Dependability Engineering of Complex Computing Systems

M. Kaâniche
LAAS / LIS
kaaniche@laas.fr

J.-C. Laprie
LAAS / LIS
laprie@laas.fr

J.-P. Blanquart
Astrium / LIS
blanquar@laas.fr

September 11-14 2000, Tokyo, Japan
Dependability

Property of a system such that reliance can justifiably be placed on the service it delivers

(IFIP WG 10.4 - Dependable Computing and Fault Tolerance)
Motivation

- Developing dependable systems able to deliver critical services with a justified level of confidence is not easy
  - increasing complexity, fault diversity, conflicting objectives, ...

- Traditional development models do not explicitly incorporate all activities needed for the production of dependable systems
  - Hardware (BSI 5760 Standard)
    - incorporation of assessments
    - fault tolerance activities focussed on physical faults only
  - Software (Waterfall, V model, spiral, incremental, process oriented, ...)
    - structuring of activities
    - focus on verification
  - System engineering (EIA 632, IEEE 1220, ...)
    - generic pluridisciplinary framework integrating products, processes and people
    - dependability related issues are not detailed

- Need for a dependability-explicit development model
Basic activities

System Creation Process
- Requirements
- Design
- Realization
- Integration

Fault Prevention Process
- Formalisms & Languages
- Project organization
- Project planning & risk assessment

Fault Tolerance Process
- System behavior in presence of faults
- System partitioning
- Error & fault handling mechanisms

Fault Removal Process
- Verification
- Diagnosis
- Modification

Fault Forecasting Process
- Dependability objectives
- Allocation
- Evaluation
Interactions: examples

- Fault prevention process activities should be tightly coupled with system creation and dependability processes activities

- Fault tolerance and fault forecasting
  - Definition of dependability related requirements and functions
  - Allocation of dependability requirements
  - Assessment of the efficiency of fault tolerance mechanisms (coverage)

- Fault removal and fault tolerance
  - Verification of fault assumptions for traceability, consistency, completeness and verifiability
  - Verification of fault tolerance mechanisms by means of fault injection, formal verification or static analyzes

- Fault removal and fault forecasting
  - Validation of fault forecasting assumptions and results
  - Definition of test stopping criteria based on dependability level achieved
  - Evaluation of dependability based on test results
Fault Assumptions

- Fault assumptions should be defined at each system refinement step
  - Support for the definition of fault tolerance strategies and mechanisms
  - Check for traceability, consistency, completeness and verifiability
A meta-model not a life-cycle model

System requirements allocated to software

traditional Waterfall

reuse with adjustments

reuse without changes

Prototyping

Software development process

System development process

Rq Requirements  FP Fault Prevention
De Design        FT Fault Tolerance
Re Realization   FR Fault Removal
In Integration   FF Fault Forecasting

Software Product

A meta-model not a life-cycle model
Checklist

- System behavior / failures
  - dependability properties
  - criticality / mission phase
  - acceptable degraded modes
  - maximum tolerable duration of service interruption
  - number of simultaneous/ consecutive failures to be tolerated for each mode
  - fault tolerance means provided by the environment

- Project organization
  - life cycle model
  - resource management

- Project planning & risk assess.
  - risks identification & mitigation
  - dev. stages, transition criteria
  - planning of project reviews, certification, config. management

- Formalisms & languages
  - standards, rules, tools, formalisms

Requirements

- Dependability objectives
  - Failure modes analysis
    - classification by severity
  - FF assumptions
  - Function-by-function dependability allocation
    - classification of functions by criticality levels
  - Fault forecasting planning
  - Data collection and analysis

Fault Prevention

- Verification planning
  - static analyzes and testing strategies (criteria, input generation)
  - test-beds, environment simulators

Fault Tolerance

- Verification assumptions
  - classes of functions/ behavior
  - predicates

Fault Removal

- Requirements verification
  - traceability analysis
  - functional / behavioral analyses
  - reviews & inspections

Fault Forecasting

- Functional/ behavioral verification scenarios

- Environment description
  - boundaries and interactions

- Development and validation, constraints
  - foreseeable evolutions
  - interoperability, portability
  - reusability, testability, …
Checklist

- Formalisms & languages
- Project organization
- Project planning & risk assessment

- System behavior / faults
  - fault assumptions
- System partitioning
  - fault/error containment regions
  - FT application layers
- Fault tolerance strategies
  - redundancy, design diversity, exception handling
- Error & Fault handling mechanisms
  - error detection, diagnosis, recovery
  - fault diagnosis, passivation, reconfiguration
- Single points of failure?

Dashboard

Design

- Architecture
  - structure
  - behavior
  - data
- Low level requirements
- Reusable components?
- Operation and maintenance procedures definition
- System integration strategy

Fault Prevention

- Verification assumptions
- Design verification
  - behavioral analysis, reviews, inspections, prototyping
- Fault tolerance verification
  - (Formal) Verification
  - Simulation-based fault injection
- Unit / Integration testing planning
- Functional/structural verification scenarios
- Verification of FF results

Fault Tolerance

- FF assumptions
- Failure Mode Analysis
- Allocation / component
- Preliminary dependability assessment
- Data Collection & Analysis

Fault Removal

Fault Forecasting

Architecture

- structure
- behavior
- data

Low level requirements

Reusable components?

Operation and maintenance procedures definition

System integration strategy

Verification assumptions

Verification of FF results
Conclusion

- Structuring and controlling the development process is a prerequisite for the successful integration of fault tolerance and dependability-related mechanisms in complex systems.

- The proposed model provides a generic framework for structuring fault prevention, fault tolerance, fault removal, and fault forecasting activities:
  - iterative process
  - tradeoffs

- The guidelines aim to ensure that dependability-related issues are not overlooked but rather considered at each stage of the development.

- The proposed framework can be used to define and structure the evidence needed to support certification.