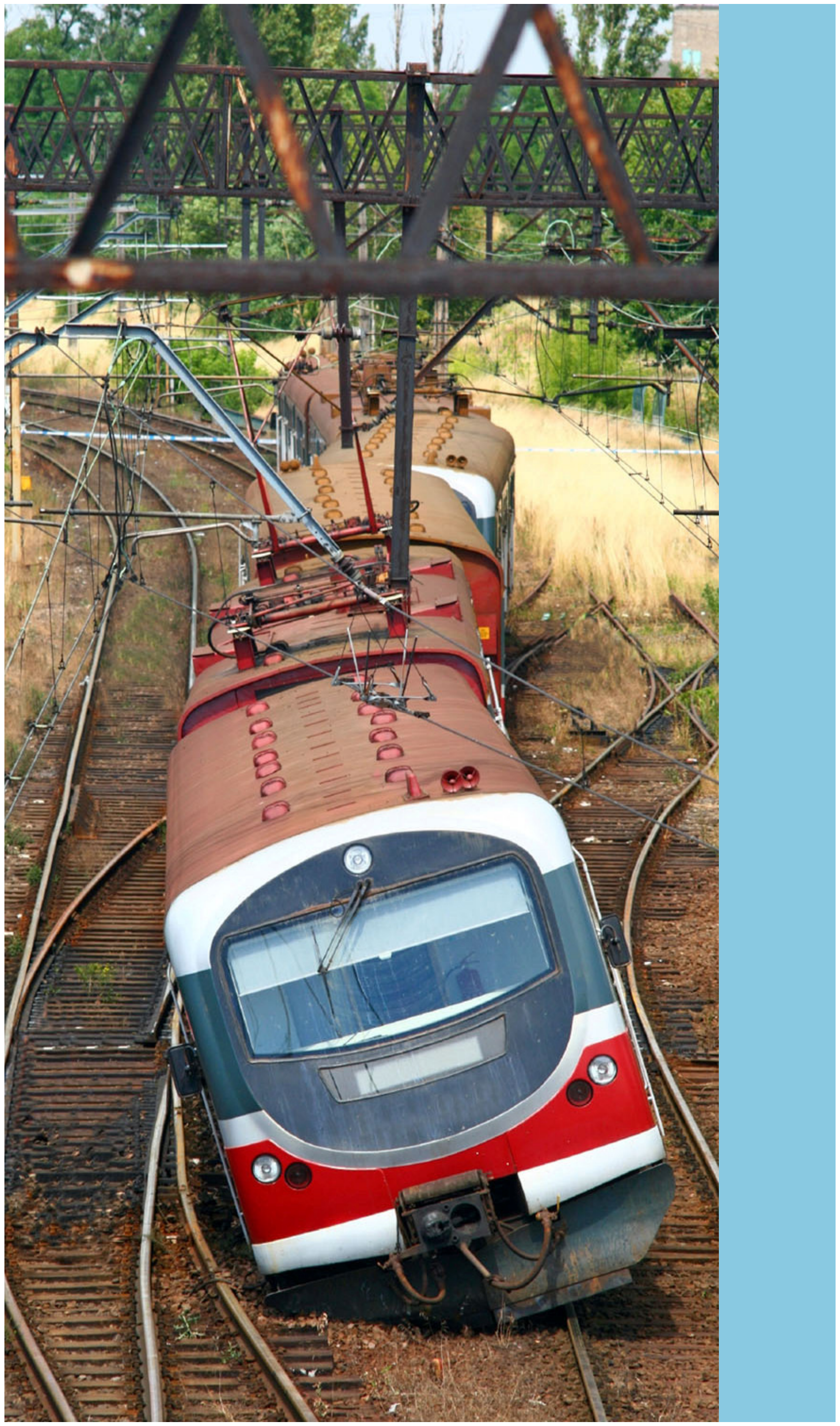
When COTS is not SOUP
Commercial Off-the-Shelf Software in Medical Systems

Chris Hobbs, Senior Developer, Safe Systems

QNX SOFTWARE SYSTEMS



Audience and Assumptions

- Who will benefit from this presentation?
  - Software designers
  - Development managers
  - Anyone who must decide on a strategy for building a safe system
    - i.e. a software system for medical device
- Assumptions — some knowledge of
  - IEC 61508, 62304, etc.
  - Safety Integrity Levels (SIL)
  - Software development
- Whitepapers
  - www.qnx.com/download

QNX SOFTWARE SYSTEMS

Contents

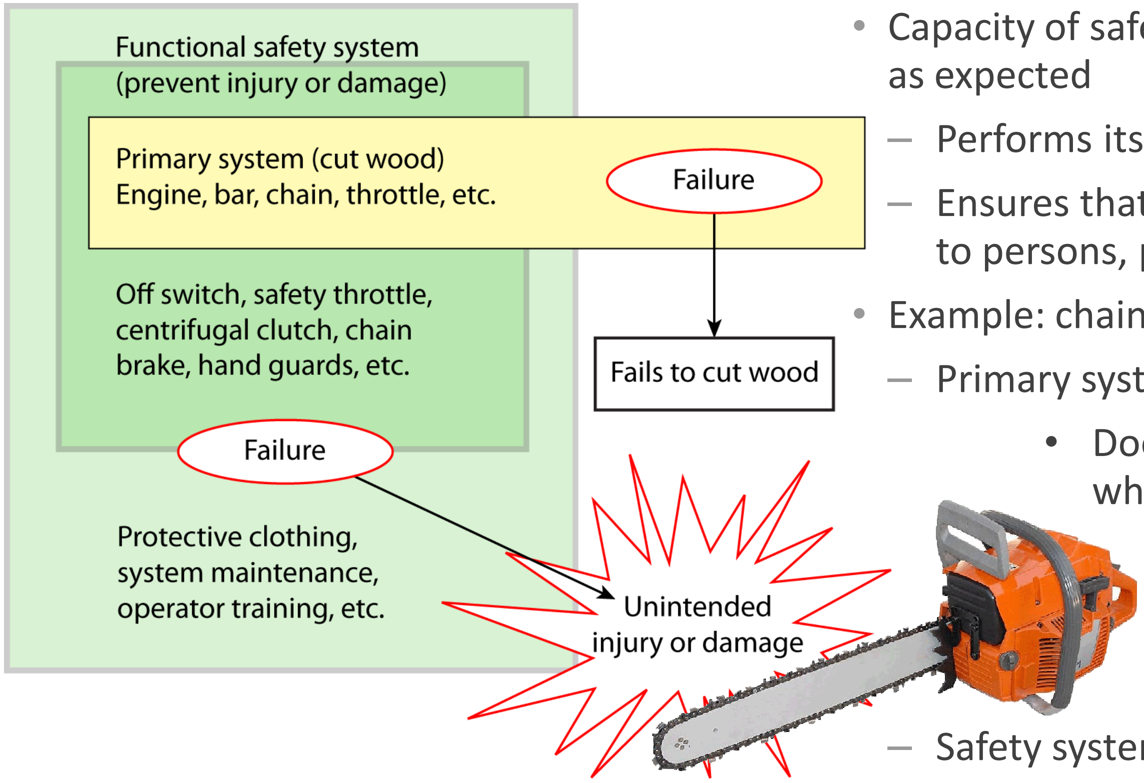
QNX SOFTWARE SYSTEMS

Part I: Functional Safety

- What is functional safety?
- How do we certify functional safety?
- Deterministic systems
- Non-deterministic systems
- Demands on development

QNX SOFTWARE SYSTEMS

Functional Safety


Functional safety system
(prevent injury or damage)
Primary system (cut wood)
Engine, bar, chain, throttle, etc.
Failure
Fails to cut wood
Off switch, safety throttle, centrifugal clutch, chain brake, hand guards, etc.
Failure
Protective clothing, system maintenance, operator training, etc.
Unintended injury or damage

- Capacity of safety-related system to function as expected
  - Performs its primary task
  - Ensures that no unacceptable risk or harm to persons, property, environment
- Example: chainsaw
  - Primary system cuts wood
    - Does damage to wood, but that's what it's designed to do
  - Safety system
    - Prevents chainsaw from cutting the operator, neighbours, etc.

Functional Safety

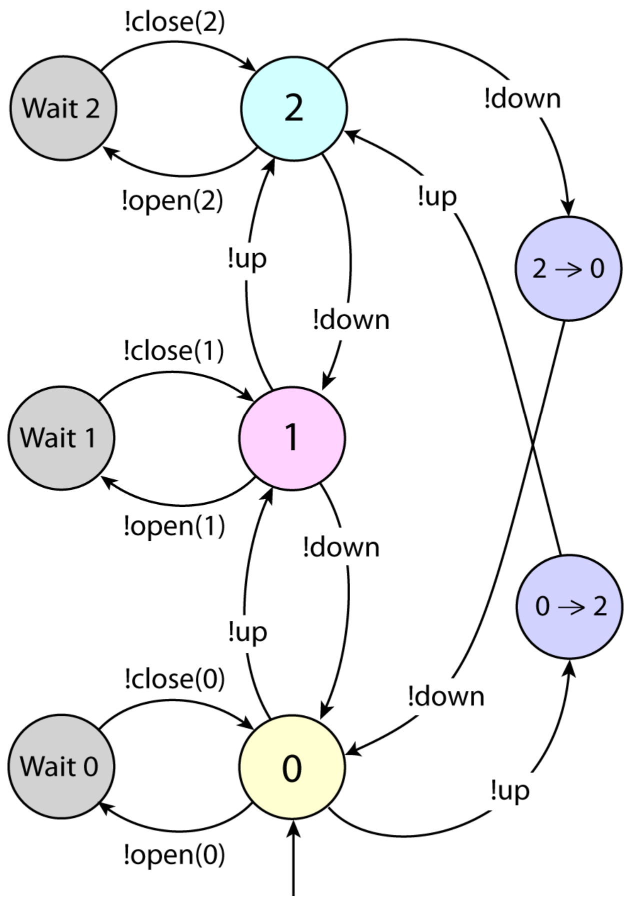
- Standards
  - IEC 62304 (medical)
  - IEC 61508 (electrical/electronic/programmable)
  - ISO 26262 (automotive)
  - CENLEC EN 5012*x* series (rail transportation)
  - etc.
- Safety Integrity Levels (SIL)
  - It's not about five-nines, or any other kind of nine
  - It's about sufficient dependability
- Proofs
  - Why can't we just test?

Failures per year	Duration of each failure	
1	5 minutes 16 seconds	Potentially catastrophic
10	32 seconds	
100	3.2 seconds	
1000	316 milliseconds	
10,000	32 milliseconds	
100,000	3.2 milliseconds	
1,000,000	316 microseconds	Possibly benign

Five-nines availability sounds good, but …

Would you submit to neurosurgery with instruments that make this claim, with no further precision?

QNX SOFTWARE SYSTEMS



Deterministic Systems

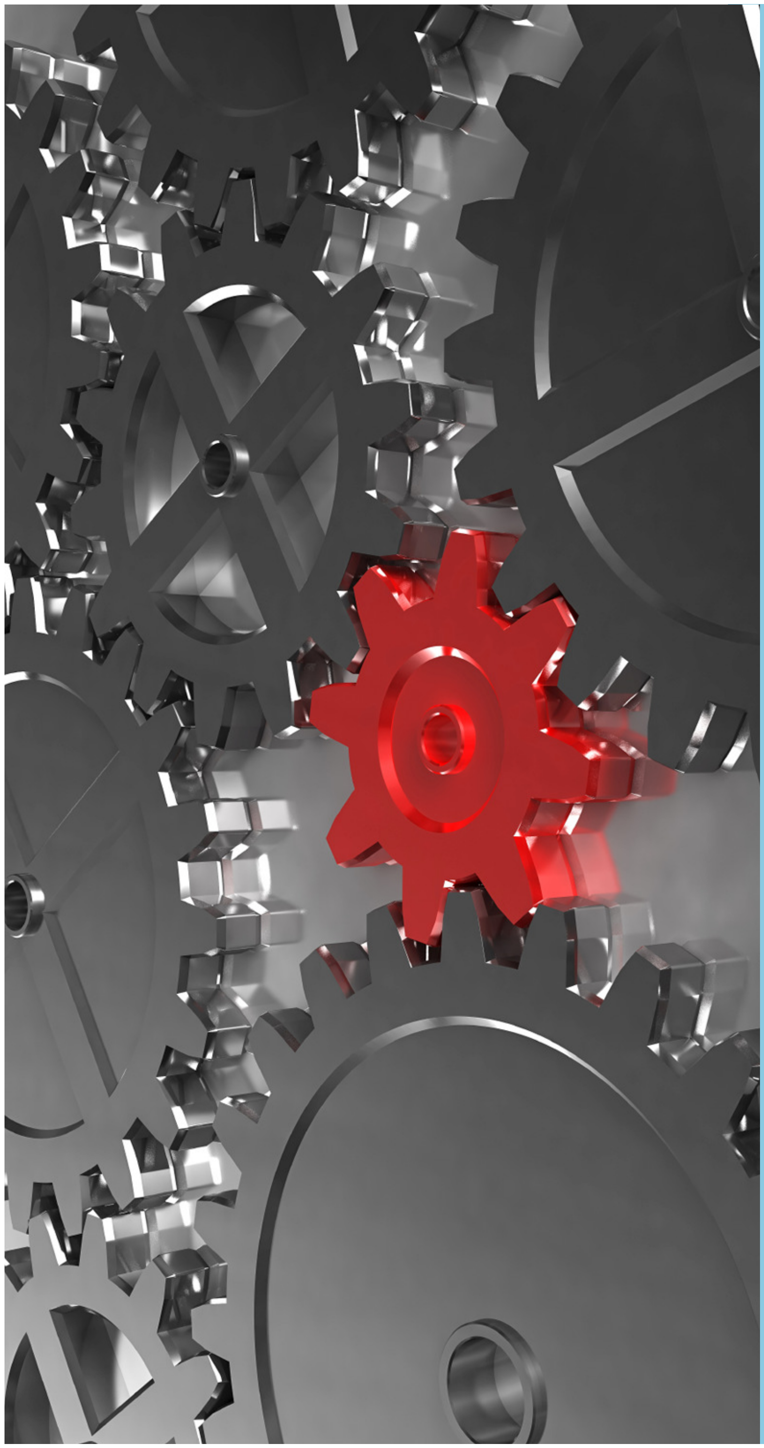
"If a device has few enough internal stored states that it is practical to cover them all in testing, it may be better to regard it as hardware"

Application Note 2: Software and EN 50128, London: Railway Safety, 2003, p. 3.

A simple elevator system.

It is simple enough for us to find the fault through testing—but only if we ask the right question.

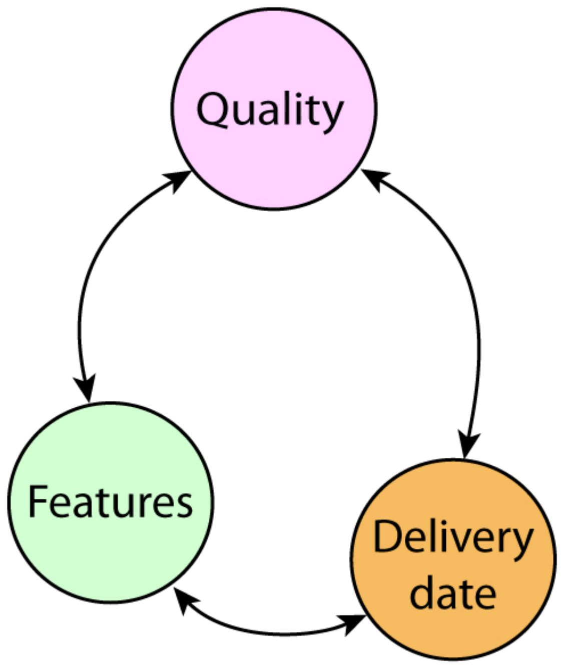
QNX SOFTWARE SYSTEMS



Non-deterministic Systems

- Complex
- Theoretically deterministic
  - A finite number of states and transitions
- In practice, non-deterministic
  - Impossible to identify all possible states and transitions
- Testing to requirements is not enough.
  - Testing will not satisfy medical certification requirements.
- One reason is sufficient to explain why testing is insufficient:
  - Testing can only demonstrate the presence of faults; it cannot prove that there are no faults.

QNX SOFTWARE SYSTEMS



Management
may choose two.

Demands on Development

- Just the usual
  - Time
  - Money
  - Features
  - Certification
    - FDA, MDD, Health Canada, etc., etc.

- And the usual solutions
  - Roll your own
  - COTS (Commercial Off-the-Shelf)

QNX SOFTWARE SYSTEMS

Part II: COTS and SOUP

- What's a girl to do?
- COTS?
- COTS versus SOUP
- False alternatives
- IEC 62304
- Waiter! Two SOUPs!
- When COTS is SOUP
- IEC 62304 and SOUP
- SOUP and Clear SOUP
- When COTS is not SOUP
- When COTS is Clear SOUP

QNX SOFTWARE SYSTEMS

What’s a Girl to Do?

- If I can’t use SOUP, then I can’t use COTS—Wrong!
- Roll Your Own
  - Control everythng from start to finish
  - But … the requirements for functional safety validation and certification do not change
  - “Had we but world enough and time”
    - … and skill

QNX SOFTWARE SYSTEMS

COTS?

- Use commercially-available components
- OS (RTOS)
- Networking
- HMI technologies
- Faster, cheaper, less risk
- But … the requirements for functional safety validation and certification do not change

COTS versus SOUP

COTS
Commercial Off-The-Shelf

- From a reputable vendor, which stands behind the product
- Development processes are known and well-documented
- Adequate records
- Validation is formal and documented
- In other words, you are getting your medicine from a reputable pharmacist (America), chemist (U.K.), or apothecary (Germany)

This view of COTS is inaccurate and misleading

SOUP
Software Of Uncertain Provenance

- No vendor standing behind it
- Development processes not known or documented
- Inadequate records
- Validation not formal or documented
- In other words, you are getting your medicine from a web site that uses spam to advertise

This view of SOUP is inaccurate and misleading

False Alternatives

- COTS versus SOUP is the wrong question
  - COTS may be SOUP
  - SOUP may be clear SOUP
  - COTS may not be SOUP—this is what we are talking about today

- Better to re-phrase the question based on our
  - Functional safety requirements
  - Certification requirements

QNX SOFTWARE SYSTEMS

IEC 62304 (2006-05): 3.29

- IEC 62304: Medical device software — Software life cycle processes
  - 3.29 SOUP
    software of unknown provenance
    SOFTWARE ITEM that is already developed and generally available and that has not been developed for the purpose of being incorporated in the MEDICAL DEVICE (also known as "off-the-shelf software") or software previously developed for which adequate records of the development PROCESSES are not available
- IEC 62304
  - Software processes
  - Risk management
    - Invokes ISO 14971
  - Safety classifications
  - Only gives examples of quantified safety levels
- IEC 61508 and EN 50128 (for example) define common numerical values for acceptable failure rates
  - i.e. Safety Integrity Levels (SIL)

Waiter! Two SOUPs!

- IEC 62304 (2006-05): 3:29 definition of SOUP implies that
  - COTS is SOUP
    - "not been developed for the purpose of being incorporated in the MEDICAL DEVICE (also known as "off-the-shelf software")"
  - There are two types of SOUP
    - "not been developed for the purpose of being incorporated in the MEDICAL DEVICE (also known as "off-the-shelf software")"
    - "software previously developed for which adequate records of the development PROCESSES are not available"

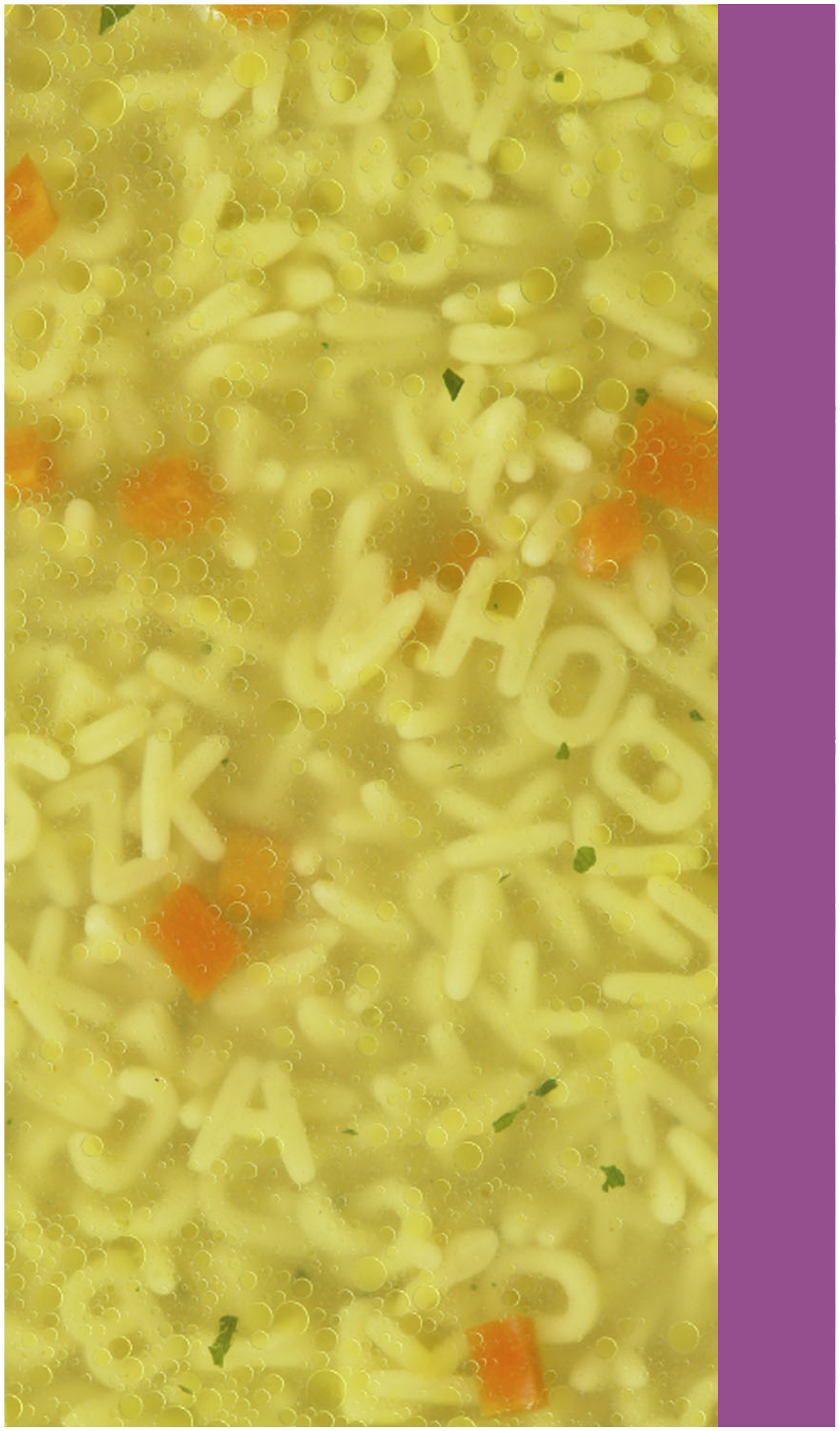
When COTS is SOUP

- Software may be sold by a reputable vendor, and still be SOUP for safety validation and certification
- For example:
  - Microsoft Windows OS
    - Reputable vendor
    - Documented processes, validation records, fault histories, etc.
  - Are these records available to support the functional safety case and the certification of a system build with this OS?
  - If they are not available, then the software (Microsoft Windows OS)
    - May be not be SOUP for Microsoft (We have no way of knowing)
    - Is definitely SOUP for us (We have no way of knowing)

IEC 62304 and SOUP

- IEC 62304 permits use of SOUP
- For example:
  - 5.3.3 Specify functional and performance requirements of SOUP item
  - 5.3.4 Specify SYSTEM hardware and software required by SOUP item
  - 7.1.3 EVALUATE published SOUP ANOMOLY list
  - 8.1.1 Identify SOUP
- These requirements suggest that we're *not* talking about software
  - "for which adequate records of the development PROCESSES are not available"

  But about software
  - "developed for the purpose of being incorporated in the MEDICAL DEVICE (also known as "off-the-shelf software")"
- In other words *clear* SOUP



SOUP and Clear SOUP

- Third-party software may be clear SOUP
- Open source (Apache and Linux)
  - Freely available source code
    Can be be scrutinized with symbolic path execution and path coverage analysis
  - Freely available fault histories
  - Long in-use histories
- Problems with open source
  - Development is not clearly defined or well documented
  - No way to know what has been validated or how
  - May include more features than required (dead code)
  - Expressly discouraged by IEC 61508
- Ergo: not recommended

QNX SOFTWARE SYSTEMS

When COTS is not SOUP

- COTS software must not be SOUP in the traditional definition of:
  - No process, no requirements, no records, etc., etc.
- COTS software may be clear SOUP
  - For functionally safe systems (i.e. medical systems), COTS must be clear SOUP

QNX SOFTWARE SYSTEMS

When COTS is *Clear* SOUP

- COTS software must have
  - Clear functional safety requirements
    - Clearly stated safety assurance claims, including the environment in which they are valid
  - Architecture suitable for safety-critical applications
  - Clearly defined and documented development and validation processes
  - Documented fault histories
- If the COTS software has the above, it is clear SOUP and, thus, can be used and certified in a medical device
- COTS software is not SOUP when it is *clear* SOUP
- Also useful
  - Some vendors release to customers the processes they use to build their software (an informal audit trail)
  - Some vendors have certified components (e.g. QNX Neutrino RTOS Safe Kernel certified to IEC 61508 SIL 3)

QNX SOFTWARE SYSTEMS

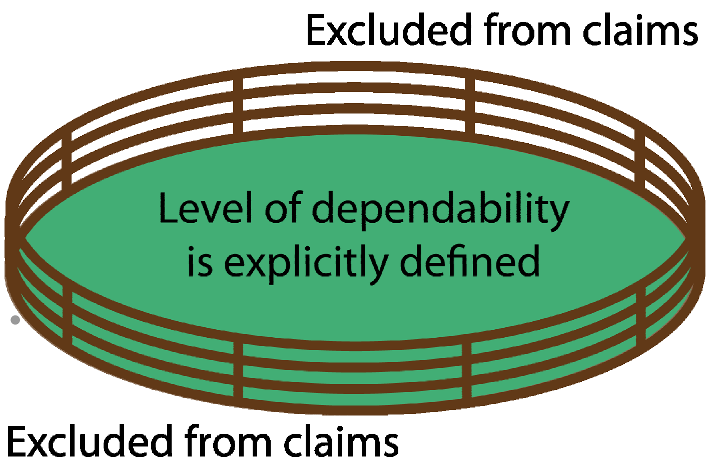
Part III: The Evidence

- What proofs does the COTS software vendor provide that his product is what you need?
  - Will it support your getting your product certified?
  - Is it clear SOUP?
- Checklist
  - Functional safety requirements
  - System architecture
  - Process
  - Fault-tree analysis
  - Design artefacts
  - Static analysis
  - Proven in-use data
  - Safety manual
  - Certified components

QNX SOFTWARE SYSTEMS

Functional Safety Requirements

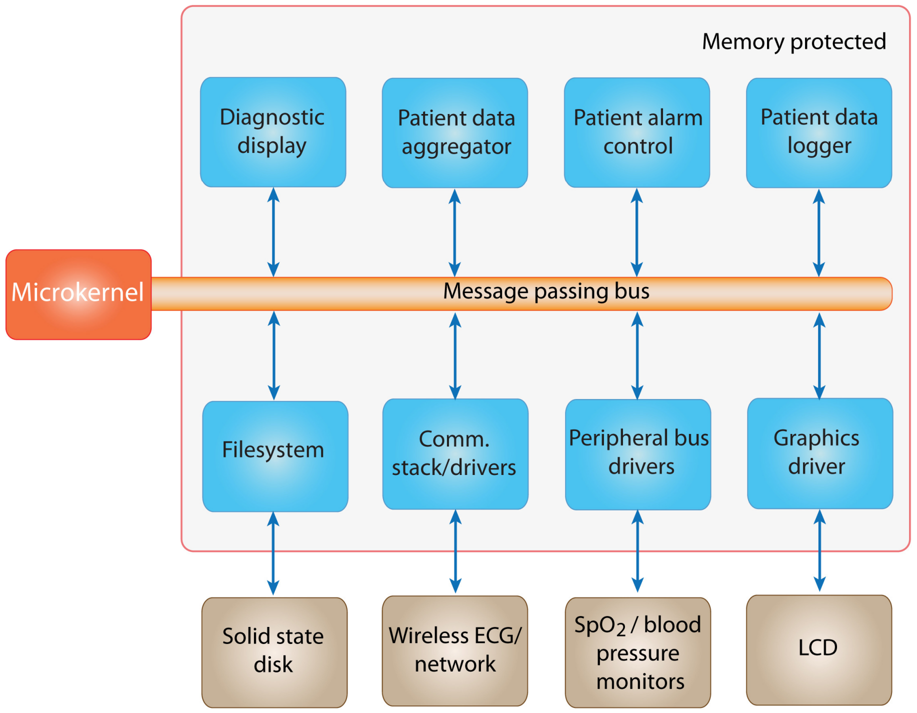
- What functional safety claims does or can the COTS software make?
- Context and limits of the claims
- Continuous or on-demand operation
- Probability of dangerous failure



- Level of dependability claimed (quantitative)
  - Availability
    - How often the system responds to events in a timely manner
  - Reliability
    - How often these responses are correct
- Define sufficient dependability

System Architecture

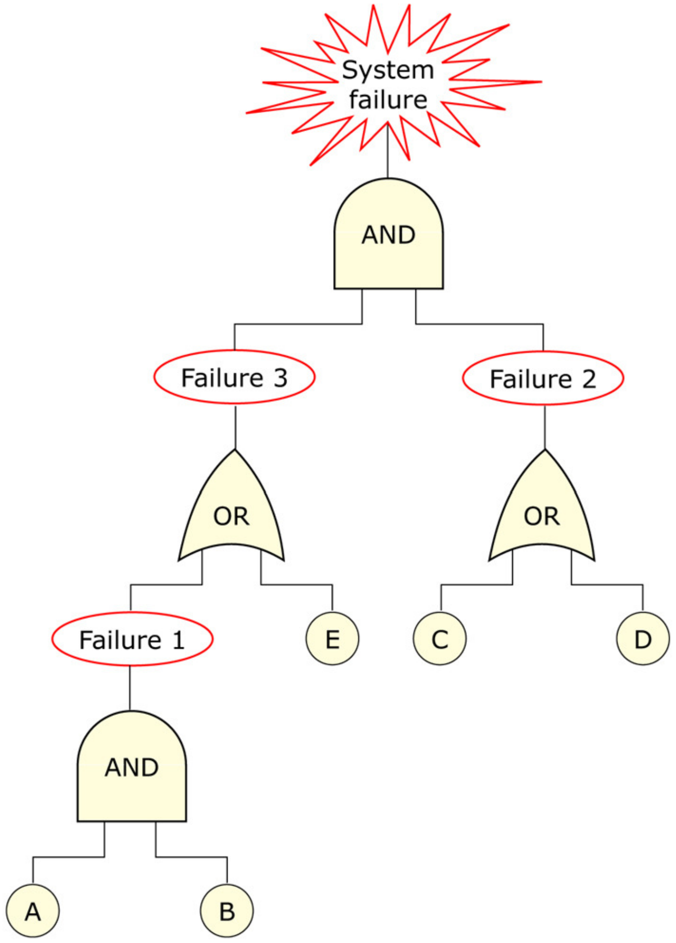
- Evaluate architectures and select the one most suitable for your safety-critical systems and certification
- Some characteristics to look for
  - Priority inversion protections
  - Guaranteed availability
    - CPU resource scheduling
    - Prevent critical process starvation
  - Facilitate migration to multicore systems (they are coming)
    - Ensure correct behaviour on these systems



QNX SOFTWARE SYSTEMS

Process

- What processes does the COTS vendor have in place and documented?
- Process — Quality management system
  - *Sine qua non* — if your COTS vendor doesn't have this, go no further
  - Quality management system
    - ISO 9000 — family of standards for quality management systems
    - ISO 15504 — Software Process Improvement Capability Determination (SPICE)
    - Capability Maturity Model Integration (CMMI)
  - Source control
    - Revision/version/source control
  - Defect tracking
    - Defects found by customers as well as through verification and validation
    - Defect classification (for fault analysis)



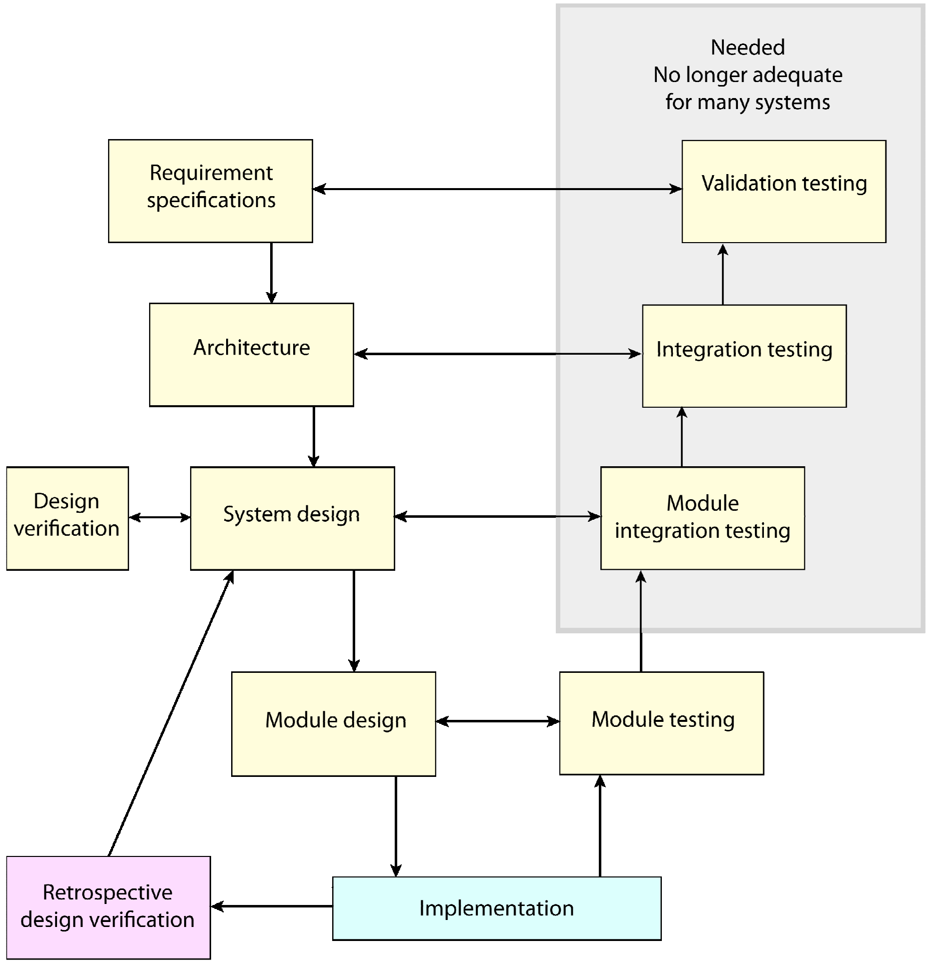
Fault Tree Analysis

- Was the COTS software evaluated with fault-tree analysis?
- Structured analysis
  - Easier for auditor
  - Easier for audited
- Example: Bayesian Belief Networks
  - tool for incorporating and providing quantitative results from
    - Hard and soft evidence
    - *A priori* (cause to effect) and *a posteriori* (effect to cause) evidence

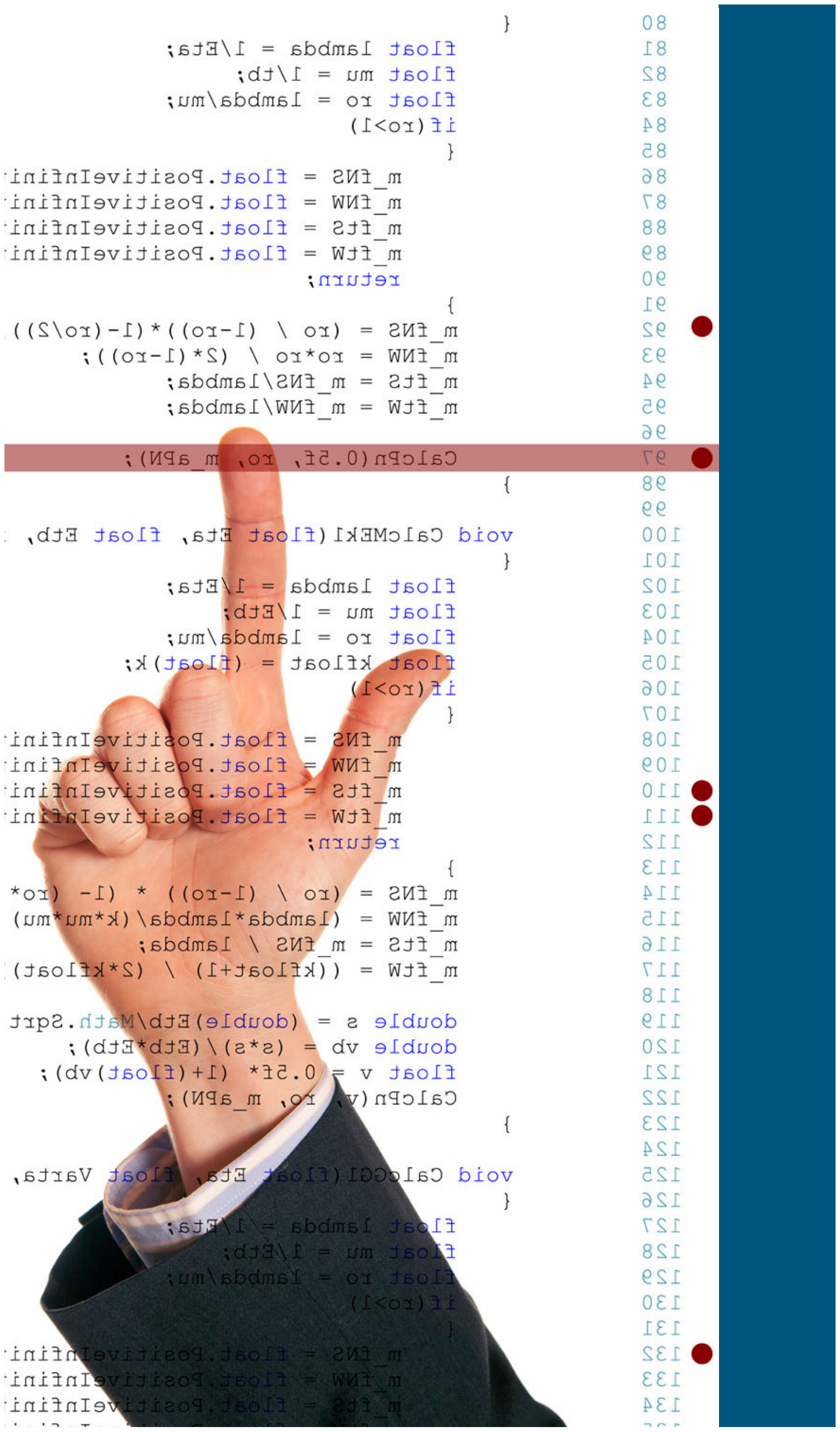
QNX SOFTWARE SYSTEMS

Design Artefacts

- What design artefacts come with the COTS software?
- Design documentation
  - Project plan
  - Quality plan
  - Architectural design
  - Detailed design
  - Test plans
  - Test results
  - Other validation methods plans and results
  - Traceability matrix
    - Requirements to delivery
- Records from software life cycle
  - Changes
  - Issues and their resolutions



QNX SOFTWARE SYSTEMS



Static Analysis

- Has static analysis been used to support the COTS safety and certification case?
- Static analysis
  - Check syntax
  - Estimate probability of faults
  - Prove correctness
    - e.g. assertions in the code
  - Symbolic execution (static analysis-hybrid)
- What artefacts can the COTS vendor provide?

QNX SOFTWARE SYSTEMS

Proven-In-Use Data

- What can the vendor provide?
- In-field usage data are invaluable
  - Build the gathering of this data into your business model — even if you are not considering attempting certification soon
  - Can be used to support safety cases
  - The more in-use data available, the stronger the evidence
  - In-use data only meaningful when scrutinized with fault analysis

- QNX used proven-in-use data as a component in its safety assurance/IEC 61508 conformance case for the QNX® Neutrino® RTOS Safe Kernel.

- Of interest
  - Proven-In-Use data for a traditional kernel is invalidated by the addition of a new driver. A micro-kernel on the other hand …

QNX SOFTWARE SYSTEMS

The Safety Manual

- This is another sine qua non
- Remember the functional safety claims
- The Safety Manual
  - Defines constraints
    - Environment
    - Usage
  - For example:
    - "This list of processor architectures is exhaustive."
    - "Floating point operations SHALL NOT be performed in a signal handler."
    - "Critical budgets are limited to the window size."
  - However, constraints must not impact usability
    - For example: a statement such as "Must not be used below an elevation of 2,000 metres" has
      - No impact on an aircraft cabin pressure system, which is not really needed below 2,000 metres
      - May not be acceptable for a dialysis system, because many people live below 2,000 metres

QNX SOFTWARE SYSTEMS

Certified Components

- Certifying agencies (FDA, MDD) certify the entire device, not the components
- Certified components can help
- Certified components (IEC 61505, IEC 62304, etc.) come with
  - Documented processes
  - Artefacts you can include in your assurance case for certification
  - Company expertise in obtaining certification

Summary

- Understand functional safety
- Make clear and understand your safety and certification requirements
- Roll your own doesn’t solve the problem
- You can use SOUP if it is *clear* SOUP
- COTS can be clear SOUP (but is not always clear SOUP)
- When selecting a COTS vendor, understand how he makes his safety assurance case, and see the evidence
- Using certified components helps you get your system certified
- A COTS vendor who has been through certification(s), can help you get through your certification(s)

Chris Hobbs, Senior Developer, Safe Systems
chobbs@qnx.com

QNX SOFTWARE SYSTEMS