Top 100 overlooked software edge cases guide

How to spot the top 100 most common software defects and affect design and test cases using proven methodology

Contents

MISSION READY SOFTWARE

Defect Foresight: Spotting Edge Cases Early

Core Value Proposition

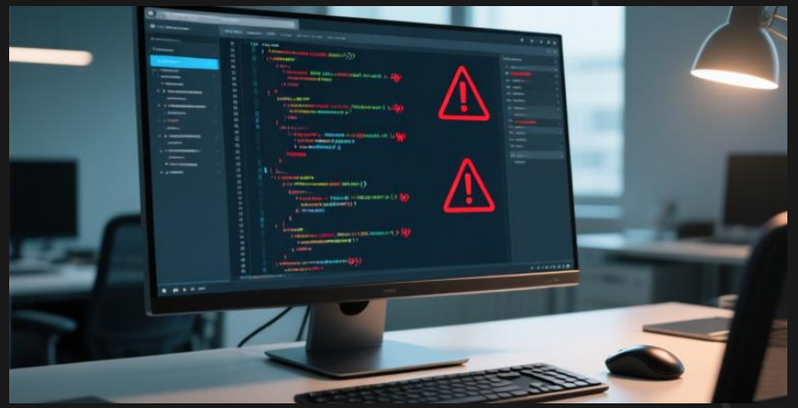
Learn how to spot edge cases hiding in plain sight before the code is even written. Built on 40 years of analyzing hundreds of thousands of software failures in mission and safety-critical systems.

Top 100 Software Defects

Learn to identify the most common software defects using proven Common Defect Enumeration methodology from decades of analysis.

Design & Test Impact

Once you know the edge cases, learn how to design systems and create test cases that prevent these defects.



Common Defect Enumeration (CDE) Methodology

Unique Research Foundation

Based on analysis of root causes across hundreds of thousands of failures in mission-critical systems.

System-Specific Filtering

Learn to filter the generic CDE list to identify which defects apply to your specific system architecture.

Design Review Insights

Review system models and focus on what's missing from specifications to spot hidden defects.

Proactive Mitigation

Implement controls and safeguards before coding begins, preventing defects rather than fixing them.

MISSION READY SOFTWARE

Common Defect Enumerations: The Edge Cases

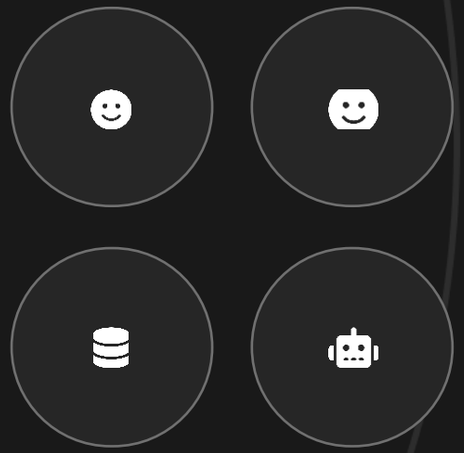
Comprehensive analysis of defect categories based on real-world failure data from mission-critical systems, organized by root cause patterns.

Failure Mode Category	Description	Root Causes
State Management	Inadvertent state changes, stuck states	18
Error Handling	Not detecting hardware, power, or computational faults	41
Functionality	Missing functions, works perfectly to wrong spec	18
Processing	Inability to execute under load or extended time	9
Timing	Race conditions, events too late/early, timing budgets	25
Usability	Not advising users of irreversible events	11
Data Definition	Incompatible units, data types, resolution	21
Sequencing	Events happen in wrong order	11
Algorithms	Algorithm overflows and computational errors	11
Machine Learning	Faulty population, modeling, training, validation	18

Critical Defect Categories Analysis

Highest Impact Categories

Error handling leads with 41 root causes, followed by timing issues with 25 causes - critical system failure points.



State & Function Issues

State management and functionality defects each account for 18 root causes - core system behavior problems.

Data & Interface Problems

Data definition issues cause 21 root causes - critical for system integration and data exchange accuracy.

Emerging AI Challenges

Machine learning defects represent 18 root causes - a growing concern in modern intelligent systems.

Proactive Mitigation and Testing Strategy

Cost effective
Requirements Enhancement

Better
Fault Injection Testing

MBSE
System Integration

Feeds
Automated Testing

Requirements Enhancement

> Improve software specifications and design by adding controls and mitigations for common defects.

- Add defect-specific controls to requirements specifications
- Enhance design patterns to prevent common failures
- Integrate CDE insights into development workflows

Test Strategy

> Gain actionable edge case testing recommendations using targeted fault injection techniques.

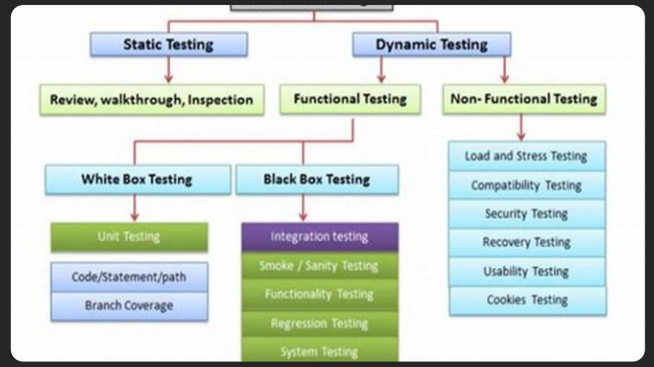
- Test for specific defects using fault injection
- Create edge case scenarios from CDE data
- Validate system behavior under adverse conditions

System Integration

> Feed identified edge cases into MBSE and automated testing tools for comprehensive coverage.

- Integrate CDE into MBSE design flows
- Automate edge case testing workflows
- Connect defect patterns to system models

Who Should Take This Training

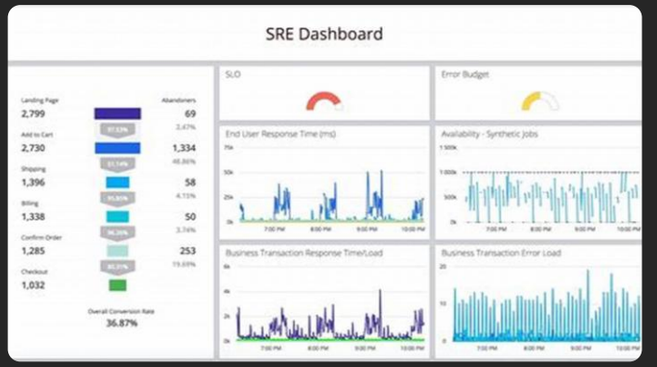


Software Test Engineers

- Learn to see defects before software is coded
- Develop effective testing strategies for edge cases
- Implement fault injection testing methodologies

Software Safety Engineers

- Spot underlying root causes for safety hazards
- Enhance safety analysis with defect patterns
- Improve hazard identification and mitigation



RAM & Site Reliability Engineers

- Identify defects that kill availability and uptime
- Improve system reliability and maintainability
- Enhance operational resilience planning

"

Learn to spot overlooked defects and edge cases hiding in your system architecture and MBSE design flows.

— For Software Engineers & Systems Engineers

This methodology empowers development teams to proactively identify and prevent common software defects by leveraging decades of failure analysis from mission-critical systems, transforming reactive debugging into predictive engineering excellence.

Take Action: Transform Your Testing Approach

Ready to Get Started?

Join thousands of engineers who have transformed their approach to software quality using the Common Defect Enumeration methodology. Available in flexible delivery formats to meet your learning needs.

Virtual Self-Guided

Learn at your own pace with comprehensive materials, interactive exercises, and real-world case studies available anytime.

Virtual Instructor-Guided

Join live sessions with expert instructors for personalized guidance, Q&A sessions, and collaborative learning experiences.

Schedule Discussion Today

Contact our team to discuss your specific needs and learn how this methodology can benefit your organization.

Ready to Transform Your Development Process?

Contact sales@missionreadysoftware.com today to schedule your consultation and start optimizing your software development trade-offs.