Best Practice Report – Challenges in Software Testing

InnoTesting 2018

Clemens Passeck, Philotech GmbH
Founding year 1987 in Munich

Business: Engineering & Consulting Services

Industries:
- Aviation
- Space
- Automotive
- Transportation
- IT & Electronics
- Defence
- Renewable Energy

Management: Hartwig Dirscherl, CEO

Supervisory Board: Horst Steinberg, Chairman (5 industry managers/experts)
Philotech Overview
Competencies - Know-how for our customers

Systems Engineering Services
- Systems Specification, Design, Development, Integration and Test
- Configuration Management
- Support for Certification

Configuration Management
- Product Lifecycle Management
- Change Management
- Requirement Management
- Planning, Documentation & Control of Product Configuration
- Software Version Control

Support Engineering
- Maintenance Planning
- Integrated Logistic Support (ILS), Logistic Support Analysis (LSA) & Life Cycle Management (LCM)
- RAMSST Analyses (Reliability, Availability, Maintainability, MSG-3, Safety, Security, Testability)
- Technical Documentation, Pricing and Spare parts

Verification & Validation
- System / Software Test Consulting & Engineering
- IVV (Independent V&V)
- Trainings and Seminars

Software Engineering
- Embedded, real-time software
- Safety-critical software
- Visualization & Simulation, HMI
- Benchmarks and assessments
- Agile methods and tools
- Databases (technical)

Safety and Security
- Safety and Security Concepts
- Support and conduct Safety Assessment Activities
- Guidance and Training
- Certification Support

Philotech Academy: Training & Seminars

© Philotech Group

... the name is our philosophy.
● Today software is an integrated part of almost all complex systems

● For example software is involved in a lot of aircraft functions:
  ■ Fly by wire
  ■ Glass cockpit
  ■ AFDX Backbone
  ■ Flight controls
  ■ Engine control unit
  ■ Cabin systems
  ■ Passenger entertainment
  ■ …

● The systems and features which are implemented in Software are increasing and thus, the software complexity is growing
Overall quality in avionics
Airliner Statistics

Airliner Accidents Per 1 Million Flights 1977-2016

Source: Harro Ranter (Aviation safety Network)

Statistics are based on all worldwide fatal accidents involving civil aircraft with a minimum capacity of 14 passengers, from the ASN Safety Database
https://aviation-safety.net
Overall quality in avionics
Airliner Statistics

Fatal Airliner Accidents Per Year 1946-2016

Statistics are based on all worldwide fatal accidents involving civil aircraft with a minimum capacity of 14 passengers, from the ASN Safety Database https://aviation-safety.net

Source: Harro Ranter (Aviation safety Network)

. . . the name is our philosophy.
Overall quality in avionics
Passenger Transportation Risk

Deaths per 100 million person kilometres

- Rail: 0.035
- Civil aviation: 0.035
- Ferry: 0.025
- Bus and coach: 0.07
- Car: 0.7
- Cycl: 5.4
- Foot: 6.4
- Motorcycl: 13.8
- Road (Total): 0.95

The primary guidance the Civil Aviation Authorities (FAA, EASA..) use to approve software for use in aircraft.

Also used in other safety-relevant areas: medical, military, etc.

Airborne software products must comply with stated objectives of DO-178x.
Critical situations can be caused by software problems in all industrial areas

AEROSPACE

AUTOMOTIVE

RAILWAY

MEDICAL

... 

DO-178 blue-print for all major SW-safety standards

- **RTCA DO-178**: Software Considerations in Airborne Systems and Equipment Certification
- **RTCA DO-278A**: Guidelines for Communications, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance
- **ISO 26262**: Road vehicles – Functional safety (Anpassung der IEC 61508 an die spezifischen Gegebenheiten im Automobilbereich)
- **IEC 61508**: Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme

Timeline:

- **RTCA DO-178**: 1980
- **DO-178A**: 1985
- **DO-178B**: 1992
- **IEC 61508**: 1998
- **ISO 26262**: 2011
- **DO-178C**: 2011
- **DO-278A**: 2011
Standards
DO-178C and its Supplements

DO-178C
Software Considerations in Airborne Systems and Equipment Certification

DO-278A
Guidelines for Communications, Navigation, Surveillance, and Air traffic Management (CNS/ATM) Systems Software Integrity Assurance

DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A

DO-332 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A

DO-333 Formal Methods Supplement to DO-178C and DO-278A

DO-330 Software Tool Qualification Considerations (covers DO-178C and DO-278A)

DO-248C Supporting Information for DO-178C and DO-278A

... the name is our philosophy.
A simple function, which has as an input three whole number values which are not dependant on each other, needs to be tested.

The input values are 16-Bit-Integer numbers, can also take on different values.

Combinations:

\[2^{16} \cdot 2^{16} \cdot 2^{16} = 2^{48}\]

Does one want to test all these Combinations, How long does it take at 100 000 tests per second?

at 281 474 976 710 656 test cases: 90 years!
1. Exhaustive testing is impossible
2. Testing shows the presence of defects
3. Testing has to start early in the process
4. Defect clustering
5. Repetitive execution of the same tests, produces no new Information
6. Testing is context dependent
7. A Software without errors against its specification is not always usable or what the customer expects
Grouping of input and output values in equivalence partitions so that for the values of a common equivalence partition the same system behaviour (detecting and non-detecting of the same failures) will be accepted.

Assumption 1: If a failure is detected by a value in one partition, then every other value in the same partition will detect this failure.

Assumption 2: A value from one partition does not detect a failure, then every value in the same partition does not detect a failure either.
Assumption: program behaviour is more likely to be erroneous at boundary regions (as a transitions to other equivalence partitions) than in regions that are distant thereof.

Often boundary regions of branch- and loop-conditions are wrongly implemented.

With a high possibility of detection of failures during aimed testing at boundary region.

In-/valid boundary values: boundary values of an in-/valid equivalence partitions.

Boundary values are used as input values.
SW Test
General things to keep in mind

- SW shall be requirements based (Black Box)
- A test level shall be defined, that has the focus on validation (did we build the right system) and not only on verification (did we build the system right)
- Tests should be executed on the target, as far as possible
- Process steps have to be defined and acknowledged (define what should be done and how it should be done before the work is started and adapt these plans to new needs)
- It has to be ensured, that normal range and robustness tests are developed and executed
- Structural coverage is used as a measure to evaluate the quality of the tests, requirements and to locate dead code
- Artefacts have to be reviewed for their correctness (also tests, …)
- Quality department has to ensure that all artifacts are generated as expected and that they are developed as recommended in processes
Engineering, Consultancy and customized solutions along the entire lifecycle of complex systems. Our customers can rely on our expertise and availability.