√EROCEL

Mission Critical and
Safety Critical Software

Medical Device Software under IEC 62304

George Romanski

IEC 62304

- Medical Device Software – Software Lifecycle Processes
  - Quality Management System*
  - RISK MANAGEMENT
  - Software Safety Classification
  - Development Process
  - Maintenance Process
  - Configuration Management*
  - Problem Reporting and Management*

* These processes are Universal between the standards

Software Safety Assessment

- For Avionics – ARP-4761 (Safety Assessment)
- For Medical – ISO 14971 (Safety/Risk) Normative reference
- For Automotive
  - Part of ISO 26262
- For Industrial
  - Part of IEC 61508
- For Trains
  - Part of EN 50128

Level chosen for System
then applied to Software

How do you assess for RTOS?

Different terms: Design Assurance Level, Software Integrity Level, Class
Same Principle: Consequence/Exposure, Severity -> Level for software

What level is Device Software?

- Software Faults do not follow “Gauss Normal” Distribution (Bell Curve)
  - Given 10,000 lines of code
  - You test and find 10 software faults
  - What is the probability of finding another fault with another test?
- We Don’t know distribution of faults
- Software must be assumed to be level C until shown by Risk Assessment that a lower level is applicable!

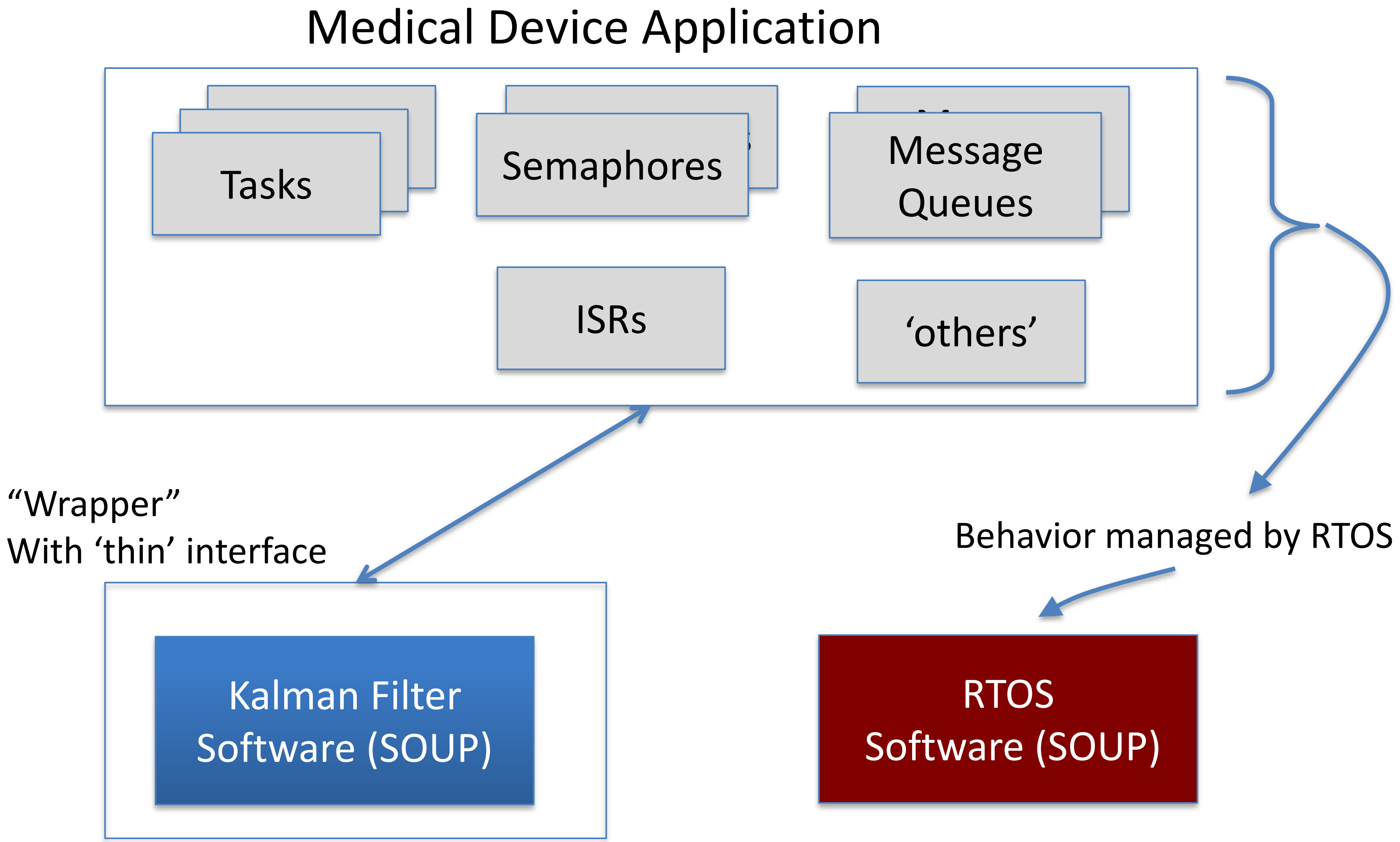
Software Risks

- Does it do what it should?
- Does it do More than it should?
- It does something wrong
- Is an action late
- Is an action too early
- Are a sequence of actions in the wrong order?

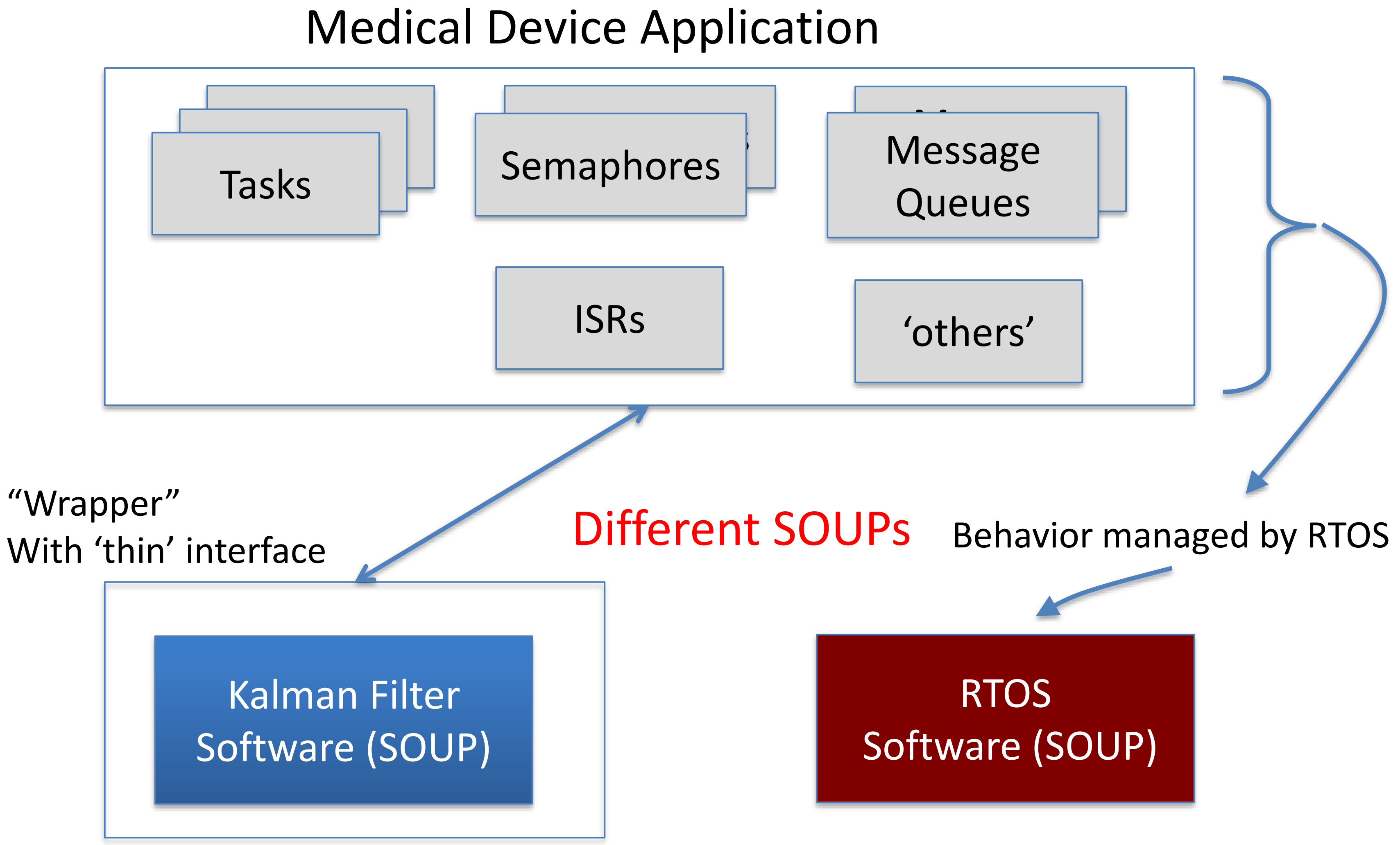
How can we be sure? "enough"

ALARP – As Low as Reasonably Practicable (RISK)

Software of Unknown Provenance (SOUP)



Software of Unknown Provenance (SOUP)



Failure conditions potentially induced by RTOS

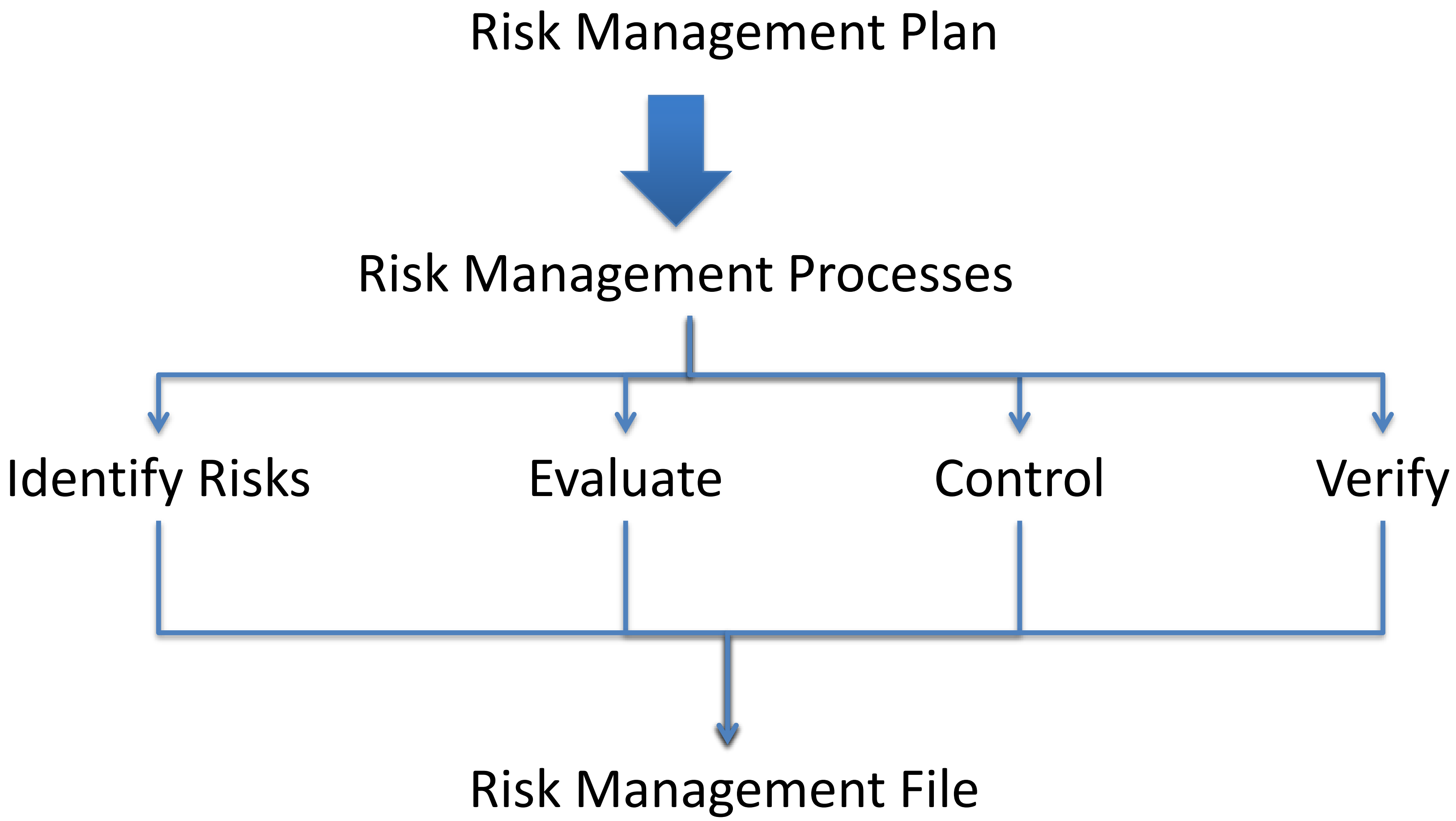
- Data is incorrectly modified.
- Incorrect results are provided to the application.
- Expected results are not provided or are provided past their deadlines.
- User code is not executed as expected (not run, incorrectly run, and incorrectly sequenced).
- Fault conditions are not detected.
- Fault conditions are handled incorrectly.
- False failure conditions are reported.
- Incorrect or untimely response provided by the RTOS to external or user generated events.

Failure conditions potentially induced by RTOS

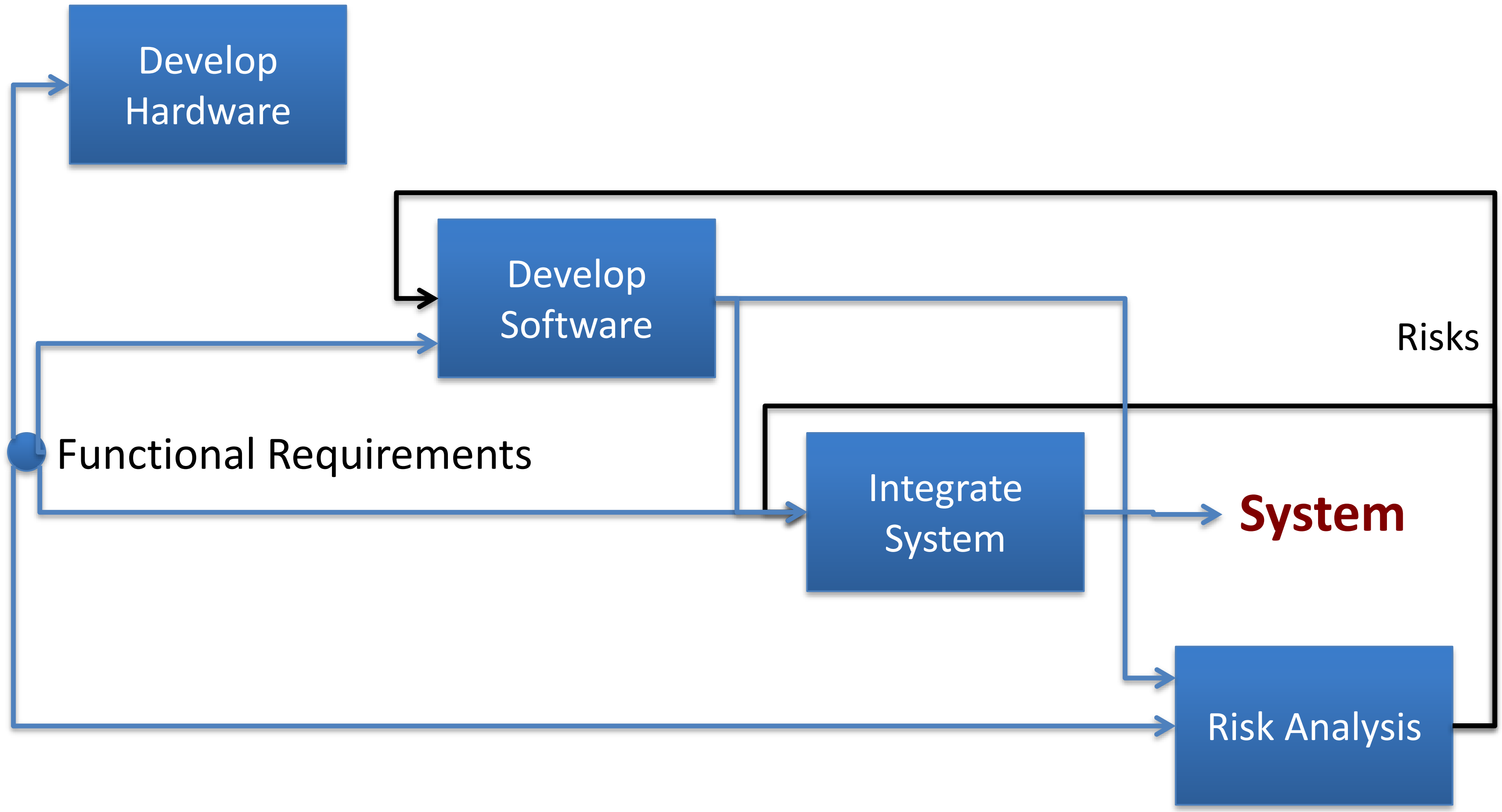
- Data is incorrectly modified.
- Incorrect results are provided to the application.
- Expected results are not provided or are provided past their deadlines.
- User code is not executed as expected (not run, incorrectly run, and incorrectly sequenced).
- Fault conditions are not detected.
- Fault conditions are handled incorrectly.
- False failure conditions are reported.
- Incorrect or untimely response provided by the RTOS to external or user generated events.

Reduce risk by Using Certified RTOS

Managing Risk - ISO 14971



Medical Device – Based on Risk Assessment



System Hazard – Software Class

- System Hazard
  - No Injury – A
  - Non-Serious injury – B
  - Death or Serious injury – C

ISO 14971

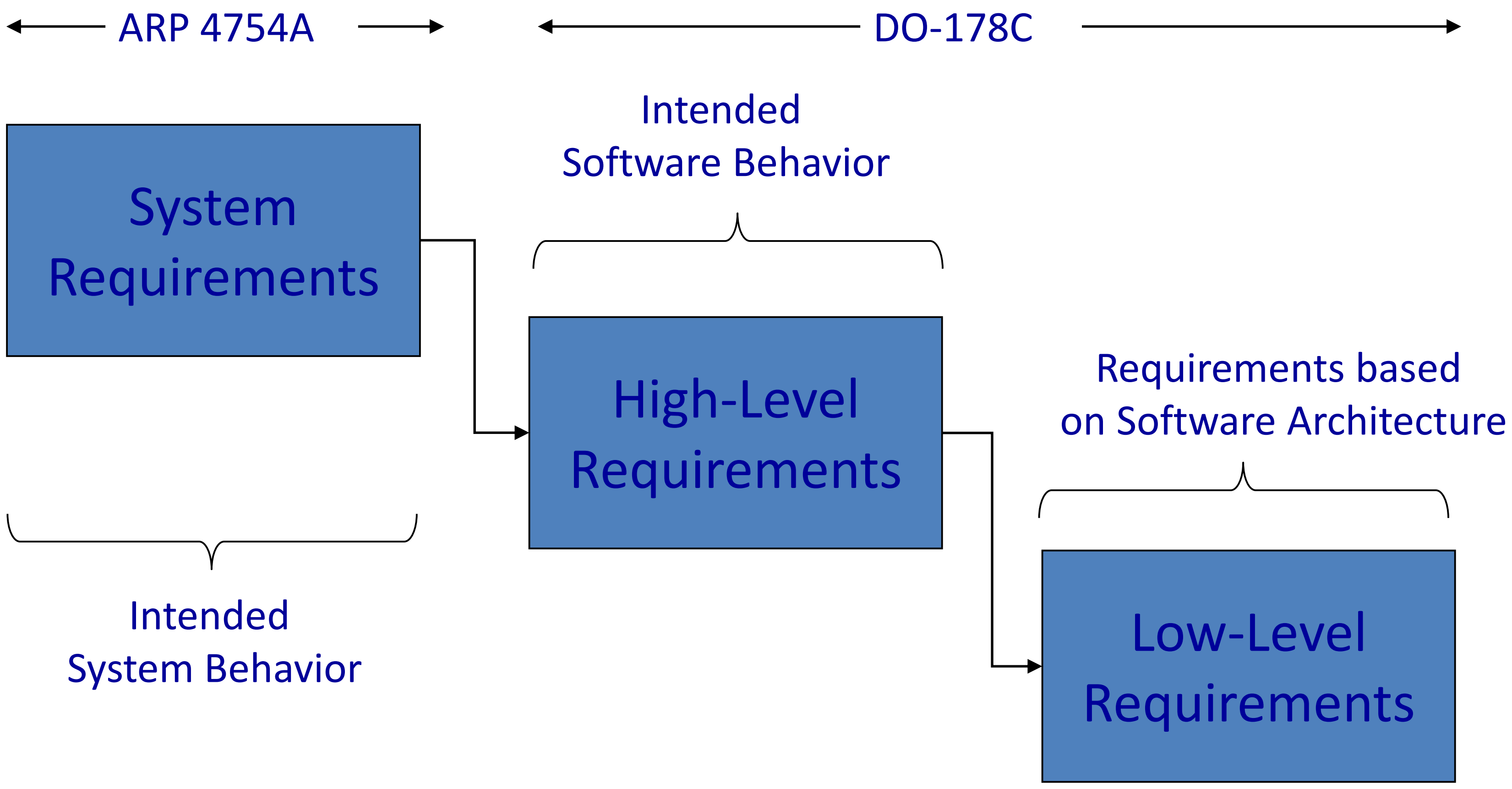
- If failure in Software leads to System Hazard
- Software categorizes as
  - Class – A
  - Class – B
  - Class – C

IEC 62304

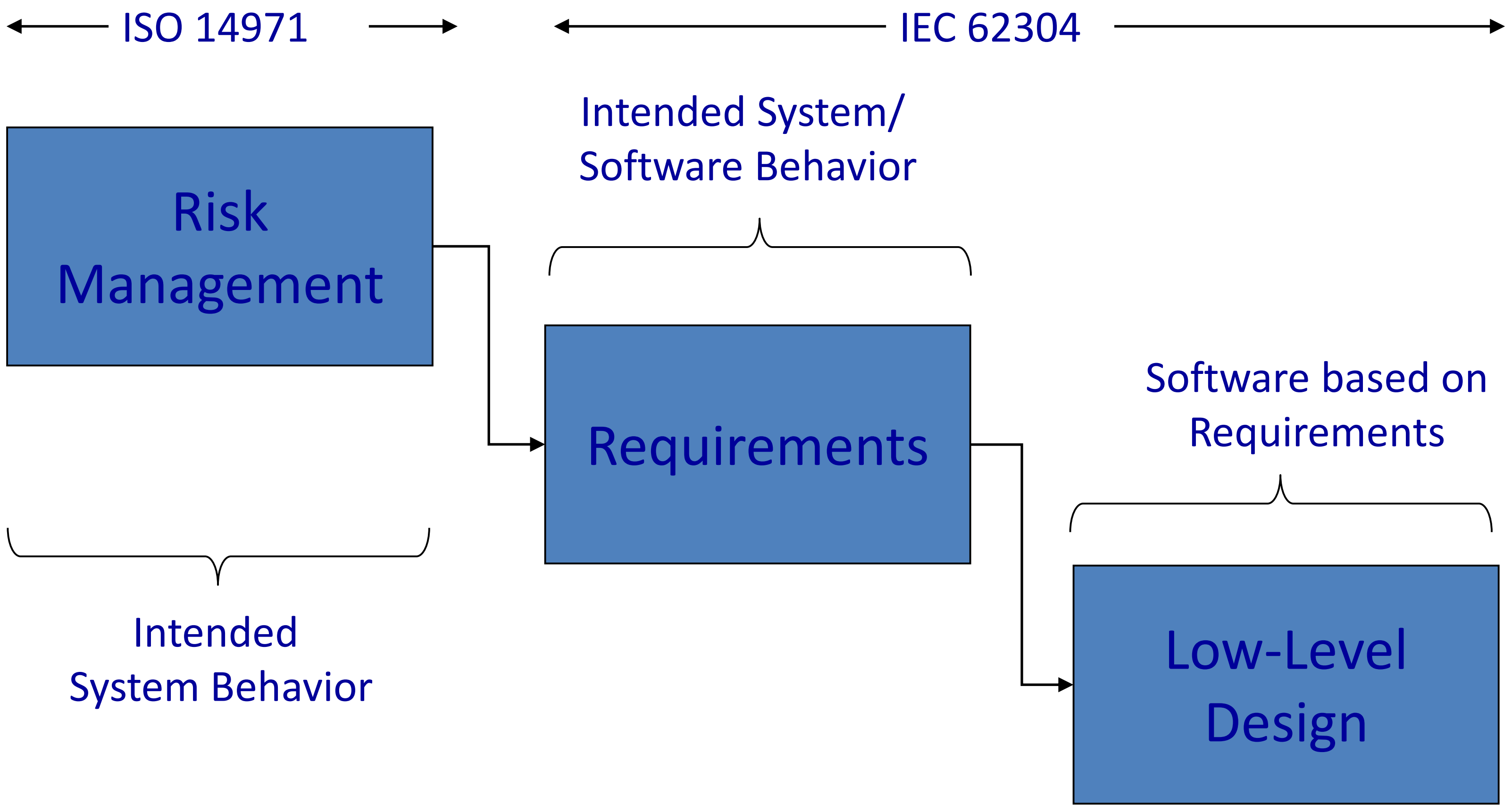
Risk Analysis

- Failure Mode Analysis
- Identify Potential Risks
- Determine Risk Exposure
  - Continuous while operational
  - Frequent
  - Occasional
- Can Software contribute to Hazards
- Can Hazards be mitigated
- Finding potential hazards in Software can be TOUGH!

Requirements Hierarchies – DO-178B



Requirement/Design organization



Parameter Data Items

- Parameter Data Items can be developed and verified separately if certain conditions are met
- The high-level requirements describe how the software uses the parameter data items
- The low-level requirements define the structure, attributes and allowable values of the parameter data items
- Verification should show that every data element has the correct value – or correct value is in equivalence class and boundaries are verified

e.g. Configuration Vectors

Our Experience

- Develop verification evidence using a DO-178C software Lifecycle
- All of the other Lifecycle processes will fall into place.
- Some additional documents required
  - Safety Plan
  - Safety Manual
  - Staff Competency Assessment
    - Actual assessment required not just Resume

More Experience

- Interpretations and negotiations are prevalent in Automotive, Medical and Rail industries.
- TUV were strict on first Verocel certification
  - Plans, procedures and standards approved
  - Subsequent certifications were straightforward
  - "Manufacturing – reject tracking" needed to be addressed (for software)?

Finding and eliminating unintended behavior

- Requirements describe intended behavior
- Software developed against requirements (TRACED)
- Tests written against requirements (ONLY) and (TRACED)
- Software coverage by tests measured
- Any software not covered demonstrates “unintended behavior”
- This is a risk that must be eliminated.

Code Coverage Analysis

- Requirements used to specify intended behavior
- Write tests using Requirements ONLY
  - Normal Range
  - Robustness
  - Equivalence Class
  - Boundary Value
- Run tests and measure how much code was executed
- Assess is non-executed code should not be there

Coverage Analysis not required explicitly by IEC 62304, but

Hard to mitigate “Unintended Functionality risk” without

Coding Standards

- Standards Required – used to show goodness of various properties
- Code Layout
- Code consistency
- Readability
- Avoidance of risky constructs
- Etc.

Code Review

- Verifies Conformance to Standards
- Verifies conformance to review criteria
- Verifies code against intended behavior
  - Low-level Design
  - Low-level requirements

Code Analysis Tools (The silver bullet?)

- Perform consistency checks
- Perform checks against defined coding rules (e.g. MISRA C) to find errors like:
  - Use of variable before initialization
  - Indexing out of bounds (simple cases)
  - Potential deadlock
  - Unreachable code (sometimes)
  - Arithmetic overflow (sometimes)

Good quality step, reduction of potential faults, BUT!

Code Analysis Tools for Certification

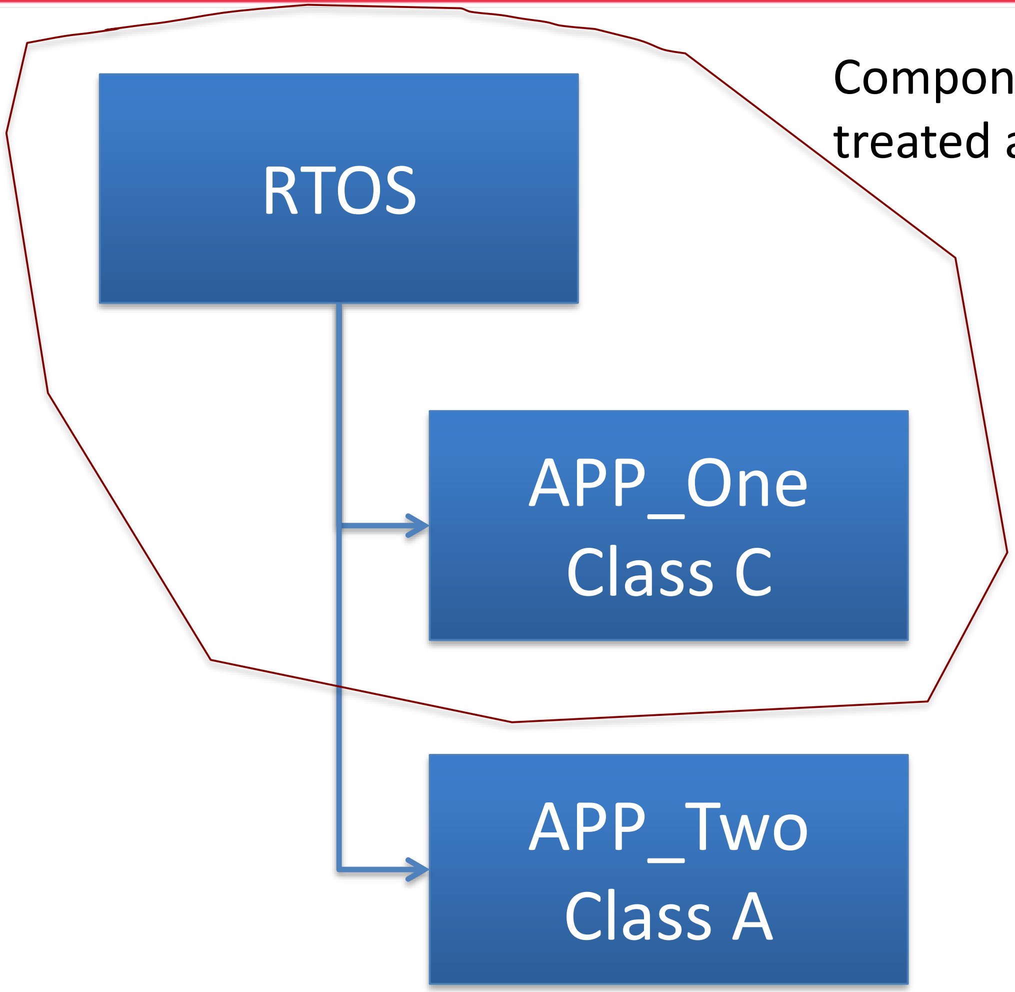
- Checks are incomplete
- Hard to assess what checks must be completed manually
- No analysis against intended behavior
  - Low-level design
  - Low-level requirements

Only partial credit may be taken!

Configuration Management Processes

- Identification
  - Versions
  - Baseline
- Change Control
  - Documented process
  - Change Control Board
- Configuration Accounting
  - History tracking
  - Access Controls

Segregation of Software Components



Components can be segregated and treated as Class C on same computer.

Required

- ◆ Fault Containment
- ◆ Memory Partitioning
- ◆ Time Partitioning

App-Two cannot adversely affect Class C software

Audit Risk

- Scenario 1
  - Prepare Verification evidence on paper
  - Instruct Engineers to give YES/NO answers
  - All information available, but difficult to locate.
  - Auditor cannot make good assessment
  - Applicant passes with substandard/incomplete audit.

Not the Verocel Way!

The Verocel Approach

- Plans, QA records, CM-data, and Certification data:
  - Hyperlinked data – easy to find and trace
- All data open and put on DVD-ROM
  - Auditors can trace their own copies
- All data extracted from Traceability database, and CM repository

Auditors approach

Developers/Certifiers

- Develop Plans and Standards
- Develop Certification evidence using P&S
- QA Checks that P&S are followed

Auditors

- Review Plans and Standards
- Sample Cert evidence
- Check QA Audits

✓ If a controlled process was used consistently
✓ And Sample is good
✓ Then we can trust the rest of the certification evidence and the software