CELENEC EN50126 for railway systems, ISO26262 for automotive systems and ISO62304 for medical devices.
3. FINAL THOUGHTS

Services such as the Safety-Critical Validation provided by CRITICAL Software are now both essential and required for safety/mission-critical applications in much of modern industry. These services support higher levels of confidence in the dependability and quality of the systems being deployed, while offering added value in development, reduction of risks, simplification of architecture and improvement in maintainability.

Safety-Critical Validation services also provide substantial financial benefits. Studies show that applying a strict verification and validation plan, as well as RAMS, early in a project leads to overall savings, largely compensating for the cost of performing these activities.

Despite some differences and domain specificities, Safety-Critical Validation activities are performed in almost all domains, whenever safety/mission-critical systems are at stake. CRITICAL Software believes our extensive experience in domains such as aerospace, space, energy, defence and transport gives us the technical knowledge to offer added value to our customers. Good practice and experience in one domain supports good practice in others.

The technical competence acquired in projects in a variety of domains means CRITICAL Software can offer:

- Knowledge to adapt and adjust the tasks to be performed according to customer’s needs
- High technical expertise in validation activities and embedded development for safety-critical systems
- Ease of performing both on-site and off-site activities
- Easy integration with customers’ teams, an international culture and excellent customer relationships.

Our flexibility is not only a major differentiator, it is key to our customers’ experience and demonstrates that CRITICAL Software positions itself as a partner in V&V and RAMS services, rather than an auditor. CRITICAL Software’s mindset has attracted regular customers who rely on us to support them in their development activities.
4. ABOUT CRITICAL SOFTWARE

The completeness and the depth of the results produced from the analysis performed in Safety-Critical Validation methods is highly dependent on the experience of the organisation providing the service. Knowledge of the domain, techniques, methods and tools are key. Enabling the automation of tasks and offering insights into the fulfilment of specific requirements (e.g. compliance with coding standards) are also important in achieving high-quality results at affordable costs.

CRITICAL Software has proven experience in delivering dependable mission-oriented critical solutions for the Aerospace, Space, Defence, Automotive, Railway and Energy industries and for high-profile customers including NASA, ESA, AgustaWestland, EADS, BAE Systems and Thales Alenia Space, among many others. CRITICAL Software has experience of hands-on cross-domain engineering experience in all the methods and techniques presented in this paper, which has been demonstrated in several successful projects. Having a complete view of safety-critical lifecycle development allows CRITICAL Software to provide validation services for a comprehensive set of activities, from initial RAMS analysis to Verification, Validation, formal Methods Verification and supporting Certification and Qualification processes.

CRITICAL Software’s participation in international research and development projects brings the academic and industry knowledge needed to improve and adapt existing techniques as applied to safety-critical systems. CRITICAL Software are also prepared to be innovative when unusual solutions are needed, with efficiency and cost and time reduction being key driving factors for our company. CRITICAL Software addresses these issues by using market-standard safety validation tools that best fit the validation activities but we have also developed in-house tools to support safety-critical validation services that allow increasingly efficient results, while maintaining costs at an affordable level.
5. REFERENCES

[RD-1] - Software Defect Prevention - In a Nutshell, SixSigma
[RD-2] - Defect Prevention: Reducing Costs and Enhancing Quality, SixSigma
[RD-8] - Girish Palshikar An introduction to model checking, February 2004