Software Analyses Training

In addition to fault tree, safety and failure modes analysis there are other analyses that can be employed to reduce defects in the software. Optionally, several of these analyses can be employed as a first step of the software FMEA to identify the weaknesses in the product and hence the most likely and applicable failure modes.

Analysis	Purpose	Associated failure modes
Safety requirements flow down analysis	Ensures safety requirements are flowed down into system, subsystem and software requirements	Faulty functionality - missing specifications, conflicting requirements, extraneous requirements
Requirements criticality analysis	Allows critical requirements to be tracked and helps find forgotten requirements	
Specification analysis	Examines specifications to uncover inconsistencies, conflicts, and ambiguous or missing requirements	
Traceability analysis	Allows a requirement to be followed throughout its life in both backwards and forwards directions	
Software process analysis	Examines whether software development requirements will be met *before* the software is implemented	
Testability analysis	Examines whether the SRS can be verified	
Functional flow block diagram analyses	Used to identify alternative or contingency operations, change to operational procedures, create the WBS	Faulty sequences, missing functionality
Control flow analysis	Identifies processes performed in series or parallel and prerequisite/dependent tasks	Faulty timing and sequences
Information flow analysis	Identifies incorrect, missing, inconsistent I/O specifications	Faulty I/O
Interface analysis	Examines functional and physical interfaces between software and hardware or software and software	Faulty data
State model analysis	Useful for uncovering missing states, dead states, unreachable states	Faulty state management
Timing analysis	Identifies deadlocks, livelocks, race conditions	Faulty timing
Schedulability analysis	Assesses whether a process is schedulable, identify priority inversions	Faulty timing
Throughput analysis	Assesses performance of software	Faulty timing
Sizing analysis	Assesses capacity requirements of software	Faulty processing

Analysis	Purpose	Associated failure modes
COTS/Reuse analysis	Determines which components should be purchased, subcontracted, reused, developed in house	Overall quantity of defects
Independence analysis	Verifies that safety critical software is separate/independent of non-critical software	Faulty maintenance
Logic analysis	Evaluates complex logic and control logic flow	Faulty sequences
Data analysis	Evaluates description and intended use of each data item in the software design; data usage is not in violation of safety requirements	Faulty data
Optimized code analysis	Identifies unused code, unreachable code, cloned code	Faulty sequences, state management
Interrupt analysis	Identifies priority inversions and infinite loops	Faulty timing
Complexity analysis	Assesses characteristics related to understandability, reliability, maintainability, testability	Faulty maintenance
Change impact analysis	Assesses how changes will impact the system	Faulty maintenance

The below analyses support the failure mode reduction by reducing the total quantity of defects.

Analyses	Purpose
Coverage analysis	Provides for a means to determine completeness, efficiency and effectiveness of unit testing and systems level testing
Cost analysis	Examines whether the staffing planned is sufficient
Test results analysis	Verifies that all identified hazards have been eliminated or controlled to an acceptable level of risk
Software element analysis	Verifies that each software element cannot cause or contribute to a hazard

Some analyses are employed throughout development while others are employed in only one phase such as the requirements phase. If the software development lifecycle is Agile or Incremental, the analyses are also conducted incrementally. The below table shows when the analyses are applicable.

Analysis	Reqs	Design	Code	Test
Safety requirements flow down analysis	X			
Requirements criticality analysis, specification analysis	X	X		
Traceability, software process, testability, cost analyses	X	X	X	X
Functional flow block diagram analysis	X	X		
Control flow, information flow, interface, state model, timing, schedulability, throughput, sizing analyses	X	X	X	
COTS/reuse analysis, independence analysis, software element analysis		X		
Logic analysis, data analysis		X	X	

Optimized code analysis, interrupt analysis, complexity analysis			X	
Test coverage analysis			X	X
Test results analysis, change impact analysis				X

Target Audience

Software engineers, software designers, software requirements engineers, software test engineers, software management, software safety engineering, software QA

Each course attendee is able to…

Analyze the software development artifacts so as to identify weaknesses that can be mitigated in phase. These weaknesses are often directly related to specific failure modes such as faulty state management, faulty timing, faulty sequencing, faulty data, faulty error handling, faulty processing, faulty memory management, etc.

Table of Contents

Topic	Expected time
Day 1	

Introduction to software analysis
   • Industry standards relevant to software analysis
   • Summary of each analysis and when used
   • Overview of each analysis
   • Tailoring guidelines
   • Artifacts to collect for each analysis
	1 hour

Software Analyses during requirements phase • Purpose of the analysis, Inputs and outputs of the analysis, Steps and techniques to perform the analysis, Documentation of results, Use the results to improve the software, Automated tools for the analysis, Examples	Remainder of day
Day 2	
Software Analyses during design phase • Purpose of the analysis, Inputs and outputs of the analysis, Steps and techniques to perform the analysis, Documentation of results, Use the results to improve the software, Automated tools for the analysis, Examples	All day
Software Analyses during coding • Purpose of the analysis, Inputs and outputs of the analysis, Steps and techniques to perform the analysis, Documentation of results, Use the results to improve the software, Automated tools for the analysis, Examples	Morning
Day 3	
Software Analyses during testing • Purpose of the analysis, Inputs and outputs of the analysis, Steps and techniques to perform the analysis, Documentation of results, Use the results to improve the software, Automated tools for the analysis, Examples	Afternoon

Conclusions	

Related Products and Services

Related products	Related services
Software FMEA toolkit	**Software FMEA services, software analyses services.**

Pricing and training options

This class is available as on site delivery. Call 321-514-4659 for a quotation.

sales@missionreadysoftware.com
https://missionreadysoftware.com
321-514-4659