Development of Centre for Safety Critical Software for Signalling Applications

- Initiative by IRISET/Secunderabad

Shri.C.K.Prasad, Professor Telecom/IRISET

1.0 Introduction

A safety-critical system is a system whose failure or malfunction may result in one (or more) of the following outcomes:

- death or serious injury to people
- loss or severe damage to equipment/property
- environmental harm

Risks of this sort are usually managed with the methods and tools of safety engineering. A life-critical system is designed to lose less than one life per billion (10E9) hours of operation. Typical design methods include probabilistic risk assessment, a method that combines failure mode and effects analysis (FMEA) with fault tree analysis. Safety-critical systems are increasingly computer-based.

2.0 Reliability Regimes

Several reliability regimes for safety-critical systems are used specific to application area

- **Fail-operational systems** continue to operate when their control systems fail. Examples of these include elevators, the gas thermostats in most home furnaces, and passively safe nuclear reactors. Fail-operational mode is sometimes unsafe. Nuclear weapons launch-on-loss-of-communications was rejected as a control system for the U.S. nuclear forces because it is fail-operational: a loss of communications would cause launch, so this mode of operation was considered too risky.
- **Fail-safe systems** become safe when they cannot operate. Railway signaling systems and sub-systems are design based on this principle. For example: Track circuits, Axle counters etc are fail-safe because if they fail, the signal to the corresponding route is put to "ON" the most restrictive aspect. Many medical systems also fall into this category. For example, an infusion pump can fail, and as long as it alerts the nurse and ceases pumping, it will not threaten the loss of life because its safety interval is long enough to permit a human response. Famously, nuclear weapon systems that launch-on-command are fail-safe, because if the communications systems fail, launch cannot be commanded. Railway signaling is designed to be fail-safe.
- **Fail-secure systems** maintain maximum security when they cannot operate. For example, while fail-safe electronic doors unlock during power failures, fail-secure ones will lock, keeping an area secure.
- **Fail-Passive systems** continue to operate in the event of a system failure. An example includes an aircraft autopilot. In the event of a failure, the aircraft would remain in a controllable state and allow the pilot to take over and complete the journey and perform a safe landing.
- **Fault-tolerant systems** avoid service failure when faults are introduced to the system. An example may include Electronic Interlocking system in hot standby configuration with power supply and VDU all duplicated. The normal method to tolerate faults is to have several computers continually test the parts of a system, and switch on hot spares for failing subsystems. As long as faulty subsystems are replaced or repaired at normal maintenance intervals, these systems are considered safe. Interestingly, the PES(Programmable Electronics System)/computers, power supplies and control terminals used by operator must all be duplicated in these systems in some fashion.

3.0 Software engineering for safety-critical systems

Software engineering for safety-critical systems is particularly difficult. There are three aspects which can be applied to aid the engineering software for life-critical systems. First is process engineering and management.

Secondly, selecting the appropriate tools and environment for the system. This allows the system developer to effectively test the system by emulation and observe its effectiveness. Thirdly, address any legal and regulatory requirements, such as General Rules, CENELEC/ RDSO & CRS for Indian Railways, FAA requirements for aviation etc. By setting a standard for which a system is required to be developed under, it forces the designers to stick to the requirements. The avionics industry has succeeded in producing standard methods for producing life-critical avionics software. Similar standards exist for automotive (ISO 26262), Medical (IEC 62304) and nuclear (IEC 61513) industries. Rail Industry standards are based on CENELEC EN 50128 FOR . The standard approach is to carefully code, inspect, document, test, verify and analyze the system. Another approach is to certify a production system, a compiler, and then generate the system's code from specifications. Another approach uses formal methods to generate proofs that the code meets requirements. All of these approaches improve the software quality in safety-critical systems by testing or eliminating manual steps in the development process, because people make mistakes, and these mistakes are the most common cause of potential life-threatening errors.

Safety Critical Systems in Rail Transportation

Rail Transportation systems use electronics for subsystems previously controlled mechanically or manually. As a result, rail systems deliver far greater flexibility and are capable of real-time adjustments for speed, route, and passenger comfort. While these new electronic control and monitoring systems present many benefits to assure their safe operation, regulations mandate that such systems comply with industry standards for hardware and software development and are thoroughly tested and documented.

In rail transportation, as more electronic systems come into play, it becomes necessary to do whatever is possible to assure correct operation of these advanced systems.

Software used in safety-critical systems is, of course, a key element in the correctness of the system's operation. Most commonly, this software consists of an application running on top of an operating system.

IEC 61508

Because it's critical that electronic systems operate safely various government-sponsored agencies and independent technical standards organizations have become involved in defining regulations for safety-critical systems. The International Electro technical Commission (IEC), a worldwide organization for standardization, promotes international co-operation on all questions concerning standardization in the electrical and electronic fields. IEC- 61508, the international standard for electrical, electronic and programmable electronic safety-related systems, sets out the requirements for ensuring that systems are designed, implemented, operated and maintained to provide the required safety integrity level (SIL).

In Europe, CENELEC—the European Committee for Electro technical Standardization—governs European railway standards, and these standards are beginning to make their way into the North American railway and public transport market as well. The CENELEC standards EN 50126, EN 50128 and EN 50129 are typically applied to define appropriate safety analysis for such systems:

EN 50126 deals with Reliability, Availability, Maintainability and Safety for the entire railway system.

EN 50129 applies to safety-related electronic control and protection systems.

EN 50128 applies to safety-related software for railway control and protection systems.

IEC standards are applied at various "Safety Integrity Levels" (SIL), representing varying degrees of criticality based on the system's use. IEC EN 61508 outlines the toleration of a probability for failure at each level with the most critical aspects of the system (i.e., SIL 4) having the least tolerance for failure. As detailed in Figure-2, the standard defines both a system's PFD (Probability of Failure on Demand) and RRF (Risk Reduction Factor) for each SIL level. The term Probability of Failure on Demand (PFD) means the likelihood that the system will fail when asked to perform a particular operation for which it is designed. The Risk Reduction Factor (RRF) is the amount of risk that can be reduced by implementing the corresponding SIL system.

SIL	PFD	RRF
1	0.1-0.01	10-100
2	0.01-0.001	100-1000
3	0.001-0.0001	1000-10,000
4	0.0001-0.00001	10,000-100,000

Figure-2

PFD and RRF for SIL Levels

Common Regulatory Elements

All international safety-critical software standards incorporate common elements that apply to all software systems, regardless of their end application. While the different standards have their own particular phraseology and individual features, they all generally require that software is developed according to a well-documented plan, and that its operation is consistent with the plan.

In particular, safety-critical software must demonstrate, through rigorous testing and documentation that it is well designed and operates safely.

- **Process:** The process through which the software was designed, developed, and tested must be fully described and shown to be consistent. This is broken down into several sub categories:
  - **Planning** - the objectives of the system are stated, along with the plan for achieving and verifying that these objectives have been achieved.
  - **Requirements -** the functional requirements of the system are explicitly identified and correlated with the system capabilities.
  - **Design** - the system design is specified, including hardware and software, with theory of operation and other aspects of design that enable examiners to understand how the system intends to achieve its objectives.
  - **Development** - the development process, including the tools used, code reviews, test plan, documentation, and staff training.
  - **Verification** - the process of assuring that the system performs in accordance with the requirements or specifications, and that it achieves its functional objectives. (The product is built the right way)
  - **Validation** – the process of assuring that the right product is being built.
  - **Configuration management** - control of incremental revisions over time, enabling reproducibility of results and protecting against the introduction of faults that cannot be backed out.
  - **Quality assurance** - processes and procedures that assure that the system has been developed and produced in accordance with its goals, and that it delivers the capabilities it is intended to provide. Techniques and methodologies need to defined with mutual agreement between the product development organization and certifiying agency to ensure the quality of the product.
- **Code** refers to the source code produced by the developers or development tool and it includes all system and application source code, test code, scripts, and object code. This code is to be reviewed as part of the regulatory compliance process, and must agree with the actual code used in the system.
- **Test** includes specific tests performed to verify the correct operation of the code, as well as its ability to achieve all design goals and system requirements. Testing includes Structural coverage analysis to insure that all program instructions are tested. Finally, unit/white-box, integration/black-box, and final acceptance testing generally are included.
- **Results** consist of complete results of all tests compiled into a unit and integration test report.

Manufacturers of rail transportation systems have development teams that are fully capable of generating the documentation required to comply with these safety-critical standards, as required by applicable regulations and have done so for years.

However, there are some aspects of this work that developers would like to avoid or that pose challenges. As safety-critical systems evolve in complexity and make use of more powerful microprocessors, they increasingly employ commercial RTOS technology. The real-time operating system controls and manages the application software to maximize system resources for a given processor.

4.0 Software reliability

Software reliability is a special aspect of reliability engineering. System reliability, by definition, includes all parts of the system,

including hardware, software, supporting infrastructure (including critical external interfaces), operators and procedures. Traditionally, reliability engineering focuses on critical hardware parts of the system. Since the widespread use of digital integrated circuit technology, software has become an increasingly critical part of most electronics and, hence, nearly all present day systems.

There are significant differences, however, in how software and hardware behave. Most hardware unreliability is the result of a component or material failure that results in the system not performing its intended function. Repairing or replacing the hardware component restores the system to its original operating state. However, software does not fail in the same sense that hardware fails. Instead, software unreliability is the result of unanticipated results of software operations. Even relatively small software programs can have astronomically large combinations of inputs and states that are infeasible to exhaustively test. Restoring software to its original state only works until the same combination of inputs and states results in the same unintended result. Software reliability engineering must take this into account.

Despite this difference in the source of failure between software and hardware, several software reliability models based on statistics have been proposed to quantify what we experience with software: the longer software is run, the higher the probability that it will eventually be used in an untested manner and exhibit a latent defect that results in a failure.

As with hardware, software reliability depends on good requirements, design and implementation. Software reliability engineering relies heavily on a disciplined software engineering process to anticipate and design against unintended consequences. There is more overlap between software quality engineering and software reliability engineering than between hardware quality and reliability. A good software development plan is a key aspect of the software reliability program. The software development plan describes the design and coding standards, peer reviews, unit tests, configuration management, software metrics and software models to be used during software development.

A common reliability metric is the number of software faults, usually expressed as faults per thousand lines of code. This metric, along with software execution time, is key to most software reliability models and estimates. The theory is that the software reliability increases as the number of faults (or fault density) decreases or goes down. Establishing a direct connection between fault density and mean-time-between-failure is difficult, however, because of the way software faults are distributed in the code, their severity, and the probability of the combination of inputs necessary to encounter the fault. Nevertheless, fault density serves as a useful indicator for the reliability engineer. Other software metrics, such as complexity, are also used. This metric remains controversial, since changes in software development and verification practices can have dramatic impact on overall defect rates.

Testing is even more important for software than hardware. Even the best software development process results in some software faults that are nearly undetectable until tested. As with hardware, software is tested at several levels, starting with individual units, through integration and full-up system testing depending on the Safety Integrity levels. Unlike hardware, it is inadvisable to skip levels of software testing. During all phases of testing, software faults are discovered, corrected, and re-tested. Reliability estimates are updated based on the fault density and other metrics. At a system level, mean-time-between-failure data can be collected and used to estimate reliability. Unlike hardware, performing exactly the same test on exactly the same software configuration does not provide increased statistical confidence. Instead, software reliability uses different metrics, such as code coverage.

Eventually, the software is integrated with the hardware in the top-level system, and software reliability is subsumed by system reliability. The Software Engineering Institute's capability maturity model is a common means of assessing the overall software development process for reliability and quality purposes.

5.0 Initiation of Proposal to Signing of the Consultancy Agreement for setting up of Centre for Safety Critical Software for Signaling Applications

IRISET proposed to set up a Centre for Safety Critical Software for Signaling Applications which shall provide advance training and a structured platform for knowledge transfer of key

technologies in various signaling systems on Indian Railways. The signaling system on the railways for the safety critical application has undergone major transformation. The present technology system and sub-systems are based on mission critical hardware with embedded safety critical software specific to rail industry.

The Electronic Interlocking(EI) installations on Indian Railway are from multiple OEM's, same is the case with other software embedded signaling system like MSDAC, TPWS etc. The executive software, application development software suite and simulator software system are propriety of the OEM's. In EI itself there are multiple OEM systems from Ansaldo STS, Invensys Rail Systems, GE transportation, AZD Praha, Medha Systems, Siemens etc. installed on various Zonal Railways. Hence maintaining these systems require training on complete suite of application development software, testing software, validation software and simulator including version updates to be procured, deployed, maintained and supported during its life cycle.

The effective working of the centre requires knowledge transfer, manpower training and development of competency in the software systems and sub-systems. The OEM's are wary of transferring the complete technology and technical knowledge of the software systems due to Intellectual Property Rights (IPR's) issues.

The safety critical centre proposed to be set up will bridge the gap which exists in the railways in terms of safety critical software knowledge, processes, competency etc, and will develop the human resource and make them competent to operate and maintain these systems which will help in enhanced reliability, availability, maintainability and safety of the software based signaling systems. These systems have direct bearing on line capacity, smooth running of train services and higher level of safety in train operations.

IRISET signed the Consultancy Agreement with Indian Institute of Technology Kharagpur for Setting of the Centre for Safety Critical Software for Signalling Applications at IRISET, Secunderabad on 28th April 2015.

6.0 Scope of Consultancy:

6.1 The Consultancy covers the following aspects.

a) Identify and document the qualifications and training requirements of the Signalling & Telecommunication personnel to be posted at the Centre.

b) Work out detailed syllabus for imparting training in Software Engineering, Reliability Engineering, Safety Standards including CENELEC, Software Testing, Verification and Validation, Version Control, Safety Cases, etc.

c) Specification and Configuration of Safety Critical Data Centre. This includes specifications for servers, hypervisor, operating system, rack, storage, security infrastructure, bandwidth, networking hardware, work station etc.

d) Suggest Software Verification tools to be procured and set up at the Centre.

7.0 Methodology and Roadmap- Jointly agreed by IRISET & RDSO for development of the Centre for Safety Critical Software

7.1 Scope of work

The broad scope of the work is as under:

7.1.1 Identify and document the qualifications and training requirement of Signalling & Telecommunication personnel to be posted at Centre.

7.1.2 Work out detailed syllabus for imparting training in Software Engineering, Reliability Engineering, and Safety Standards including CENELEC, Software Testing, Verification and Validation, Version Control. Safety cases etc.

7.1.3 Specification and Configuration of Safety Critical Data Centre. Including specifications for networking hardware, work station etc and execution of the work.

7.1.4 Software Application, Data preparation, Simulation, Verification, Validation and testing tools to be procured and set up at the Centre in consultation with RDSO, carrying out preparation of application data for all kinds of modern electronic equipment for requirement of zonal railways, along with modifications in present working installation using the tools as well as at manual level. The tools procured will be utilised for fulfilling the training needs of S&T personnel.

7.1.5 Validation and Verification of Application Data at Lab level before issue for implementation and verification by zonal railways.

7.1.6 Study in detail the relevant reliability issues for modern electronic equipment and provide inputs to RDSO for identification of solution and impact on implementation.

7.1.7 Create, Maintain and update database of all electronic equipment in complete detail, including failure position by coordinating with zonal railways and carry out quarterly analysis to find the trends and identify relevant issues and propose solutions in consultation with RDSO.

7.1.8 Roll out the network connectivity to all stakeholders, monitor, maintain and update the relevant items on network for use by all stakeholders.

8.0 Role of RDSO:

8.1 IRISET in consultation with RDSO shall finalise on the infrastructure to be developed along with the software and hardware tools to be procured by IRISET.

8.2 RDSO shall collaborate with IRISET in developing the Centre for Safety Critical Software and its operationalisation.

8.3 RDSO shall lay down the process, provide all necessary technical assistance and knowledge sharing to IRISET, who will be analysing the defects/errors in the software based signalling systems/sub-systems from various manufacturers installed in the field and providing necessary technical inputs to the OEM/Signalling Vendor/ Zonal Railways as the case may be. This will be extremely useful in monitoring and enhancing the RAMS(Reliability Availability Maintainability & Safety) of the system/ sub-system.

8.4 It is understood that RDSO is executing a project on formalization of Interlocking rules and safety assessment in circuit design for yard layouts. The system being developed by RDSO can be utilized on reciprocal basis by IRISET for training and imparting knowledge to the S&T personnel from the Zonal Railways. The formal verification tool for modeling simulation and validation of interlocking requirements being as envisaged in the RDSO proposal shall be made accessible over a network infrastructure so that it can be gainfully utilized by all the Zonal Railways and IRISET.

8.5 Development of new technology in Signalling and train control systems, its deployment and absorption of technology is a dynamic and continuous process and hence RDSO shall frame the methodology and extend its support for continuously augmenting the requirement of the proposed software centre

9.0 Role of IRISET:

9.1 IRISET will frame the necessary bid documents in consultation with RDSO and based on the consultant's report.

9.2 IRISET to carry out, Procurement of Tools, Training Material and training to staff for software centre by OEMs, in collaboration with RDSO, including development of knowledge for design and verification of Application Data.

9.3 IRISET shall setup a software Verification and Validation process with necessary tools and technology with multi-site licence as part of the project for evaluation of Signalling and Interlocking software in collaboration with RDSO.

9.4 IRISET will execute the work for development of centre for Safety Critical Software in consultation with RDSO.

9.5 IRISET will train the design staff from the zonal railways in phased manner on various software driven systems / sub systems deployed / approved for deployment on the railways to develop application logic and data preparation, however such time till confidence level is generated in Zonal Railways, IRISET shall carry out preparation of application data, modifications in application data and verification and validation of the same, staff for same may be deputed to IRISET by zonal railways for this purpose.

9.6 IRISET will train the project executing organization/field units in a phased manner on testing and verification of the application before installation and commissioning of the Signalling system/sub-system.

9.7 IRISET will train the maintenance Organisation personals on testing & verification of the application, fault identification, rectification and debugging of the application software pertaining to the Signalling system/sub-system.

9.8 The software & simulation tools for developing/modification of the application logic for each of the systems/sub-systems of various OEM/Vendors will be procured/ acquired from the Zonal Railways, as the

case may be, and installed in the application server at IRISET.

9.9 The centre will develop competency under guidance of RDSO in analyzing the error logs of critical failures of Software based Signalling and Train Control systems installed over Indian Railways and suggest remedial action by taking inputs from the Zonal Railways.

9.10 The infrastructure so developed at IRISET will be provided over a private cloud architecture wherein all the simulation tools and data preparation applications for interlocking and yard layout will be hosted and made accessible to the zonal headquarters, RDSO and the field units over a secure network infrastructure.

9.11 The inventory of all the software applications, simulation tools and verification and validation software will be maintained and managed at the data proposed data centre. The activity like version control, software updates and other software maintenance activities will be managed from the centre.

9.12 The software's for applications logic for existing systems / sub-systems and new systems approved by RDSO will be hosted in IRISET and can be accessed by the Zonal HQ's over the MPLS VPN Network in a secured environment.

10.0 Role of Zonal Railways:

10.0 Identify and share the list of S&T personnel's for training on Software systems.

10.1 Zonal Railway will provide the initial data of all the software based signalling systems to IRISET along with version of the existing configuration and the yard / layout data version of the systems.

10.2 Provide a copy of the application data, Data preparation tool and simulator for the systems installed/ in pipe-line to IRISET.

10.3 Provide copy of the relevant documentation available with them.

11.0 Benefits:

11.1 The project is oriented towards development of competency in the area of Application software, Data preparation and testing by simulation of field functions for safety critical signaling and train control systems by training and skilling the design office team in the Zonall Railways,Divisional units, as well as in the construction and project units. The design team in the Headquarters will be trained in the development of the applications program and data preparation for various Signalling and Train Control systems and sub systems deployed and/or approved for deployment on Indian Railways. Prominent among them are EI, MSDAC, SSDAC, AWS, TPWS, TCAS, ETCS etc.

11.2 The software centre will get involved right from the beginning at the stage of type approval / cross acceptance of the system/technology for deployment on Indian Railways. IRISET will work with RDSO and OEM/vendor community to adopt the best practices to standardize the process of developing the application logic, data preparation, verification, validation, testing and troubleshooting procedures.

11.3 The project will help in creating Subject matter experts (SME) in the zonal headquarters for design, development & Verification of application logic based on SIP, TOC etc for the software based systems & sub systems.

11.4 The center will aid in developing testing, verification and trouble-shooting capability with in the field units who will be directly executing the work on ground and maintaining the systems.

11.5 The application for the yard can be pretested by the simulation/data validation tool or compiler etc by the design/testing team before it is deployed in the Interlocking system in the field ex: before loading and burning in the EPROM.

12.0 Organisation & Human Resource

12.1 It is envisaged that to keep pace with fast growing developments in the fields of software engineering the selection criteria for S&T engineers to be posted at the proposed centre shall include minimum matching skill set of a Graduate Engineer with Bachelor's degree in Computer Science / Computer Engineering / Software Engineering with domain knowledge of Signalling. If needed some officers can also be sponsored for M. Tech / Ph. D programs in Computer Science / Engineering / Software Engineering preferably with some project on assessment and validation of safety critical software.

12.2 It is proposed that the centre may be headed by an officer in SAG who in turn shall be assisted by an officer in JAG/SG and team of 12JS/SS officers,12 SE/SSE and 2 JE(IT), out of this proposed strength 4 officers (JS/SS) and 4 Supervisors shall be allocated to RDSO/Lucknow on Permanent basis for coordination, collaboration and adoption of new technologies and further dissemination of know-how from RDSO to IRISET. A separate cadre will be required to be created for supervisors at IRISET/RDSO to maintain continuity in future, the cadre posted in this category shall remain devoted permanently.

12.3 The tenure of officers selected and posted in this centre shall be minimum ten years depending upon the requirement. There is need to protect the career growth and promotion prospects of the officers posted in the centre. If need be by upgrading the post itself.

14.0 Proposed stage-wise implementation:

Stage 1:

i) Engaging consultants to provide a comprehensive approach and deliverables as per the scope of work.

ii) Training of the identified S&T personnel on safety standards for railway signaling.

iii) Creation of posts, identification of personnel and their postings in IRISET and RDSO.

Stage 2:

i) Setting up of a data centre with private cloud platform connected to zonal and divisional headquarters over secure MPLS, VPN network.

ii) Procurement of simulation tools and application/data preparation tool for EI, MSDAC, SSDAC, TPWS etc. from the OEMs which shall be hosted over the private cloud platform. (for the new systems for which RDSO may consider for giving type approval / cross acceptance the OEM shall offer the simulation tool and application development tool in advance to the Centre for Safety Critical Software and give a Crash course to the concerned S&T personnel. This will help to seamlessly integrate the new interlocking / Train Control system with the existing systems and streamline the adoption of new technology /systems on the railways.

iii) Identification and nomination of S&T personnel involved in design and execution of work connected with EI, MSDAC, SSDAC, TPWS etc.

iv) Training and Skill development of the design team in zonal headquarters and the execution team in the field as mentioned in the SI no. 3 & 4.

v) Training of the IRISET faculty and instructors on the data centre technology, deployment of application and simulation tools over the private cloud architecture, inventory management and maintenance of software.

Stage 3:

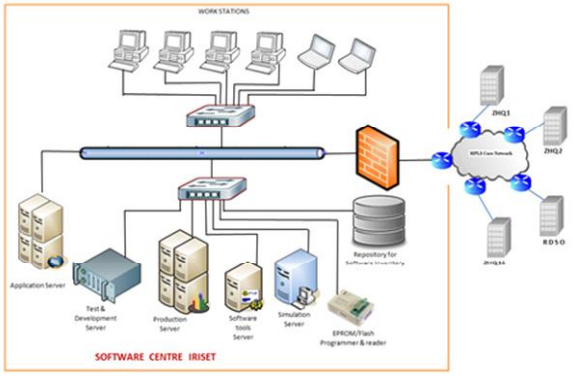
i) Training of the identified S&T personnel for RDSO and Zonal Railways on software engineering practices, reliability engineering, software language like C / C++ and areas given by the consultant.

ii) Development and deployment of software verification and validation tool being developed by RDSO.

iii) Training of identified S&T personnel on the software verification and validation tool.

iv) Post completion of stage 3 regular courses will be held by IRISET.

v) As required by Zonal Railways, support for Data Preparation shall be provided by IRISET.



CONCEPTUAL ARCHITECTURE OF SOFTWARE CENTRE @ IRISET WITH ACCESS TO ZONAL RAILWAYS, RDSO, CORE