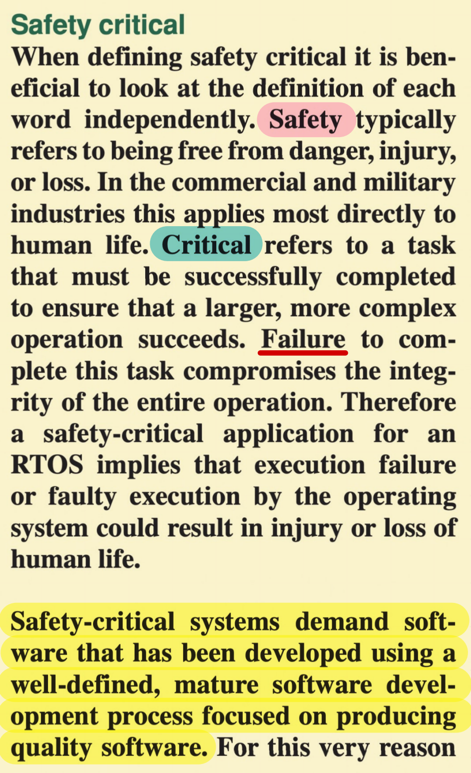
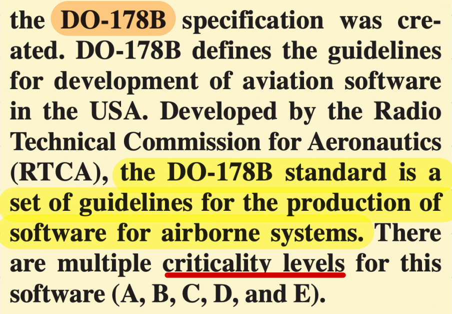
Mission-Critical vs. Safety-Critical

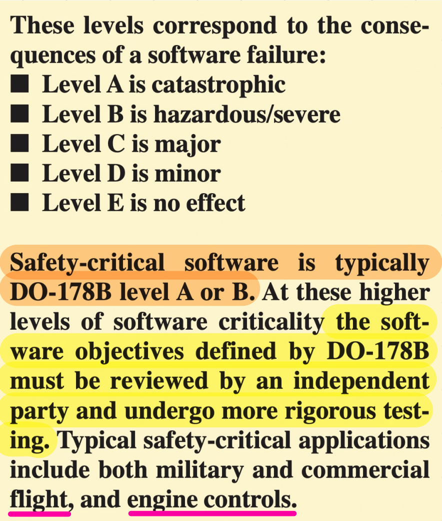


Safety critical

When defining safety critical it is beneficial to look at the definition of each word independently. Safety typically refers to being free from danger, injury, or loss. In the commercial and military industries this applies most directly to human life. Critical refers to a task that must be successfully completed to ensure that a larger, more complex operation succeeds. Failure to complete this task compromises the integrity of the entire operation. Therefore a safety-critical application for an RTOS implies that execution failure or faulty execution by the operating system could result in injury or loss of human life.



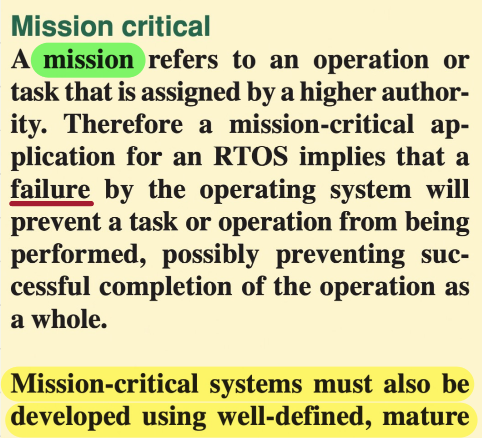
Safety-critical systems demand software that has been developed using a well-defined, mature software development process focused on producing quality software. For this very reason the DO-178B specification was created. DO-178B defines the guidelines for development of aviation software in the USA. Developed by the Radio Technical Commission for Aeronautics (RTCA), the DO-178B standard is a set of guidelines for the production of software for airborne systems. There are multiple criticality levels for this software (A, B, C, D, and E).



These levels correspond to the consequences of a software failure:

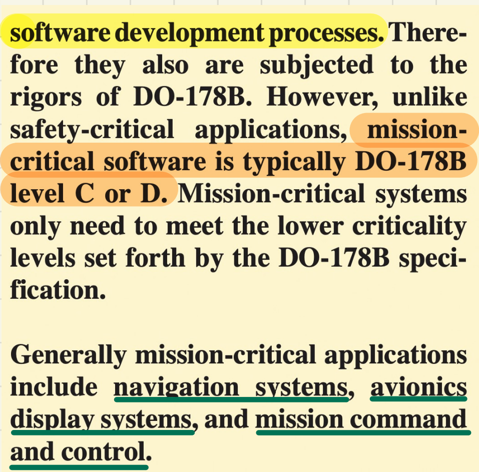
- ■ Level A is catastrophic
- ■ Level B is hazardous/severe
- ■ Level C is major
- ■ Level D is minor
- ■ Level E is no effect

Safety-critical software is typically DO-178B level A or B. At these higher levels of software criticality the software objectives defined by DO-178B must be reviewed by an independent party and undergo more rigorous testing. Typical safety-critical applications include both military and commercial flight, and engine controls.



Mission critical

A mission refers to an operation or task that is assigned by a higher authority. Therefore a mission-critical application for an RTOS implies that a failure by the operating system will prevent a task or operation from being performed, possibly preventing successful completion of the operation as a whole.

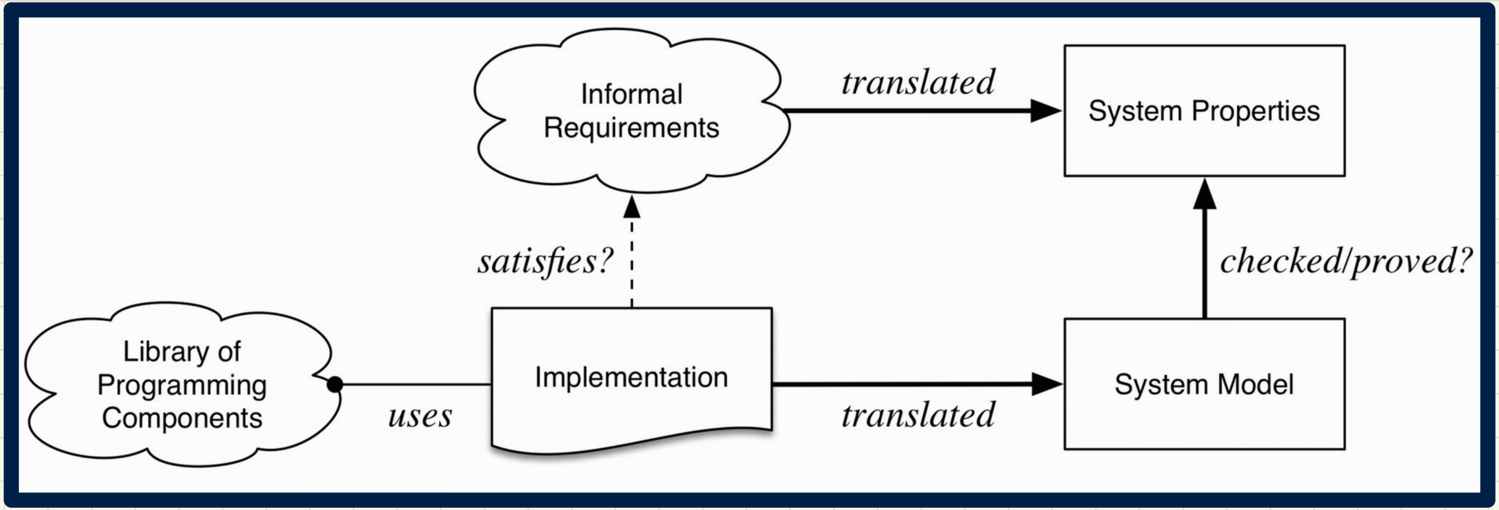


Mission-critical systems must also be developed using well-defined, mature software development processes. Therefore they also are subjected to the rigors of DO-178B. However, unlike safety-critical applications, mission-critical software is typically DO-178B level C or D. Mission-critical systems only need to meet the lower criticality levels set forth by the DO-178B specification.

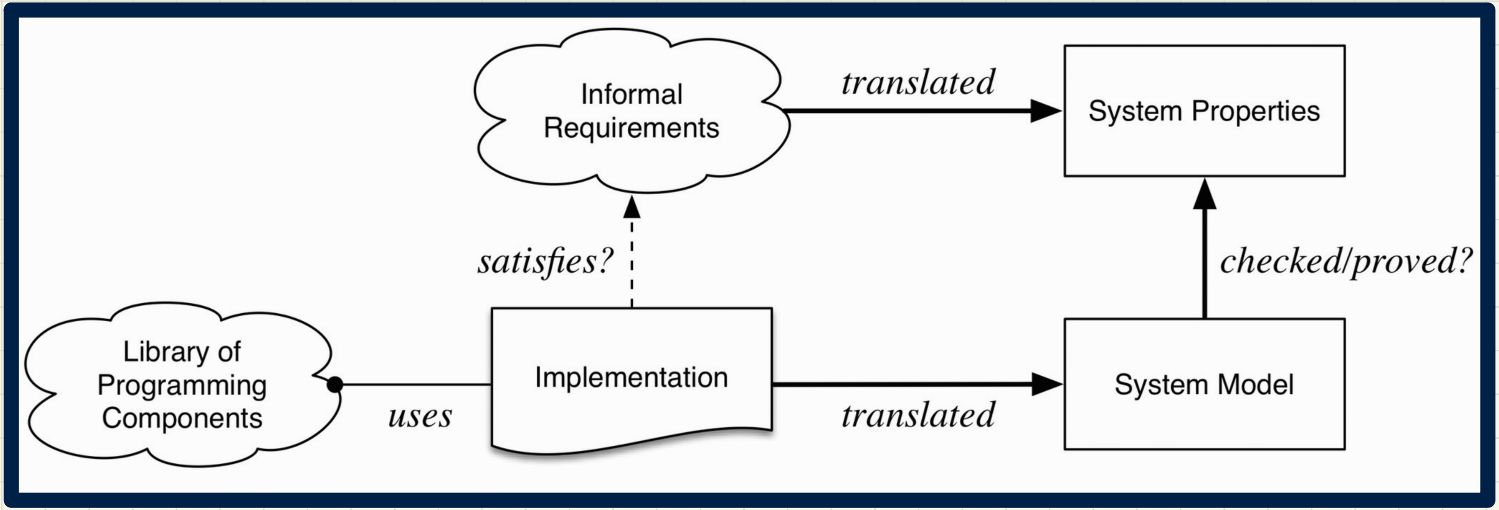
Generally mission-critical applications include navigation systems, avionics display systems, and mission command and control.

Source: http://pdf.cloud.opensystemsmedia.com/advancedtca-systems.com/SBS.Jan04.pdf

Building the product right?

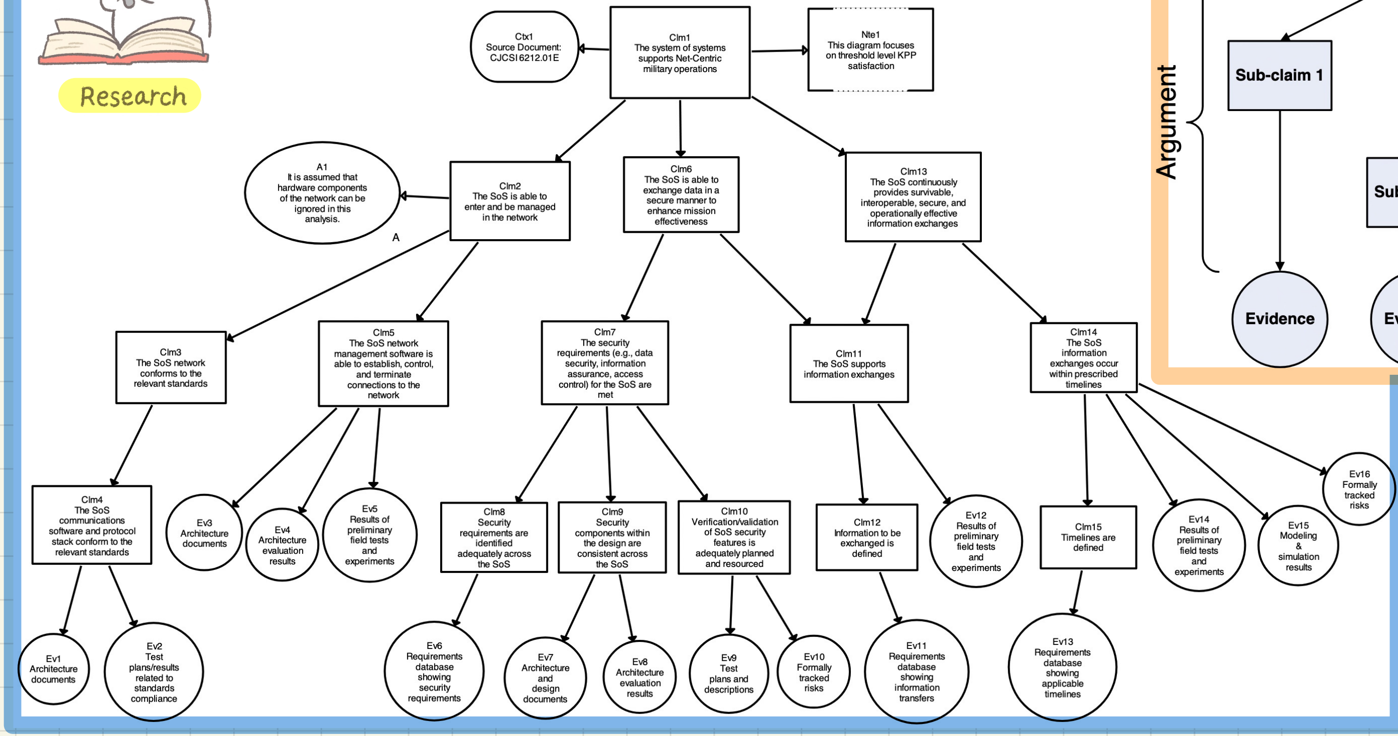


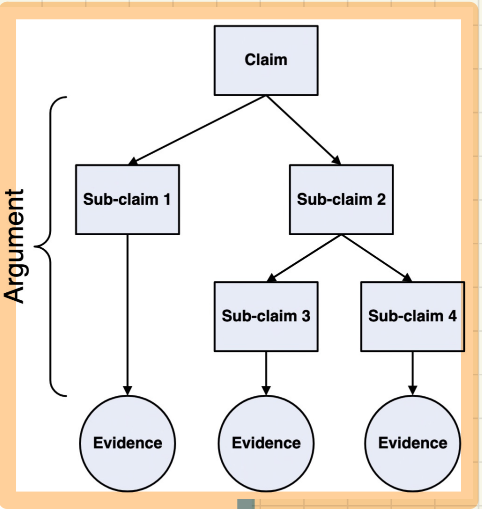
Building the right product?



Certifying Systems: Assurance Cases

Research on "Assurance Cases" if interested!


Ctx1
Source Document: CJCSI 6212.01E
Clm1
The system of systems supports Net-Centric military operations
Nte1
This diagram focuses on threshold level KPP satisfaction
A1
It is assumed that hardware components of the network can be ignored in this analysis.
A
Clm2
The SoS is able to enter and be managed in the network
Clm6
The SoS is able to exchange data in a secure manner to enhance mission effectiveness
Clm13
The SoS continuously provides survivable, interoperable, secure, and operationally effective information exchanges
Clm3
The SoS network conforms to the relevant standards
Clm5
The SoS network management software is able to establish, control, and terminate connections to the network
Clm7
The security requirements (e.g., data security, information assurance, access control) for the SoS are met
Clm11
The SoS supports information exchanges
Clm14
The SoS information exchanges occur within prescribed timelines
Ev16
Formally tracked risks
Clm4
The SoS communications software and protocol stack conform to the relevant standards
Ev3
Architecture documents
Ev4
Architecture evaluation results
Ev5
Results of preliminary field tests and experiments
Clm8
Security requirements are identified adequately across the SoS
Clm9
Security components within the design are consistent across the SoS
Clm10
Verification/validation of SoS security features is adequately planned and resourced
Clm12
Information to be exchanged is defined
Ev12
Results of preliminary field tests and experiments
Clm15
Timelines are defined
Ev14
Results of preliminary field tests and experiments
Ev15
Modeling & simulation results
Ev1
Architecture documents
Ev2
Test plans/results related to standards compliance
Ev6
Requirements database showing security requirements
Ev7
Architecture and design documents
Ev8
Architecture evaluation results
Ev9
Test plans and descriptions
Ev10
Formally tracked risks
Ev11
Requirements database showing information transfers
Ev13
Requirements database showing applicable timelines



Source: https://resources.sei.cmu.edu/asset_files/whitepaper/2009_019_001_29066.pdf

Exam Info

- When: 9am to 12pm, Thursday, December 11 (ACW 206)
- Coverage: **Everything** (lecture materials & labs)
  + slides, iPad notes
- Even problems that look **challenging** at first are built on the **same** foundational techniques you've learned and practiced in **lectures** and **labs**. A solid, reflective grasp of the basics will take you **farther** than memorizing examples.
- Format: Mostly Written
  + explanations/justifications + write math expressions + calculations, proofs
- Restrictions:
  + **One-sided**, computer-typed, **min** 10pt data sheet
  + No sketch paper (Exam booklet includes it) + No calculator
- What you should bring:
  + Valid, Physical Photo ID (strict)
  + Water/Snack