GUIDE

IEEE 1633 Explained: What the Standard Requires, Task by Task

> IEEE 1633 is the IEEE Recommended Practice on Software Reliability. It defines a task-based process: plan the effort, predict reliability early, allocate requirements to software, analyze failure modes, model reliability growth during test, and feed field failures back through FRACAS.

What IEEE 1633 is for

IEEE 1633 exists because software failures are rarely random wear-out events; they are latent design and requirements defects that surface under specific conditions. The standard gives programs a repeatable way to plan for, predict, analyze and track those defects rather than treating software as an unmodeled block in the system reliability budget.

The core tasks

1. Software reliability program planning

Define objectives, the tasks to be performed, who performs them, the data to be collected and how software reliability is reported. This becomes the software reliability program plan, which is often a contractual deliverable.

2. Prediction

Estimate defect density, failure rate and availability early, using product and process characteristics rather than test data. Prediction is the only task available before code exists.

3. Allocation

Distribute the system-level reliability or availability requirement across software components, so that each component has a defensible target instead of an implicit assumption of perfection.

4. Software failure modes analysis

Identify how the software can fail — including software FMEA and software fault tree analysis — and drive mitigations into requirements, design and test. Section 7 of SAE J1025 and Annex of IEEE 1633 both address this analysis.

5. Reliability growth modeling

During test, fit observed failure data to growth models to estimate remaining defects and the time required to reach the reliability objective. Different models fit different defect discovery profiles, so model selection and goodness-of-fit matter.

6. FRACAS and field data

Software failures must enter the Failure Reporting, Analysis and Corrective Action System with enough detail — trigger conditions, state, inputs — to support root cause analysis and to calibrate the next prediction.

How the tasks fit the lifecycle

Phase	IEEE 1633 activity
Concept / proposal	Program planning, first prediction, allocation
Requirements	Prediction refresh, requirements-driven failure mode analysis
Design	Software FMEA, fault tree analysis, edge case identification
Test	Reliability growth modeling, defect discovery tracking
Fielded	FRACAS, root cause analysis, model calibration

Tailoring the standard

IEEE 1633 is a recommended practice, not a compliance checklist. Programs tailor the task set to criticality: a safety-critical avionics program will perform every task, while a commercial service may prioritize prediction, edge case analysis and availability modeling.

Frequently asked questions

▼ **What is the difference between IEEE 1633 and SAE J1025?**

IEEE 1633 covers the full software reliability program — planning, prediction, allocation, growth modeling and FRACAS. SAE J1025 addresses failure modes and effects analysis, with Section 7 specifically covering software FMEA. They are complementary, not competing.

▼ Is IEEE 1633 mandatory?

It is a recommended practice. It becomes mandatory when a contract, statement of work or higher-level standard invokes it.

▼ How do you include software in the system reliability model?

Convert the predicted software defect behavior into a failure rate or availability figure for each software item, then place those items in the system reliability block diagram alongside hardware.

Related pages

IEEE 1633 standard page SAE J1025 FMEA Section 7 Software in the system reliability model IEEE 1633 & SAE J1025 training