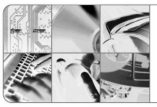


metrowerks

Building Quality into Safety Critical Applications

組み込みソフトウェアの性能、品質の向上 – セーフティクリティカル産業から学んだ事例 教訓

Nat Hillary
Field Applications Engineer
nath@metrowerks.com
+1 (425) 487 - 5985

2004年5月 30日

MOTOROLA

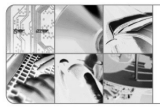
"You can't control what you can't measure"
Tom DeMarco

metrowerks

Agenda

- **Introduction**
- **Building Safety Critical Software**
  - A focus on the Avionics approach
- **Conclusion**

metrowerks



metrowerks

Introduction

MOTOROLA

2004年5月 30日

Safety Critical Systems – a definition

- **A system is 'safe' when it does not endanger human life or the environment**
- **A safety critical system is one whose action, or inaction, may cause death, dismemberment or injury**
- **Examples:**
  - Brake-by-wire system
  - Cardiac Defibrillator
  - Fly-by-wire system

metrowerks

Implementing a Safety critical system

- **Safety critical computer systems have both hardware and software components**
- **Overall system must be reliable and fail safe**
- **Reliability:**
  - Measure of how long system must operate correctly without failure
- **Fail safe:**
  - Does NOT mean that system is safe from failure!
  - Means that should a failure occur, the system design ensures that the failure does not affect safety, i.e. system should FAIL SAFELY

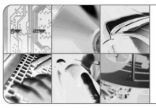
metrowerks

Designing Reliable Hardware & Software

- **Hardware**
  - Techniques and criteria for building reliable hardware well understood (e.g. use of redundancy)
  - Common for industry regulations to define specific design, implementation and test criteria for hardware
- **Software**
  - Similar criteria for software rarely exist
  - As our dependency on computers increase, so does our sensitivity to Computer Error

Industry Response to proliferation of computer systems

- **Growing number of guidelines addressing safety critical software development, e.g.:**
  - RTCA DO-178B "*Software Considerations in Airborne System and Equipment Certification*"; Aerospace
  - MISRA "*Development Guidelines for Vehicle Based Software*", Automotive
  - FDA "*General Principles of Software Validation; Final Guidance for Industry and FDA Staff*", Medical
- **Common themes in approach used by most safety critical industries**



metrowerks

Building Safety Critical Software

A focus on the Avionics approach

MOTOROLA

2004年5月 30日

General Approach Shared by Safety Critical Industries

- **Producing high quality software requires high quality processes**
- **Have a quality management system (e.g. ISO 9000)**
- **Perform hazard analysis to determine safety integrity level**
- **Define development process appropriate for safety level, e.g.:**
  - H/W & S/W Architectural Design (level & type of redundancy, etc.)
  - Choice of implementation language
  - V & V activities
  - Choice of tools
- **Make product measurements**

Unique aspects of Avionics approach

- **Approach to V&V identified to FAA prior to development**
- **Process documented throughout**
- **Software lifecycle data submitted to authorities to obtain certification 'credit'**
- **Software verification activities are described independently from software verification tools**
- **Intent is that focus is on verification process first, and process automation second**
- **Three lessons to learn from this approach**

Lesson One

- **Execute functional tests as normal**
- **Repeat, making CodeTEST coverage measurements**
- **Analyze coverage results, and for functions where coverage < 100%, determine cause:**
  - Missing requirements?
  - Missing test cases?
  - Dead or unreachable code?
- **Improve test effectiveness by updating requirements, test cases or code**
- **Repeat**

The state of industry …

- **In summarizing his findings from working with many companies, Richard Bender revealed the following figures:**
  - Current best practice in industry
    - Code Coverage > 70%, with 95% of bugs being found before shipment
  - Average test results before shipping
    - Code Coverage < 35%, with > 25% of bugs being found in the field

Ref: *Development Dilemmas and the SEI Model – case studies in Software Process Development*, presented by Chuck House to the 9th IEEE International SoftwareQuality Week, San Francisco May 23rd, 1996

Lesson Two

- **There is no "one size fits all" coverage analysis metric – choice of coverage is based on how safety critical software is**

S/W Level	Failure causes system to fail, resulting in …	Coverage Req'd
A	catastrophic failure for aircraft	MCDC, DC & SC
B	hazardous/severe failure condition for aircraft	DC & SC
C	Major failure condition for aircraft	SC
D	Minor failure condition for aircraft	N/A
E	No effect on aircraft	N/A

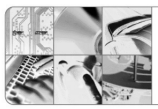
Lesson Three

- **To have high quality code, you need high quality tools**
- **Tool qualification process used in Avionics to ensure that a tool provides confidence at least equivalent to processes eliminated, reduced or automated**

Golden Rule

- **"Automation is a great idea. To make it a good investment, as well, the secret is to think about testing first and automation second."**

Bach, James: Testing Automation Snake Oil, www.satisfice.com/articles/test_automation_snake_oil.pdf



metrowerks

Conclusion

Building quality into safety critical systems

- **Safety critical industries publish software development guidelines seeking to ensure systems meet specific reliability objectives, and they are fail safe**
- **Challenge is to build-in software reliability by reducing or eliminating defects using advanced testing and analysis techniques**
- **Much can be learned from Software V&V Techniques used by avionics industry**
- **Use of Coverage Analysis early and often in development process has demonstrated significant impact on reducing defects**
- **Coverage Analysis does not replace other good processes and practices, but is added and integrated with them**

Not Just Aerospace …

- **A large system vendor on east coast of US introduced Coverage Analysis early and often in their development, resulting in:**
  - 50% reduction in bugs found service
  - 95% reduction in system test time!
  - 23% reduction in total development time

Ref: ***Development Dilemmas and the SEI Model – case studies in Software Process Development*****, presented by Chuck House to the 9th IEEE International SoftwareQuality Week, San Francisco May 23rd, 1996**

Questions?

www.metrowerks.com

or please stop by our booth!

References

- **RTCA/DO-178B "*Software Considerations in Airborne System and Equipment Certification*", RTCA Inc., Washington D.C., December 1992**
- **RTCA/DO-248B "*Final Report for Clarification of DO-178B 'Software Considerations in Airborne Systems and Equipment Certification'*", RTCA Inc., Washington D.C., October 2001**
- **FAA Order 8110.49 "*Software Approval Guidelines*", Federal Aviation Administration, June 2003**
- **Draft IEC 1508 "*Functional safety: safety related systems*", Geneva: International Electrotechnical Commission (IEC ref 65A Secretariat 123), June 1995**
- ***"An Investigation of the Therac-25 Accidents"*, Nancy Leveson, Clark S. Turner, IEEE Computer, Vol. 26, No. 7, July 1993, pp. 18-41**
- **"*Testing Computer Software*", Cem Kaner, et al, International Thomson Computer Press, 1993**
- **"*Testing Automation Snake Oil*", Bach, James, www.satisfice.com/articles/test_automation_snake_oil.pdf**
- **"*Emerging Software Best Practices and how to be Compliant*", Rivett, Roger S., Rover Group Ltd., www.misra.org.uk/papers/EAEC97.PDF**