

GUIDE

How to Calculate Software Failure Rate and Availability

Software failure rate is derived from the number of defects expected to remain in the delivered code, the fraction of those defects that are actually triggered in operation, and the operating hours over which they can be triggered. Availability then follows from the failure rate and the time to restore service.

From defects to failure rate

Not every latent defect becomes a failure. A defect only causes a failure when its triggering condition occurs during operation. The chain is therefore:

1. Predict total defects for the release.
2. Subtract the defects expected to be found and fixed before delivery.
3. Estimate what fraction of the remaining defects are exercised in the operational profile.
4. Spread the resulting failures over the expected operating hours to obtain a failure rate.

MTBF is simply the reciprocal of that failure rate. The operational profile step is the one most often skipped, and it is the one that makes the number defensible — a defect in a rarely used maintenance mode is not the same risk as one in the main control loop.

From failure rate to availability

Availability = uptime / (uptime + downtime). With a failure rate and a mean time to restore service:

- **Inherent availability** uses only failure rate and repair/restart time.

- **Operational availability** also includes detection time, diagnosis, logistics delay and scheduled downtime.

For software, restore time is often dominated by detection and restart rather than by the fix itself, so automatic fault detection and fast recovery frequently improve availability more than reducing the defect count.

Placing software in the system reliability model

Situation	Modeling approach
Software item required for the function	Series element with its own failure rate
Identical software on redundant processors	Common-cause element — redundancy does not help for the same input
Dissimilar software implementations	Parallel element, with a correlation factor for shared requirements defects
Software that detects and safes the system	Model as detection coverage, not as a reliability improvement

Error budgets: the same math, different vocabulary

An error budget is an availability target expressed as the amount of unreliability permitted in a period. A 99.9% monthly target allows roughly 43 minutes of downtime. Predicted software failure rate and restore time tell you before the period starts whether that budget is realistic, and reliability allocation tells you which components must own which share of it.

Practical cautions

- State every assumption — defect removal efficiency, trigger fraction, operating hours — next to the result.
- Report a range. A single-point software MTBF invites false precision.
- Re-run the calculation after each test phase using actual defect discovery data.
- Never assume software contributes zero to the system failure rate; an unmodeled contributor is not a zero contributor.

Frequently asked questions

▼ How do you calculate software failure rate?

Estimate the defects remaining at delivery, apply the fraction expected to be triggered by the operational profile, and divide the resulting failures by the operating hours. The reciprocal is software MTBF.

▼ How do you allocate software reliability in a system reliability model?

Start from the system availability or reliability requirement, assign each software item a share based on criticality and usage, and verify each share against that item's predicted failure rate.

▼ Does redundant hardware improve software reliability?

Not for identical software. Two copies of the same code fail identically on the same input, so redundancy only helps against defects that are input-dependent across dissimilar implementations or against hardware faults.

Related pages

Predict failure rate & availability training Software in the system reliability model Reliable
software allocation System availability & error budgets