Why We Need Something New in the Automotive Industry

Dr. Qi Van Eikema Hommes
April 19, 2012
Henry Ford in his first car, the Quadricle, built in 1896
Let The Robot Drive

Wired, Feb 2012, http://www.wired.com/magazine/2012/01/ff_autonomouscars/all/1
Automotive Systems Today and Tomorrow

- Cyber Physical Systems - complex embedded devices networked to control physical hardware components.
- Software intensive.
- Automating many human tasks.
- The development teams are multidisciplinary and globally distributed.
Quality Problem With no Component Failure

- Trouble-Not-Identified Engine Control Module warranty problem.
- No component failure was found.
- Insufficient resource to conduct exhaustive bottom-up testing, after the product was already released to market.
- Many such quality problems are never resolved.
Toyota Unintended Acceleration

The Detroit News

April 6, 2010

Toyota faces $16.4M fine for hiding safety defect

Proposed penalty is largest ever sought by NHTSA officials

DAVID SHEPARDSON
Detroit News Washington Bureau

Washington -- Federal safety regulators are seeking to fine Toyota Motor Corp. $16.4 million -- the largest ever penalty against an automaker -- for failing to disclose problems with sticky accelerator pedals in a timely manner.

Wall Street Journal

NASA to help probe unintended auto acceleration

DAVID SHEPARDSON
Detroit News Washington Bureau

Washington -- The U.S. Transportation Department will launch two major investigations to discover whether vehicle electronics or electromagnetic interference are to blame for unintended vehicle acceleration incidents.
Typical Decomposition Scheme

• **Physical**: usually stated as systems, subsystems, subassemblies, parts
  – Car systems and subsystems include seats, engine, suspension, steering

• **Organizational**: Usually stated as divisions, departments, groups, etc.
  – Powertrain department, Research and Development division, etc.

• **Process**: usually stated as phases of the product development process.
  – Concept development, detailed design, etc.
Example of a Vehicle Engineering
Physical Decomposition

Vehicle

Chassis

Powertrain

Body

Engine

Transmission

Control

Valvetrain

Cranktrain

Head
Organization Decomposition

The Effect of Decomposition

• Quality and safety = component failure prevention
  • Failure: Not performing intended function

• Quality and Safety Engineering = Reliability Engineering

• Component failures are random hardware failures
  – Not useful for complex software system
  – Not useful for social systems

• Bottom-up hazard analysis based on linear chain-of-events model, ignoring systemic factors.

• The reality: many unresolved quality problems.
An Example of System Interactive Complexity: The Powertrain Control Software System

- 1 production-level software
- 117 software modules (red dots)
- 1423 interactions (black lines)
- 39 such production software releases per year
- <2 weeks per release

Hommes, DETC2008-DTM-49140
We Rely Heavily on Experts’ Tacit Knowledge to Handle System Interactions and Integration

![Bar Chart]

- **CVC**: 97% Experience, 3% Document
- **JNJ**: 84% Experience, 16% Document
- **Ford**: 70% Experience, 30% Document

©4/19/12 Qi D. Van Eikema Hommes
ISO 26262 Functional Safety for Road Vehicle

• The first comprehensive standard that addresses safety related automotive systems comprised of electrical, electronic, and software elements that provide safety-related functions.
• Adaptation of IEC 61508 to road vehicles
• Influenced by ISO 16949 Quality Management System
# General Structure of ISO 26262

## 1. Vocabulary
- 1-5 Overall safety management
- 1-6 Safety management during item development
- 1-7 Safety management after release for production

## 2. Management of functional safety
- 2-5 Overall safety management
- 2-6 Safety management during item development
- 2-7 Safety management after release for production

## 3. Concept phase
- 3-5 Item definition
- 3-6 Initiation of the safety lifecycle
- 3-7 Hazard analysis and risk assessment
- 3-8 Functional safety concept

## 4. Product development: system level
- 4-5 Initiation of product development at the system level
- 4-6 Specification of the technical safety requirements
- 4-7 System design
- 4-8 Item integration and testing

## 5. Product development: hardware level
- 5-5 Initiation of product development at the hardware level
- 5-6 Specification of hardware safety requirements
- 5-7 Hardware design
- 5-8 Hardware architectural metrics
- 5-9 Evaluation of violation of the safety goal due to random HW failures
- 5-10 Hardware integration and testing

## 6. Product development: software level
- 6-5 Initiation of product development at the software level
- 6-6 Specification of software safety requirements
- 6-7 Software architectural design
- 6-8 Software unit design and implementation
- 6-9 Software unit testing
- 6-10 Software integration and testing
- 6-11 Software verification

## 7. Production & Operation
- 7-5 Production
- 7-6 Operation, service and decommissioning

## 8. Supporting processes
- 8-5 Interfaces within distributed developments
- 8-6 Overall management of safety requirements
- 8-7 Configuration management
- 8-8 Change management
- 8-9 Verification
- 8-10 Documentation
- 8-11 Qualification of software tools
- 8-12 Qualification of software components
- 8-13 Qualification of hardware components
- 8-14 Proven in use argument

## 9. ASIL-oriented and safety-oriented analyses
- 9-5 Requirements decomposition with respect to ASIL tailoring
- 9-6 Criteria for coexistence of
- 9-7 Analysis of dependent failures
- 9-8 Safety analyses

## 10. (Informative) Guidelines on ISO 26262

Source: ISO 26262

©4/19/12 Qi D. Van Eikema Hommes
Strengths

• Emphasizing safety management and safety culture
• Prescribes a systems engineering process
• Departure from safety as an after-thought:
  – IEC 61508: safety function
  – ISO 26262: provides the framework and vocabulary for hazard elimination in the first place
    • Systems engineering framework
    • Safety measure vs. safety mechanisms
Suggestions for Improvements

• Safety measure is not clearly explained in the document, while Safety Mechanism is explained in detail throughout the document.

• The standard may want to add a section in Part 1 to further clarify the departure from IEC 61508’s design philosophy.
Reliability Engineering Methods in ISO 26262

• **Hardware Architecture Metrics**—Based on random failure of components.
• **Failure Modes and Effects Analysis (FMEA)**
• **Fault Tree Analysis (FTA)**
• **Safety Case Approach**
  – Confirmation bias
  – Independent reviewers are less familiar with the design
  – The use of Quantitative Risk Assessment
• **Investigate the effectiveness of STPA and how to integrate it with the standards to provide higher safety assurance.**
Software Safety

• Follows software system engineering process
• Promotes good software architecture practices
• Best practices in software design
• Addresses hardware failure
• On Par with other software safety standards such as DO-178

Comments:
• Unlike hardware, software does not fail.
• Software faults are due to design errors, but the standard does not offer a way to identify design errors that can cause hazard.
• Good systems engineering process and software architecture design are necessary but not sufficient to ensure system safety.

©4/19/12
Qi D. Van Eikema Hommes
Summary

• Automotive systems have changed—more complex, software intensive, more automation.

• Reductionist approach is no longer adequate.

• ISO 26262 is our latest effort to address our new challenges. It can be improved by incorporating STPA.
Proposal: Research Consortium on Automotive Functional Safety

• Industry – Government – Academia Collaboration
• Funded research projects
  – Develop a scientific framework for automotive electronics safety engineering
  – Develop a non-proprietary test bed that reflect the real world challenges
  – Educate future engineers
• Shared learning among members to
  – Improve design for safety
  – Improve industry standards
  – Support safety regulation

©4/19/12

Qi D. Van Eikema Hommes
Thank you!

Questions?