

Enterprise Risk Map

**A self-assessment framework for
enterprise CMS migration decisions**



Executive summary

This is a self-assessment framework to help you determine when to migrate your content management system (CMS). It covers five risk domains that get worse when legacy systems stick around, with diagnostic scales so you can see where you actually are right now.



WHAT WE MEAN BY “LEGACY SYSTEMS”:

Software that’s still running critical business functions but built on outdated architecture, meaning integration models that are custom-first rather than API-first, updates that require extensive testing and manual coordination instead of automated deployment, and scaling that’s nonlinear rather than elastic. It might still get vendor patches and technically connect to modern tools, but every integration requires custom work, changes carry more risk than they should, and the system is expensive to maintain and hard to evolve.

Here’s the thing: most organizations aren’t evenly at risk. You might be Stage 1 in Security Risk but already at Stage 3 in Workforce Risk. This assessment helps leadership align on which domains actually matter to your business and on the timeline that makes sense.

THIS ISN’T PASS/FAIL

Stage 1-2 across most domains? You’re in strategic planning mode — no rush. Stage 3 in multiple domains? Migration’s becoming a real priority. Stage 4-5 anywhere? Time to build that business case. Use your scores to figure out timing, not to freak out.

Table of Contents



4

HOW TO USE THIS ASSESSMENT:

Step 1: Plot your current stage

Step 2: Identify your highest-risk domains

Step 3: Map to timeline

Step 4: Use role-based questions to validate

6

DETAILED RISK ANALYSIS

Workforce risk

Security risk

Integration entropy

Vendor lock-in risk

Cost volatility

14

ROLE-BASED CONVERSATION STARTERS

For the CTO

For the CISO

For the VP of engineering

For the enterprise architect

For procurement

For the head of digital

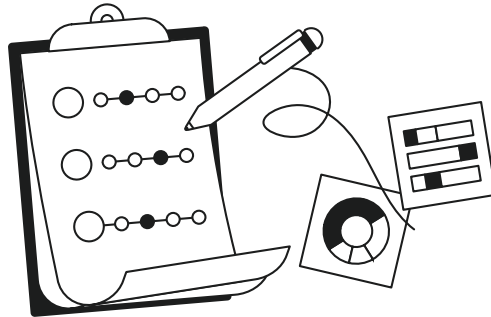
16

BUILDING YOUR MIGRATION BUSINESS CASE

Quantify the risk exposure

Frame the migration as risk mitigation

How to use this assessment



This framework is designed for cross-functional leadership teams who need to align on migration timing. Here's how to work through it:

STEP 1

Plot your current stage

Read through each of the five risk domain scales in the next section. For each domain, identify which stage best describes your organization today. Don't overthink it if you're between stages, just pick the one that feels closest.

Most organizations are Stage 1-2 in some areas, Stage 3-4 in others.* That's normal. The task here is to look for patterns, not perfection.

**See the stages in the next section.*

STEP 2

Identify your highest-risk domains

Which domains are at Stage 3 or higher? These are the areas where inaction carries measurable business impact. If you're Stage 4-5 in any single domain, that alone may justify migration urgency regardless of your scores elsewhere.

STEP 3

Map to timeline

Use this rough guideline to translate stages into migration timing:

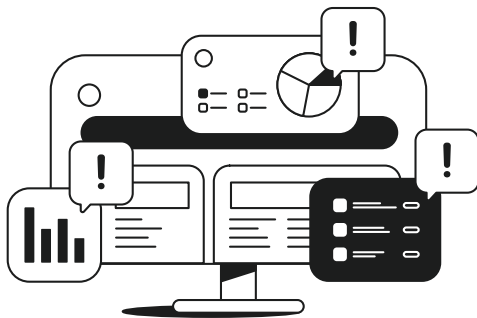
YOUR RISK PROFILE	WHAT IT MEANS FOR TIMELINE
Stage 1-2 across most domains	Migration is part of strategic planning. 12-24 month horizon. You have time to evaluate options and build consensus.
Stage 3 in 2-3 domains	Migration becomes a priority. 6-12 month planning window. Start building the business case and identifying internal champions.
Stage 3 in 4+ domains OR Stage 4 in any domain	Migration is urgent. Immediate business case needed. Risks are compounding and affecting operations or competitive position.
Stage 5 in any domain	Migration is overdue. You're managing around system limitations. Every quarter of delay increases switching costs and organizational exposure.

STEP 4

Use role-based questions to validate

After scoring each domain, review the role-based conversation starters at the end of this document. These are concrete questions leadership can ask their teams to verify whether your self-assessment matches ground truth.

Detailed risk analysis



Each risk domain includes a diagnostic scale to help you assess where your organization stands today and what to expect if migration is delayed.

1

WORKFORCE RISK

Definition: The growing difficulty of attracting and retaining engineering talent when you’re running legacy tech stacks.

Data context: [ISACA’s 2025 survey](#) of 7,700+ tech professionals found that one in three switched jobs in the past two years, and 74% of organizations are worried about retention. The top stressors? Heavy workloads and long hours are both worse when teams are maintaining aging systems while trying to ship new features.

A [2025 Saritasa survey](#) of 500+ U.S. IT professionals found 62% of organizations still run legacy software. These systems often lack documentation particularly around custom integrations and modifications built over time because the people who built them leave. When institutional knowledge walks out the door, it becomes even harder to find qualified maintainers who can understand how the system actually works.

The impact shows up in day-to-day work. [Storyblok’s 2026 Speed-to-Market Benchmark](#) found that 38% of marketing and digital teams require developer support for most or every campaign, with over a third of developers spending 25–50% of their time on GTM campaign support. And 42% say tech platform limitations make that support harder than it should be. When developers are stuck maintaining legacy systems instead of building new capabilities, retention problems get worse.

STAGE	SYMPTOMS
Stage 1: Early Warning	• Job postings take longer to fill than they used to • Candidates ask about your tech stack in interviews • Junior devs want exposure to “modern” tools

STAGE	SYMPTOMS
Stage 2: Pressure Points	• People don't stay as long as they used to • Difficulty finding candidates willing to work with the current stack • Team complains about lack of community support/documentation
Stage 3: Retention Crisis	• Employees demand higher salaries, citing niche/legacy skillset • You're losing people to competitors with modern stacks • Backfill candidates need 6+ months of onboarding due to system complexity
Stage 4: Knowledge Risk	• There are 1-2 people you're afraid to lose, as the entire system depends on them • Institutional knowledge walkout: each departure erases critical context • New hires refuse offers after learning about the tech stack
Stage 5: Critical	• Hiring freezes because you can't find qualified maintainers • Paying premium salaries (20-40% above market) to retain legacy expertise • System stability depends on contractors who know the codebase

Primary impact: CTO (talent strategy), VP Engineering (team velocity and morale)

WHITEPAPER

Learn more about talent economics in legacy vs modern architecture.

DOWNLOAD THE WHITEPAPER [↗](#)

2

SECURITY RISK

Definition: Increased vulnerability exposure from patch lag, end-of-life software, and compliance drift in legacy systems.

Data context: Legacy systems are high-value targets. [As of September 2025](#), 44.6% of vulnerabilities in CISA's Known Exploited Vulnerabilities catalog had been known for at least two years before active exploitation, meaning old vulnerabilities stay dangerous. About 67.5% of the most-exploited vulnerabilities over the past year were legacy flaws.

According to [Storyblok's 2026 State of Security report](#), 39% of security teams experienced a security issue impacting their content strategy in the last year, and when broken down by system type, the majority were using legacy systems. Meanwhile, [nearly 29% of exploited vulnerabilities in 2025](#) were weaponized on or before their CVE(common vulnerabilities and exposures) publication date, giving organizations running unpatched legacy systems essentially no grace period.

STAGE	SYMPTOMS
Stage 1: Early Warning	- Patches are applied within vendor-recommended windows, but it takes coordination - Some third-party dependencies haven't been updated in 12+ months - Audit logs show the system is working, but you're not monitoring them actively
Stage 2: Patch Lag	- CVEs are identified, but patching takes weeks instead of days - Some integrations/plugins/extensions are past vendor support windows - Security team manually tracks which systems are vulnerable
Stage 3: Exposure	- Known vulnerabilities remain unpatched for 30+ days - Pen tests reveal issues that can't be fixed without system upgrades - Compliance audits flag "acceptable risk" items that are growing in number
Stage 4: Compliance Drift	- Running software versions no longer covered by vendor support - Audit findings require compensating controls you can't fully implement - CISO briefs board on "legacy system risk" as a standing agenda item

STAGE	SYMPTOMS
Stage 5: Critical	- Active exploits exist for vulnerabilities you can't patch - Insurance premiums increase, or coverage is denied due to legacy infrastructure - Breach incident response plans specifically account for legacy system limitations

Primary impact: CISO (breach accountability, compliance posture)

WHITEPAPER

Learn where the security team's attention and budget really belong this year.

DOWNLOAD THE WHITEPAPER 



3

INTEGRATION ENTROPY

Definition: The gradual loss of clarity, control, and reliability across system integrations as your tech stack grows.

Why it escalates: Legacy CMSs often lack modern API architecture, so teams build custom middleware or workarounds. Each new integration adds complexity that doesn't usually get documented properly. "Temporary" solutions become permanent, and nobody's sure which system is actually authoritative anymore.

STAGE	SYMPTOMS
Stage 1: Early Warning	- Each new marketing tool requires IT involvement to connect - Integration documentation exists but is scattered - Occasional sync failures that resolve with restarts
Stage 2: Middleware Sprawl	- Custom connectors built for each new tool (no reusable API layer) "Temporary" workarounds from 2+ years ago still in production

STAGE	SYMPTOMS
Stage 2: Middleware Sprawl	Only 1-2 people understand how certain integrations work
Stage 3: System-of-Record Ambiguity	- Teams argue over which system is authoritative for key data - Data syncs create conflicts, and different tools show different values - New integrations require auditing whether they'll break existing ones
Stage 4: Cognitive Overload	- Architecture diagrams are out of date within weeks of being created - IT says "no" to new tool requests because integration risk is too high - Changes to one system cause unpredictable failures in others
Stage 5: Critical	- Major incidents traced to integration failures (cascading outages) - Business capabilities are blocked waiting for integration work "Integration freeze" policies prevent marketing from launching campaigns

Primary impact: Enterprise Architect (system coherence),
VP Engineering (maintenance burden), Head of Digital (speed to market)

4

VENDOR LOCK-IN RISK

Definition: Proprietary tooling dependencies, migration exit costs, and reduced negotiating leverage that pile up over time.

Why it escalates: The longer you run a legacy system, the deeper the dependencies get. Custom modules, proprietary workflows, and vendor-specific data formats make switching costs grow exponentially — and vendors know it. Renewal negotiations stop being about value and start being about how hard it would be for you to leave.

STAGE	SYMPTOMS
Stage 1: Early Warning	- Some customizations exist but could be replaced with standard features in other platforms - Your team has explored what migration would look like - Contract renewals are routine, but prices creep upward annually

STAGE	SYMPTOMS
Stage 2: Proprietary Workflows	• Business processes are built around vendor-specific features that don't have equivalents elsewhere • Custom code is tightly coupled to the vendor's architecture, making it costly to decouple • Procurement estimates 6-12 months for migration if needed
Stage 3: Deep Integration	• Vendor's tooling is embedded across multiple business functions • Data formats are vendor-specific with no clean export path • License negotiations feel one-sided as the vendor knows you can't leave easily
Stage 4: Strategic Constraint	• New business capabilities are blocked by vendor roadmap delays • Competitive features unavailable because your vendor doesn't support them • Contract renewal prices increase 15-25% with take-it-or-leave-it terms
Stage 5: Critical	• Migration cost estimated at 18-36 months and 7-figure budget • Vendor acquisition or policy change forces unplanned architecture decisions • Board identifies vendor dependency as enterprise risk

Primary impact: Procurement (contract leverage), CTO (strategic flexibility)

⑤

COST VOLATILITY

Definition: Hidden maintenance overhead, unpredictable scaling fees, and budget creep from legacy infrastructure.

Why it escalates: Legacy systems don't have stable costs. Maintenance hours creep up as people who know the system leave. Licensing models change when vendors get acquired or consolidate. Professional services costs spike during upgrades you can't avoid. This way, the real costs are hidden, and you can't actually measure how much engineering time goes to keeping the lights on versus building anything new.

STAGE	SYMPTOMS
Stage 1: Early Warning	- Annual maintenance costs are stable but trending upward - Occasional unplanned consulting fees for upgrades - Engineering time split is roughly 60% new features and 40% maintenance
Stage 2: Maintenance Creep	- Engineering time shifts to 50/50 or worse (features vs. maintenance) - Vendor raises license fees citing “market adjustments” - Unplanned costs appear each quarter (patches, support extensions)
Stage 3: Budget Unpredictability	- Can’t forecast annual IT costs accurately as variances exceed 15% - Professional services required for routine updates - Scaling costs are nonlinear (2x traffic ≠ 2x cost, it’s 3-4x)
Stage 4: Hidden Overhead	70%+ of engineering capacity goes to maintenance, not new features - Finance flags CMS infrastructure as budget risk - Delayed launches due to “infrastructure not ready” (capacity, not features)
Stage 5: Critical	- Total cost of ownership exceeds modern alternatives by 40-60% - Emergency vendor support contracts signed at premium rates - CFO mandates migration business case due to unsustainable cost trajectory

Primary impact: Procurement (budget predictability), CFO (total cost of ownership)

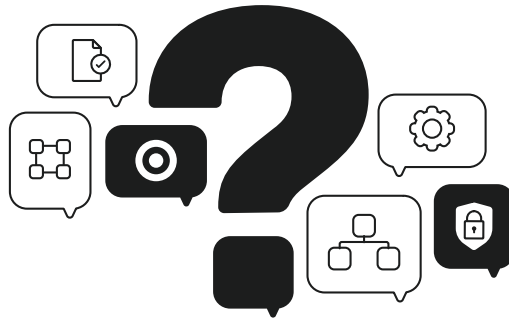
WHITEPAPER

Explore CMS spending for implementation, annual maintenance, and yearly training & support.

DOWNLOAD THE WHITEPAPER [↗](#)



Role-based conversation starters



Use these questions to validate your self-assessment scores with the teams who feel the ground truth every day. If the answers surprise you, revisit your stage ratings.

FOR THE CTO

- When candidates ask about our tech stack, what specifically are they concerned about? Are they worried about legacy system maintenance, or asking standard technical questions? Do we lose candidates because of the stack itself?
- Of the senior engineers who left in the past 18 months, how many cited tech stack or maintenance burden as a factor (even partially)?
- If we lost our top 2-3 engineers tomorrow, which systems would be at risk?
- What's our vendor renewal negotiation leverage compared to three years ago?
- How much has our architecture decision-making been constrained by vendor roadmap delays?

FOR THE CISO

- How many known vulnerabilities are currently past our target remediation window?
- What percentage of our infrastructure components are past vendor support windows?
- How many 'acceptable risk' items have we carried forward in the past two audits?
- If we had a breach tomorrow, which legacy systems would our IR team flag as unmonitorable?
- Has cyber insurance pricing or coverage changed based on our infrastructure profile?

FOR THE VP OF ENGINEERING

- What's our actual feature development vs. maintenance split over the past quarter?
- How many engineers can explain how our core integrations work?
- How often we said 'no' to a new tool request because integration risk was too high?
- How many workarounds that were supposed to be temporary are still in production?
- What's our average onboarding time for new engineers, and how much of that is legacy system training?

FOR THE ENTERPRISE ARCHITECT

- How current are our architecture diagrams, and how much manual work goes into keeping them updated?
- Which systems do teams argue over when determining system-of-record for key data?
- How many integration failures in the past six months required cross-team coordination to diagnose?
- What percentage of new integrations require custom middleware instead of using standard APIs?
- How many 'temporary' point-to-point integrations are still running after 2+ years?

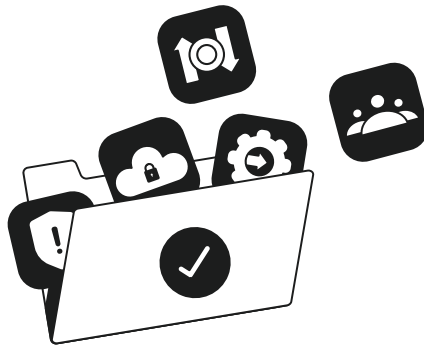
FOR PROCUREMENT

- How have our CMS vendor's renewal prices changed over the past three contract cycles?
- What's our estimated migration cost if we decided to switch vendors today?
- How many unplanned professional services engagements have we needed in the past year?
- What's our variance between budgeted and actual IT infrastructure costs over the past two years?
- How much negotiating leverage do we have in our next renewal, realistically?

FOR THE HEAD OF DIGITAL

- How many marketing initiatives were delayed because IT couldn't integrate a new tool in time?
- What's the average time from 'we want this tool' to 'it's connected and working'?
- How many campaigns have we postponed or scaled back due to technical constraints?
- What competitive capabilities are we missing because our CMS doesn't support them?
- How much time does your team spend on workarounds instead of strategy?

Building your migration business case



Your stage scores across the five domains give you the raw material for a migration business case. Here's how to translate self-assessment into internal justification.

QUANTIFY THE RISK EXPOSURE

For each domain where you're Stage 3 or higher, attach a business impact metric:

DOMAIN AT STAGE 3+	SAMPLE BUSINESS IMPACT METRIC
Workforce Risk	• Average time-to-fill for engineering roles: X months • Annual churn rate: Y% • Premium paid for legacy skillsets: \$Z above market
Security Risk	• Known unpatched vulnerabilities: X • Components past vendor support: Y% • Cyber insurance premium increase: Z%
Integration Entropy	• Average time to integrate new tool: X weeks • Integration failures per quarter: Y • Engineering hours on integration maintenance: Z/month
Vendor Lock-In	• Contract renewal price increase: X% • Estimated migration cost today: \$Y • Features blocked by vendor roadmap: Z initiatives
Cost Volatility	• Variance between budgeted and actual IT costs: X% • Engineering capacity on maintenance: Y% • TCO vs. modern alternative: Z% higher

FRAME THE MIGRATION AS RISK MITIGATION

Modern headless CMS architecture addresses four of these five risk domains by design:

- **Workforce risk** → Modern stacks attract talent. API-first architecture, cloud-native infrastructure, and active developer communities reduce onboarding time and increase retention.
- **Security risk** → SaaS headless platforms maintain vendor-managed security posture, automated patching, and compliance certifications. You're no longer responsible for maintaining infrastructure security.
- **Integration entropy** → GraphQL and REST APIs provide clean integration patterns. Composable architecture replaces brittle point-to-point connections with a structured content layer.
- **Vendor lock-in** → While it doesn't disappear with migration, it resets. The difference is that modern headless platforms use open standards and provide clean export paths, reducing switching costs if you need to move again. And because everything connects through APIs, you're not stuck in the vendor's proprietary ecosystem. You can swap out your frontend framework, add any third-party tools, or change your hosting setup without touching the CMS itself.
- **Cost volatility** → Transparent SaaS pricing models replace unpredictable infrastructure costs. Predictable scaling, minimal maintenance overhead, and reduced professional services dependencies stabilize the total cost of ownership.

CONCLUSION

Migration timing is a risk calculation

Migration isn't the risk. Staying put is.

Every organization has a different risk tolerance. Some can absorb higher workforce churn. Some can't afford any security exposure. Some have more vendor leverage than others. There's no universal "right time" to migrate.

But there is a framework for making the decision explicit.

This assessment maps the five risk domains that escalate when legacy systems stay in production. It gives cross-functional leadership a common language to align on what you're actually risking and which risks matter most to your business.

Use these to pressure-test your current migration timeline. If you're planning to migrate in 18 months, map that against the risk escalation curve. Which domains hit Stage 4 or 5 before you've moved? That's where the conversation starts.

Migration, made easy.

Replatform without regret.

A structured, technical, and security-first path to moving from your current CMS to Storyblok's infrastructure that holds up for the next decade and beyond.

[READ MORE](#) 



Get Joyful. 

**See how Storyblok can revolutionize
your content at storyblok.com.**

TRY FOR FREE 

CONTACT US 