Formal Methods for Certification

Certification of Avionic Systems

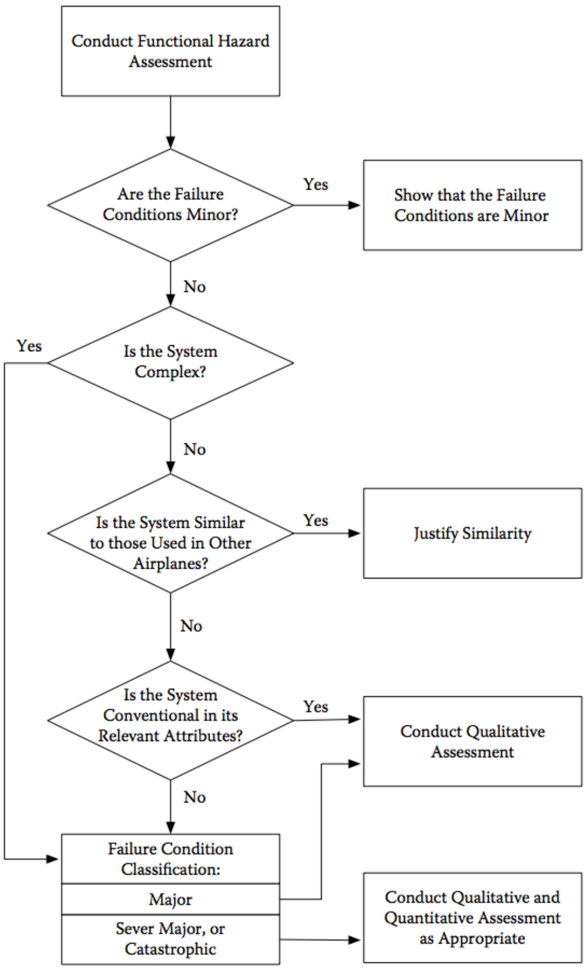
- certification ≡ the process by which a system is demonstrated to comply with a set of regulations and standards ⇒minimum safety requirements
- avionics ⇒“airworthiness”
  - *standard certificate* standard operations
  - *special airworthiness certificates* aircraft with special permissions, e.g. experimental aircraft
- *type certificate* TC: all airplanes of this family are ok
- *issue of interpretation*: clarifications are issued
  e.g. Advisory Circular: qualitative techniques that can be used to demonstrate compliance

  design appraisal qualitative appraisal of integrity and safety of design with emphasis on failure conditions

  installation appraisal qualitative appraisal of integrity and safety of installation

  failure mode and effect analysis

  fault tree or reliability block diagram analysis
- *issue of new technologies*: “special conditions” can be negotiated



So Many Standards, So Little Time

- research in the past often in single countries
- regulations may be there to protect local economy
- *common aspects*

  prescription level some documents are compulsory, other recommended (best practices)

  reference sector different markets, different (safety requirements), different traditions, different engineering practices/ constraints

  Scope standards for software, hardware, complex systems

The ECSS System of standards

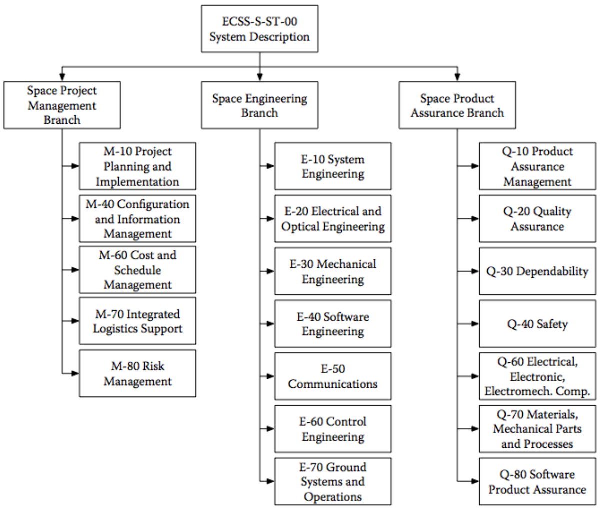
- standards of the european space agency
- no legal standing, but covers all aspects of space engineering

  M *space project management branch* management

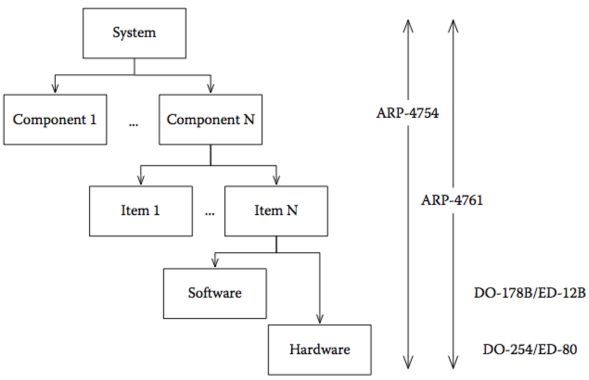
  Q *space product assurance* quality assurance, system safety

  E *space engineering branch* systems engineering, software engineering

  S *standards*



Avionics Reference Standards



- standards cover different aspects of system development

  Highly-Integrated or Complex Aircraft Systems

  Safety Asessment on Civil Airborne Systems and Equipment

  Software Considerations

  Airborne Electronics Hardware

ARP 4754

- certification aspects of system that implement aircraft-level functions (e.g. fly-by-wire system)
- document focuses on development and certification of complex systems to include

system development top-down development

- describes how the previously mentioned processes integrate into coherent development process
- iterative process, increasing levels of refinement
- runs parallel and integrates with safety assessment activities
- main development activities:

  aircraft-level functional requirements high-level requirements and functions are defined

  allocation of aircraft functions to systems functions of previous step are allocated to systems

  development of system architecture architecture development to implement functions and satisfy safety requirements

  allocation of item requirements to hardware and software

  system implementation
- *support processes*
  * certification coordination
  * safety assessment
  * requirement validation
  * implementation validation
  * configuration management
  * process assurance

certification process and coordination methods and techniques to demonstrate compliance of system with certification authorities

1. *certification planning* allows to conciliate development constraints, system complexity and certification activities
2. *agreement on the proposed means of compliance* allows to accommodate special conditions
3. *Compliance substantiation* implementation of certification plan

requirement determination and assignment of development assurance levels how allocation of critical safety requirements determines assurance levels of components

- five levels of criticality
- techniques for safety improvement

  partitioning isolate critical components ⇒lower levels of assurance for non-critical components

  redundancy

  monitoring redundancy with hot or cold standby
- redundancy:

  similar design redundant design is based on same physical phenomena and component

  dissimilar independent design redundant design is based on different physical phenomena and components

  dissimilar dependent design redundant design based on same physical phenomena, but other components
- redundant design ⇒reduced chance for complete failure ⇒lower level of assurance for single components (if dissimilar)
- human intervention ⇒lower level of assurance may be possible

safety assessment process overview over activities and techniques

- if assurance level are A to C, probability below certain value must be shown

validation of requirements process and methods to demonstrate that the requirements are complete and specify the "right system"

validation planning methods to demonstrate compliance are determined

execution of checks actual validation activities

validation of assumptions all assumptions during development are listed and validated

- traceability is important

implementation verification demonstrate, that implementation implements all requirements

inspections and reviews often based on checklists

analysis evidence of compliance through detailed examination

testing demonstrate implementation of requirements, maybe demonstrate, that undesired functions are not implemented

service experience already certified components are used to their specifications

configuration management describe activities to be performed to ensure coherent documentation

process assurance describe activities necessary to ensure that development and support process have been dutifully applied and executed

ARP 4761

- "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment"
- *Safety Assessment Process*

  Fault Hazard Analysis (FHA) all failures are classified according to severity, including justification for classification

  Preliminary System Safety Assessment (PSSA) failures identified in FHA are allocated to design components, defines maximum tolerable failure rate

  System Safety Assessment (SSA) actual design is evaluated wrt target goals of system analysis, failures found in FHA ⇒FTA, FMEA, FMES
  Common Cause Analysis is essential

- *Safety Assessment Analysis Method* techniques, that can be used to support safety assessment

DO-178B

- EURICAE WG 12 and RTCA Special Committee 167 (1992) ⇒common guidance in development of software systems that satisfy airworthiness
- different life cycles possible ⇒certification achieved by compliance with set of goals:
  - development activity type
  - software category
  - control category
- three kinds of activities characterize development:

  software planning process organize development and support activities

  development process build a software product

  integral process ensure correctness and quality of final system

- *software categorized* into five classes A to E according to effect of malfunction
- *control category* $CC1$ and $CC2$ defined through 13 characteristics, e.g. protection against unauthorized changes, $CC1$ is more stringent

Goals

- for each goal:

  description of goal in natural language

  applicability allocate achievement of objective to software categories

output artifact obtained by achieving the goal

control category $CC1$ or $CC2$

- lot of emphasis on testing

Verification activities

reviews on high-level artifacts (requirement/design) using common practices

analyses often algorithmic

testing requirements-based, number of goals dependent on software level

coverage different coverages for different software levels

- statement coverage: A, B, C
- decision coverage: A, B
- modified condition/decision coverage: A
- data and control coupling: A

Certification Goals

1. *certification basis* (agreement between certification authority and applicant, contains special conditions)
2. *assessment of the "Plan for Software Aspects of Certification"* how applicant achieves and demonstrates compliance with regulations
3. *compliance of software* analyze "Software Accomplishment Summary"
   - overview of system and software, certification considerations
   - software life cycle, data produced during project
   - software history (change history, unresolved issues)

Role of Tools

- any tool can be used, as long as outputs are verified through other means (e.g. automate a task, but not verification)
- under special conditions: tools can be used to demonstrate achievemnt of goal

  software development tools output is part of airborne software

  software verification tools cannot introduce errors, but might fail to detect them

Role of Formal Methods

- specific part in "alternative methods" part

Safety Case

- approach complementary to demonstrating compliances with norms ⇒shows that system can be safely operated
- demonstrate actual safety instead of compliance with regulations

  safety requirements and objectives define goals of analysis

  safety evidence define evidence on which analyses rely

  safety argument describes and argues, how safety evidence is sufficient to demonstrate achievements of safety objectives

- safety case evolves and is refined during development

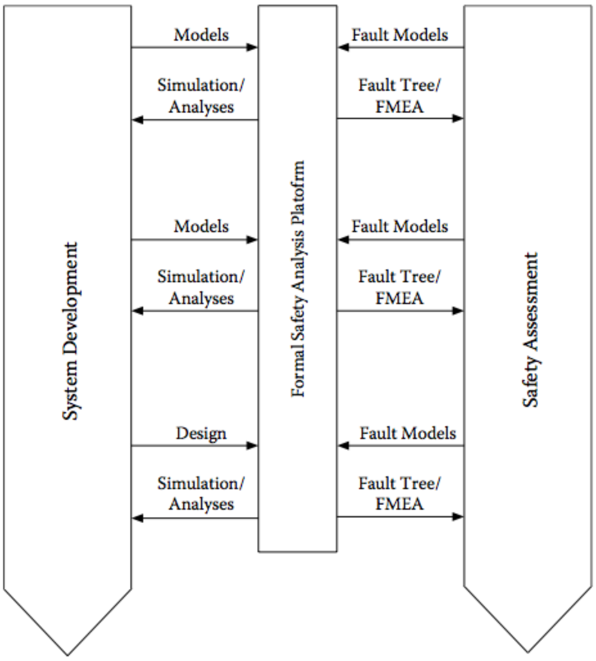
  preliminary safety case after system requirements have been defined ⇒justify the way in which Software Safety Plan delivers System Requirement Specifications that meet safety requirements

  interim safety case after specifications, demonstrate, that requirements meets safety specifications

  operational safety case includes set of evidence, that safety requirements have been met

Formal Methods and Certification

- training, tool robustness, tool qualification ⇒have to be addressed
- *after the fact* system is built, formal methods are used on final product
- *parallel* formal activities performed parallel to development of system
- *integrated* formal methods are used to drive system development
-



678-792