Embracing our digital future

Railway system technology & software

Brian Tomlinson
Chief Systems Engineer
Network Rail

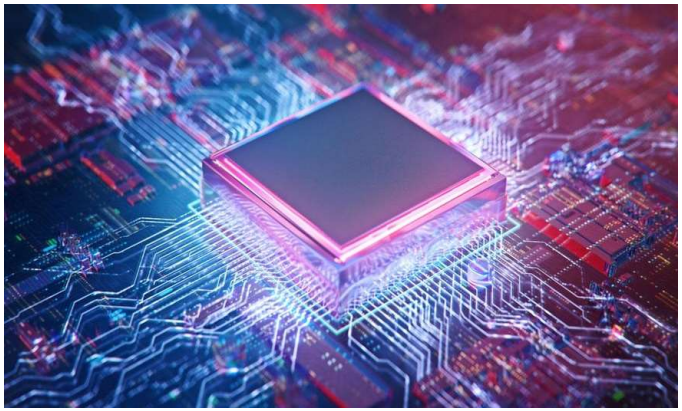
Presentation overview

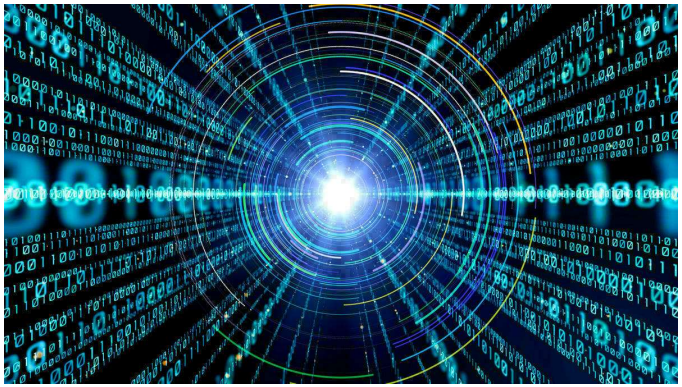
- Our use of technology and software - opportunities and challenges
- Potential failure modes, consequences and controls
- Improving digital competency and case studies
- Role of the client on projects involving high integrity software-based systems

Increasing use of technology

Our need for technology

- Can perform complex operations very quickly
- Removes/reduces hardware e.g. weight, size, cost, carbon
- Offers improved performance
- Provides data capture, storage and retrieval
- To meet customer specific needs
- Offers new opportunities/capabilities
- Can interconnect/interface with other things
- Adaptable during development and design
- Updates can be deployed at a later stage
- Elements can be re-used and improved



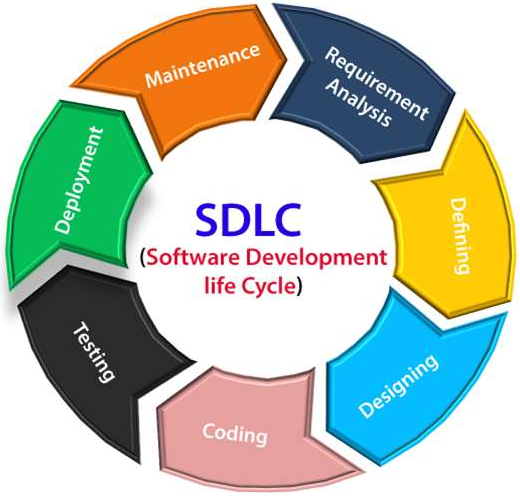


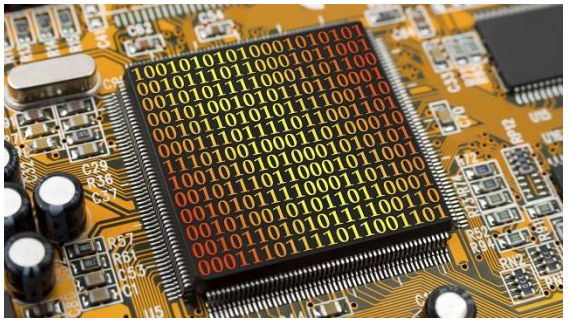
The future

- Faster processing
- Greater storage capability
- Greater interconnectivity
- Improved functionality
- Artificial intelligence
- Improved performance

- Increased complexity
- More interfaces
- System and data security
- Error potential
- Imported/exported risk
- Obsolescence

Software asset management and lifecycle





- Functionality
- Configuration
- Compatibility
- Instructions
- Supporting documentation
- Warranty
- Technical support

- Operation
- Inspection e.g. error logs
- Maintenance
- Renewal
- Decommissioning
- Asset inventory

- Supplier costs
- Customer price
- Support costs
- Value of asset
- Life of asset

Potential consequences of software failures on railway system

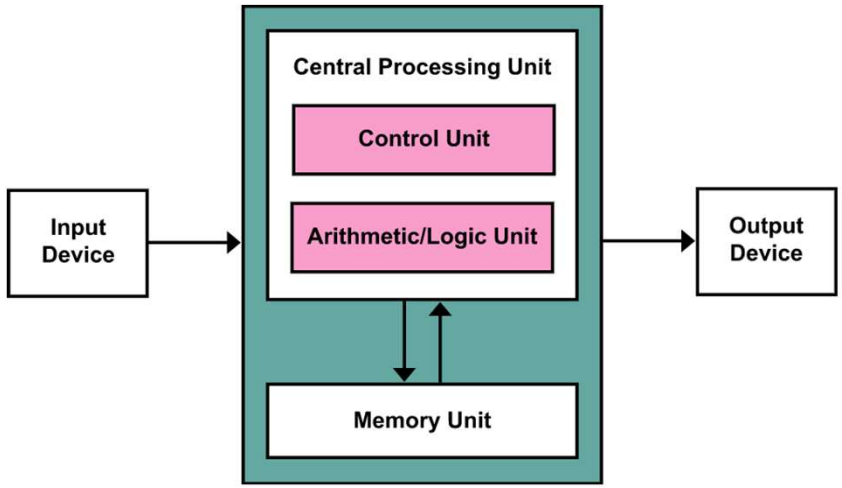
- Safety, health & environment
- Performance
- Cost
- Reputation & satisfaction
- Asset management

Some examples

- Train accident
- Personal injury or health effect
- Release of hazardous material
- Impact on service/asset reliability & availability
- Loss of system security
- Unnecessary action due to false alarms

Software failure modes and effects

Examples of software failure modes
Incorrect software installation/configuration
Incorrect functionality provided
Required functionality not provided
Response too quick or too late
No event or incorrect event sequence
Components not synched or communicating
Incorrect inputs/actions not detected or recovered from
Data is corrupt, incorrect or in wrong units/format
Unable to detect and recover from failure
False detection of failure
Out of memory or processing slowly
Incorrect use, misuse or abuse of system

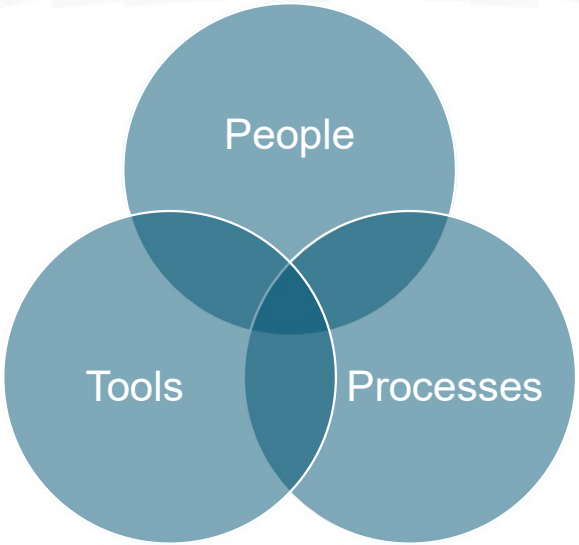


Examples of software failure effects

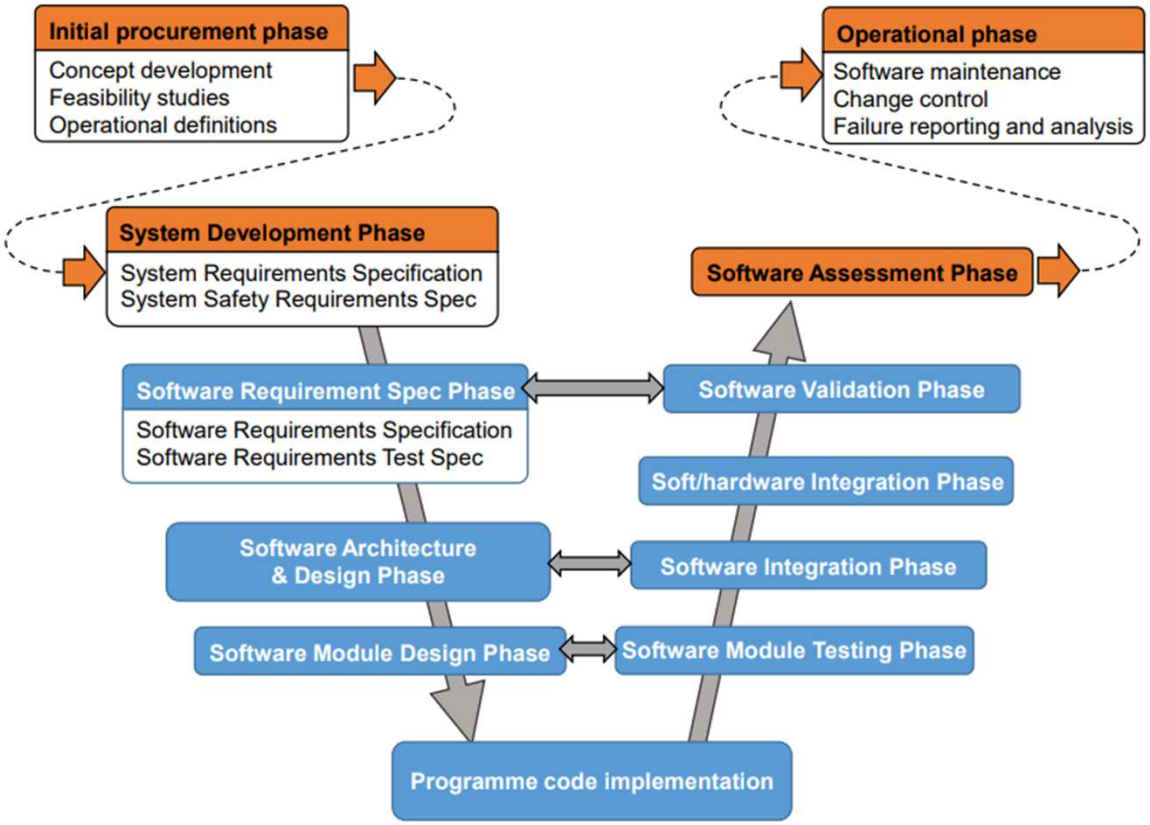
- System crashes, freezes, ‘hangs’ or runs slowly
- Performs right function at wrong time
- Programme terminates early
- Incorrect or no data generated
- Performs wrong function
- No function performed
- Input not acted upon
- Loss or corruption of data

Key standards and guidance for railways

- BS EN **50126**. The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) (IEC 62278)
- BS EN **50128**. Communication, signalling and processing systems. Software for railway control and protection systems (IEC 62279)
- BS EN **50129**. Communication, signalling and processing systems. Safety-related electronic systems for signalling (IEC 62425)
- BS EN 50155. Rolling stock. Electronic equipment (IEC 60571)
- BS EN **50159**. Communication, signalling and processing systems. Safety-related communication in transmission systems (IEC 62280)
- BS EN **50657**. Rolling stock applications. Software on board rolling stock
- Rail Industry Guidance Note **GEGN8650**. Guidance on High Integrity Software-Based Systems for Railway Applications



Software development lifecycle



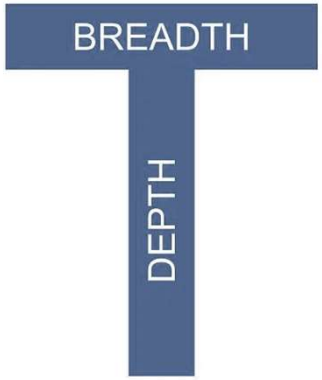
Example failure modes

- Missing or conflicting requirements
- Misunderstanding of requirements
- Incorrect specification
- Design or coding errors
- Errors not identified in testing
- Insufficient error detection
- Failures introduced by changes

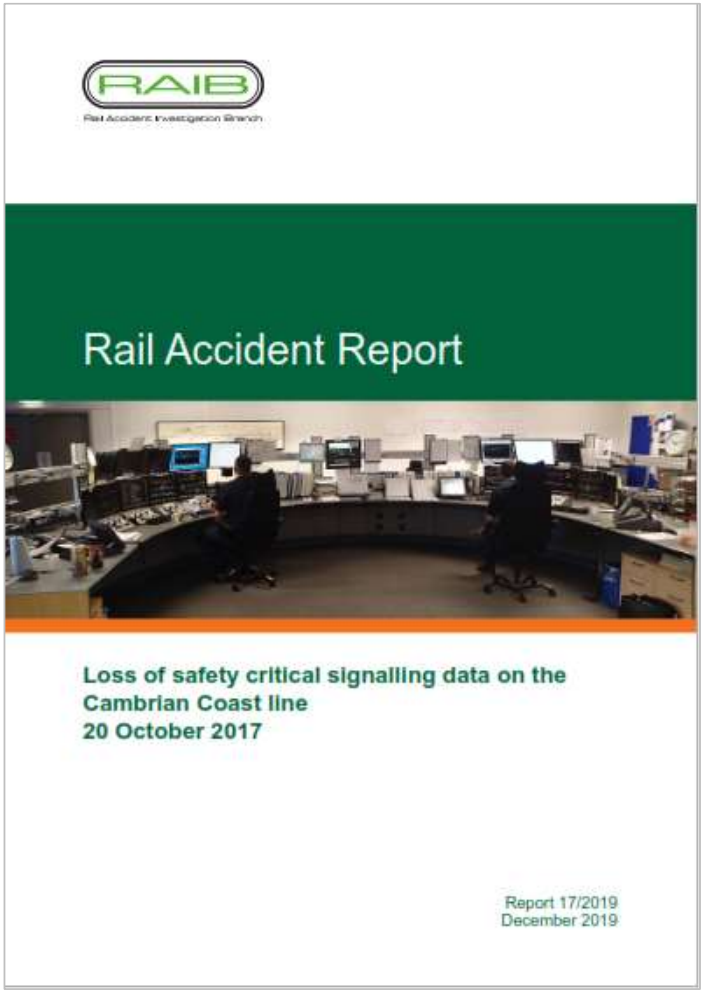
Examples of mitigations

- Design out failure mode
- Redundancy in design
- Enter fail-safe mode
- Health monitoring
- Regression testing
- Training
- Assurance activities

Improving our digital competency



Case Study: RAIB Investigation into incident on 20 October 2017



- Four trains travelled over the Cambrian Coast line while temporary speed restriction (TSR) data was not being sent to the trains by the European Rail Traffic Management System (ERTMS) signalling system
- TSR data was not uploaded during an automated signalling computer restart the previous evening, but a display screen incorrectly showed the restrictions as being loaded for transmission to trains
- Immediate cause: the ERTMS signalling system was returned to service following a Radio Block Centre (RBC) software automatic reset, known as a 'rollover', without temporary speed restriction information for transmission to trains.

Design Issues

- Key operational data not uploaded to the main system after a software rollover due to a sub-system fault condition.
- No indication provided that the system had failed to the operators.
- Loss of key operational data during system rollover due to use of volatile memory.
- Sub-system software unable to detect and manage the corruption of its own database.

Process Issues

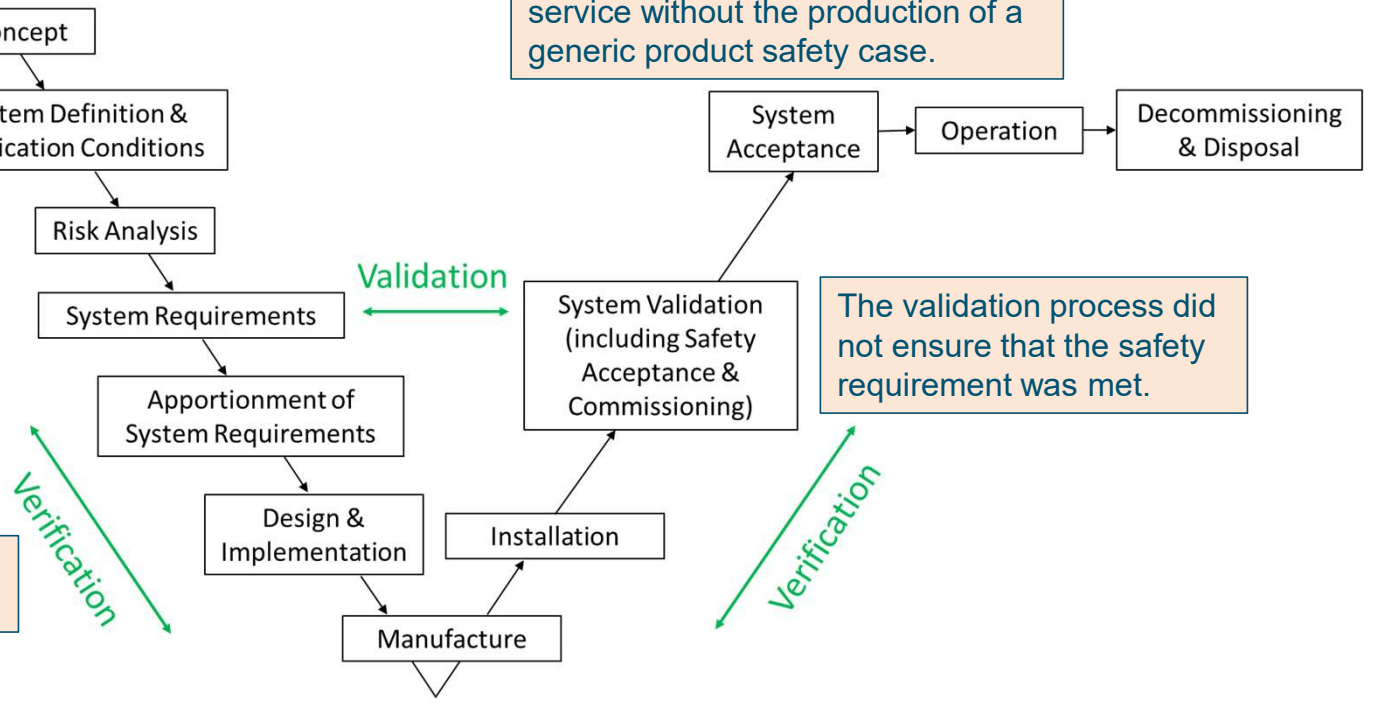
The sub-system was accepted into service without the production of a generic product safety case.

Hazard analysis process did not identify and mitigate against the software thread failure mode.

Insufficient definition of the safety-related software requirements for the sub-system software.

Vulnerability of the system to a single point of failure not detected or corrected during the design, approval and testing phases.

Required level of safety integrity for validation of the data uploaded not achieved by the design.



The validation process did not ensure that the safety requirement was met.

Learning from experience

- Important to be able to understand the potential failure modes and effects
- Building a library of case studies for transferrable learning
- Looking for further examples from rail and other industries
- Improved reporting and investigation of events involving software-based systems
- System event data recording and playback is important for investigation
- Use of Data Reporting, Analysis and Corrective Action Systems (DRACAS)

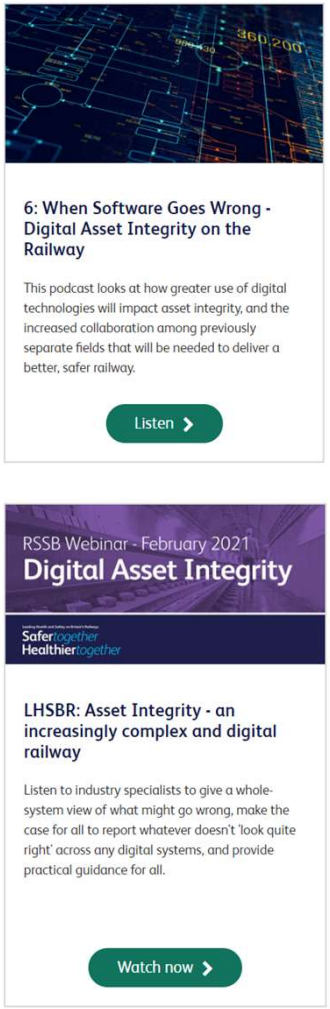
Some useful resources:

https://safety.networkrail.co.uk/improving-railway-system-safety/

https://www.rssb.co.uk/safety-and-health/learning-from-experience/accident-investigation-and-learning/learning-from-other-sectors

When Software Goes Wrong (rssb.co.uk)

https://www.rssb.co.uk/en/services-and-resources/services/events/Past-Webinars



Client role on projects involving high integrity software-based systems

Define clear, complete and nonconflicting requirements

Set expectations on supplier(s) throughout project lifecycle

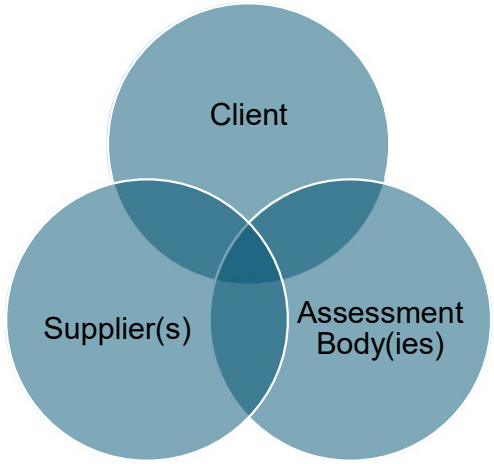
Select competent and capable suppliers

Specify the role of independent assessment bodies

Involvement in key project stages such as:

- determining safety requirements and integrity levels
- providing operational context and understanding of interfaces
- hazard identification and mitigation
- understanding the residual risks
- testing, commissioning, operation and maintenance
- verification and validation processes
- audit and inspection as part of the assurance process

Contractual requirements based on good engineering safety management principles



Consider assurance and independent assessment outputs

Retain key project and asset records

Risk-based approach to assurance

External Interfaces

Human factors

Sub-system interfaces

Compatibility

Change control

New/novel approach

Multiple suppliers

Re-using or adapting items from previous applications

Timescale constraints

In summary

more technology = exciting prospects + potential challenges

standards + guidance + tools = reduced risk

learn from experience = improved competence

client = key role on projects

Thank you for listening