

Critical Software X-Ray Checklist

A founder and CTO checklist for inherited software, modernization, rescue work, and controlled AI decisions. Use it before funding a rewrite, adding AI, or changing a revenue-critical workflow.

When to use this

Use this before	A rewrite, AI feature, large refactor, payment/access change, migration, or rescue sprint.
The goal	Expose hidden business rules, silent failure paths, rollback gaps, and ownership risk before major spend.
Best next step	If 2 or more risk signals are true, run a focused X-Ray before committing to a larger change.

Six risk signals to score

Revenue or access can fail silently	Can payment, subscription, entitlement, onboarding, renewal, or data handoff fail without leadership seeing it quickly?	Stabilize the critical path before adding scope.
Business rules live outside the code	Do support, sales, operations, or one senior developer know exceptions that are not mapped or tested?	Build a Business Rule Atlas before changing behavior.
Releases feel risky	Do small changes require disproportionate regression effort because dependencies are uncertain?	Trace the journey and add verification around it.
AI is discussed before controls	Are permissions, audit, abstention, fallback, and human ownership unclear?	Define controlled AI boundaries before implementation.
Critical workflows are hard to explain	Can leaders explain user action, system state, data change, and business outcome in one path?	Create a shared map before funding delivery.
Modernization scope keeps expanding	Does improvement keep turning into rewrite because the current system is not understood?	Diagnose first, then sequence the safest moves.

What to collect before an X-Ray

Journey boundary	One business journey: checkout, onboarding, renewal, reporting, approval, data exchange, admin workflow, or another critical path.
Known failures	Recent incidents, recurring bugs, slow releases, customer complaints, manual workarounds, or risks leadership worries about.
System map	Applications, databases, queues, APIs, third-party tools, hosting, cron jobs, admin panels, and ownership points.
Evidence samples	Logs, screenshots, tickets, diagrams, test notes, audit trails, support examples, or anonymized walkthroughs.
Decision constraint	What cannot break, what must improve first, timing pressure, budget range, and who owns the decision.

Score interpretation

0-1	Improve deliberately. You may be ready for a focused implementation or light modernization plan.
2-3	Run an X-Ray before larger change. The system likely has hidden dependencies or leadership uncertainty.
4-6	Stabilize the critical path first. Avoid rewrite or AI expansion until the failure path and rollback plan are visible.

Recommended next action

Request a 20-minute Risk Triage. Tech Mitrise will review the business consequence, timing, and fit, then suggest a practical first step: direct guidance, paid Software X-Ray, Critical Path Rescue, modernization sprint, or another specialist if that is better for the buyer.

Website: techmitrise.com/risk-triage/ | Email: biz@techmitrise.com