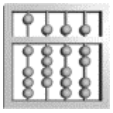
Challenges in Automotive Software Engineering: From Demands to Solutions

Manfred Broy

Technische Universität München
Institut for Informatik



Points of Focus

- innovation by embedded automotive software
  - ◇ risks and potential
  - ◇ requirements
  - ◇ domain architectures
- managing complexity
  - ◇ models
  - ◇ process
  - ◇ tool support
- costs
- success factors
  - ◇ connectivity
  - ◇ reliability and safety
  - ◇ security
  - ◇ supplier management

IT Trends

- computing power and performance increase
  - ◇ hardware - higher performance
  - ◇ software - more functionality ⇒ complexity
- reusability of solutions
- interoperability
- connectivity
  - ◇ network infrastructures
  - ◇ logical dependencies ⇒ complexity
- convergence
- dependability (safety and security)
- (Quasi)standards

Automotive trends and the significance of software

- software as innovation driving force
  - ◇ new and more functionality
  - ◇ new and cheaper technical solutions
- customer orientation
  - ◇ personalisation
  - ◇ individualisation
- market trends
  - ◇ interoperability
  - ◇ dependability and privacy
  - ◇ comfort by additional services
- new business cases
  - ◇ new business partnerships
  - ◇ additional services
  - ◇ after sales business

Challenge: Quality management - model driven SE

- requirements
  - ◇ validation
- component architecture
  - ◇ decomposition
  - ◇ interfaces
  - ◇ integration
- component reliability assurance
  - ◇ test
  - ◇ model checking
  - ◇ Verification
- error management
  - ◇ in the development process
  - ◇ in the software product

Key goal: reliability

- aspects of reliability
  - ◇ correctness
  - ◇ simplicity - user transparency
  - ◇ availability
  - ◇ robustness (fail safe, tolerance against errors in environment or user handling)
- methods for reaching reliability - built in quality
  - ◇ requirements: risk analysis and model orientation
  - ◇ design: error handling
  - ◇ realisation: quality assurance - verification
  - ◇ maintenance: error documentation and software updates

Challenge: requirements

- getting requirements right
  - ◇ design space exploitation
  - ◇ user expectations capture
  - ◇ implications estimate
  - ◇ dependability analysis
  - ◇ risk management
  - ◇ precise requirements documentation
- overall requirements concept work out
- systematic requirements management
  - ◇ management of change
  - ◇ tracing
- comprehensive view (costs, risks, implied problems, market advantages)

Key factor in requirements engineering

- Model based RE
- enlarged design spaces
- RE process
  - ◇ user participation
  - ◇ abstraction and structuring
  - ◇ model construction
- Requirements structuring
  - ◇ functional requirements
  - ◇ non-functional requirements
- requirements understanding
- requirements rationale

Challenge: dependability and connectivity

- connectivity of the functions
  - ◇ functional dependability
  - ◇ feature interaction
  - ◇ connectivity to the outside
- new functions by functions composition

consequences for the development

- overall architecture
  - ◇ functions/Use cases
  - ◇ component decomposition
  - ◇ interdependabilities
- more concentration on the integration

Challenge: error management in software

- risk analysis
  - ◇ which errors can occur
  - ◇ which consequences
- error modelling
  - ◇ how look the errors precisely like
- error management
  - ◇ how on errors react
  - ◇ how minimal an functionality maintain
- error protocols and documentation
- error management in the maintenance
  - ◇ diagnose
  - ◇ debugging

Challenge: error tolerance

- redundancy
  - ◇ hardware redundancy
    - protection of hardware defects
  - ◇ software redundancy
    - idea: software multiple versioning
    - problem: software hat no material defects but logic errors
- defensive programming
  - ◇ additional check
  - ◇ avoiding error intensive and critical constructions
- error management
  - ◇ central error tracing
  - ◇ error recovery

Challenge: Cost management and benchmarking

- cost factors
  - ◇ development costs
  - ◇ production costs
  - ◇ maintenance costs (maintenance costs in the field)
  - ◇ implied risks
- interdependencies hardware/software costs
- benefits
  - ◇ improvement an functionality
  - ◇ improvement an technical quality
  - ◇ cost decrease
  - ◇ image/marketing

cost dependencies insufficiently understood

Key issue: architecture

mile stones of an embedded system

- domain - architecture the functions (use cases)
  - ◇ tree/hierarchy the use cases
  - ◇ variants and dependability
- component decomposition - design architecture
  - ◇ logic of co-operation and interaction
- software architecture
  - ◇ software components
  - ◇ interfaces
  - ◇ resource demands
- platform - hardware and software
  - ◇ Power and resources

Challenge: hard- and software-development

- the role the software and the hardware
  - ◇ model
  - ◇ model of the dependencies
  - ◇ optimisation
- software - functionality
- hardware - resources
- open systems - portability and extendibility
  - ◇ layered architectures
  - ◇ platform
  - ◇ software architectures
  - ◇ flexible hardware/software-mapping
- cost optimisation

Challenge: productivity increase - reusability

The increasing investment costs for software demand investment defence:

- systematic reusability of development results

Difficult themes:

- experiences with reusability in the software techniques ambiguous
  - ◇ ad hoc reusability
  - ◇ systematic reusability
- product lines

Key issue: Product lines

Systematic

- domain architectures
  - ◇ requirements
- system architectures
  - ◇ design
- software architectures
  - ◇ platform independent software components
- hardware architecture
- hardware/software codesign
  - ◇ deployment mapping

Challenge: interface management supplier – OEM – customer

key problems:

- requirements imprecise
- interfaces not systematic specified
- no defined process supplier/OEM

approach

- precise technical documentation
- seamless process
  - ◇ integrated QS
  - ◇ common process/product agreement and understanding

problem area: flexibility and co-ordination

Challenge: IT dependability - security

With connectivity growing security requirements:

- identification
- authentication
- authorisation
- privacy

In respect on

- user
- maintenance
- software infrastructure

Conclusion

Automotive software remains an area of high dynamics:

- high potential
- many fields of problems
- fast technology progress
- new application fields
- new technologies
- fast learning processes needed
  - ◇ market
  - ◇ organisation
  - ◇ processes
  - ◇ reusability
  - ◇ tools