ACCELION INTELLIGENCE PLATFORM — INSIGHTS REPORT

NovaCrest Platform

Engineering Intelligence Assessment

CLASSIFICATION	Executive — Strategic Planning
REPORT DATE	February 2026
ANALYSIS PERIOD	65 days · 6,500+ commits · 1.1M lines of code
PLATFORM	.NET / ASP.NET · SQL Server · Cloud Services
PREPARED BY	Founders Led Studio — Accelion Intelligence

> **SAMPLE REPORT** — This document demonstrates the Accelion Insights reporting capability using an anonymized real-world engagement. Company name, developer identities, and identifying details have been changed. All findings, metrics, and analysis methodology are authentic.

Contents

SECTION 01

Executive Risk Summary

Three independent analyses of the NovaCrest Platform have been completed — covering dead code, security vulnerabilities, and operational dependencies. This is what we found.

COMPOSITE RISK SCORE
7.5 / 10
HIGH — Immediate action required

GHOST CODE RATIO
37%
22,850 of 61,757 components are dead

SECURITY VULNERABILITIES
995
366 critical · CVSS 9.0+ findings

BUS FACTOR
1
82% of commits by a single developer

MONTHLY TECH DEBT COST
$27K
Escalating to $69K within 12 months

COMPLIANCE SCORE
25%
GLBA: 3 of 12 controls passing

Bottom Line: The cost of comprehensive remediation ($1.0M–$1.2M one-time) is **less than one year** of the estimated risk exposure from doing nothing ($830K–$2.6M annually). Every month of delay increases the probability of a security incident, regulatory finding, or compliance failure.

Key Numbers at a Glance

METRIC	VALUE
Total Codebase	1,148,497 LOC across 49 C# projects
Zombie Code	153,000 LOC potentially removable (22,850 components, 37%)
Security Vulnerabilities	5 CRITICAL, 8 HIGH, 5 MEDIUM (consolidated unique)
CVSS 9.0+ Findings	2 (hardcoded credentials, SQL injection)
EOL/Deprecated Packages	12+ packages, 33 version conflicts
Total Remediation (all reports)	$1.0M – $1.2M over 12 months
Annual Risk Exposure (do nothing)	$830K – $2.6M/year

SECTION 02

Codebase Archaeology

Accelion's static and dynamic analysis engine classified every component in the NovaCrest codebase. The result: **37% of the codebase is ghost code** — dead components that consume build time, inflate the attack surface, and obscure the business logic that actually matters.

TOTAL COMPONENTS
61,757
Analyzed across all modules

LIVE COMPONENTS
38,907
63% of codebase is actively used

ZOMBIE COMPONENTS
22,850
37% confirmed removable

Where the Dead Code Lives

Approximately **153,000 lines of code** are removable today across four zones, with zero functional impact on live features:

CATEGORY	RISK
Orphaned projects (3 complete modules with no entry points)	HIGH
ORM Framework (auto-generated entity metadata, unused by live features)	CRITICAL
Dead configuration frameworks (built but never called)	MEDIUM
Abandoned admin tools (zero registered routes)	MEDIUM

Live vs. Dead Feature Architecture

A striking pattern emerged: the *active* business features use hardcoded validation and embedded rules, while sophisticated configuration frameworks built to manage those same rules sit completely unused.

Live Features (Hardcoded)

FEATURE	IMPLEMENTATION
Premium Feature Access	Hardcoded validation
Compliance Rule Checking	Embedded compliance rules
Reapplication Logic	Hardcoded timing thresholds

Dead Configuration Systems

SYSTEM	STATUS
Entity Metadata Framework	Built, never called
Static Relations Engine	Built, never called
Permission Configuration	Built, never called

> **Business Impact:** Approximately **153,000 lines of code** can be safely removed *immediately* with zero business impact, including 3 orphaned projects with no entry points or deployments. Cutting the dead weight reduces build times and shrinks the active attack surface.

SECTION 03

Knowledge Concentration

The NovaCrest Platform has a **bus factor of 1**. A single contributor owns 82% of all commits across the entire repository, and 12 critical business modules have 80–100% single-person ownership. If one person is unavailable, development stops.

REPOSITORY BUS FACTOR

1

Complete halt if primary dev unavailable

PRIMARY DEVELOPER CONCENTRATION

82%

Single developer owns 82% of all commits

TOTAL CONTRIBUTORS

10

Severely limited backup resources

Critical Single-Owner Modules

MODULE / COMPONENT	OWNER	OWNERSHIP	CHANGES	RISK LEVEL
Loan Calculator	Dev Delta	100%	14	EXTREME
Core Application Templates	Dev Alpha	98%	568	EXTREME
Cloud Identity (Auth) System	Dev Beta	94%	17	CRITICAL
Development Scripts	Dev Alpha	92%	712	CRITICAL
Credit Bureau Integration	Dev Mu	88%	348	HIGH
Data Access Layer	Dev Delta	81%	16	HIGH

What This Means: The loan calculation engine — the core revenue-generating function — has **100% ownership by a single developer** with zero cross-training. The authentication system is 94% owned by one person. These aren't just technical risks; they are *business continuity risks* that should be escalated to executive leadership.

SECTION 04

Security Posture

Comprehensive security assessment across 6,500+ commits revealed **995 security vulnerabilities**, with 366 requiring immediate attention. The system demonstrates a pattern of security debt accumulation with hardcoded credentials as the dominant vulnerability class.

CRITICAL VULNERABILITIES
366
Immediate action required

HIGH SEVERITY
202
Action within 1–2 weeks

TOP CVSS SCORE
9.5
Hardcoded cloud credentials

Vulnerability Distribution

CATEGORY	CRITICAL	HIGH	MEDIUM	LOW	CWE
Credentials & Secrets	**366**	45	12	3	CWE-798
Cryptographic Issues	**314**	89	156	8	CWE-327
SQL Injection	**89**	34	67	5	CWE-89
Path Traversal	**51**	23	89	7	CWE-22
Information Disclosure	**49**	11	73	7	CWE-200

Attack Chain Analysis

The compound effect of these vulnerabilities creates a viable attack chain:

1. Extract hardcoded cloud credentials from source code (366 instances)
2. Leverage cryptographic weaknesses to compromise data integrity
3. Execute SQL injection against financial databases
4. Perform path traversal to access borrower documents
5. Exploit information disclosure for further reconnaissance

Compliance Impact

FRAMEWORK	STATUS	GAP
GLBA (Financial Data Protection)	25% COMPLIANT	Weak encryption, exposed credentials, no incident response plan
SOC 2 Type II	NON-COMPLIANT	EOL software, no SAST/DAST, weak session management
PCI DSS	NON-COMPLIANT	jQuery CVEs fail quarterly scans, hardcoded secrets, debug mode
NIST CSF	MULTIPLE GAPS	No formal risk assessment, limited detection/response

Financial Risk of Inaction

RISK SCENARIO	PROBABILITY	IMPACT
Data breach (customer PII)	High	$2.4M – $4.8M
Regulatory penalties	Very High	$500K – $2M
Business disruption	Medium	$1M – $3M
Expected Loss		**$3.9M – $9.8M**

SECTION 05

Cost of Waiting

Technical debt is not static. The NovaCrest Platform carries **$27,000 in monthly technical debt cost** that our model projects will escalate to **$69,000 monthly** without intervention — a 2.56x increase driven by stalled migrations, legacy workarounds, and compounding security exposure.

CURRENT MONTHLY COST
$27K
Active technical debt burden

PROJECTED 12-MONTH COST
$69K
2.56x escalation without action

MIGRATION PROGRESS
15%
Average across all initiatives

Technical Debt Cost Breakdown

COMPONENT	MONTHLY HOURS	MONTHLY COST	12-MO ESCALATION
Legacy Framework Workarounds	40	$9,000	2.5x
Manual Report Generation	60	$13,500	3x
Migration Complexity Overhead	80	$18,000	4x
Total	**180**	**$40,500**	**2.56x average**

Migration Initiative Status

INITIATIVE	PROGRESS	DURATION	SUB-TASKS	STATUS
PartnerPortal Migration	20%	69+ days	9 of 46 done	CRITICAL
CoreLend Migration	0%	42+ days	0 of 1 done	CRITICAL
Reports Processing	0%	62+ days	0 of 13 done	HIGH
Reports Management	0%	41+ days	0 of 9 done	HIGH

Team Velocity Analysis

DEVELOPER	TICKETS	COMPLETED	RATE	FOCUS
Dev Gamma	28	0	0%	Reports migration
Dev Beta	23	10	43%	Core features
Dev Eta	7	7	100%	Operational tasks
Dev Delta	5	2	40%	Infrastructure
Dev Zeta	4	3	75%	Configuration
Dev Epsilon	4	0	0%	Data migration

> **The Bottleneck:** 55% of engineering effort is tactical (firefighting). Only 16% is strategic (migration/modernization). One team member holds 28 tickets with 0% completion, blocking $13,500/month in manual report costs. Redistributing that work alone has a $162K annual ROI.

SECTION 06

Hotspot Analysis

Hotspots are files where *change frequency, ownership concentration, and security risk* intersect. These are the places where a single bad commit is most likely to cause a production incident — and where that incident will be hardest to diagnose and fix.

Top Hotspots by Risk

FILE	CHANGES	CHURN (LOC)	BUS FACTOR	SECURITY	RISK
`WebPortal/WebPortal.csproj`	186	9,819	2	Low	CRITICAL
`WebPortal/MainSection.aspx.cs`	179	9,317	1	Medium	CRITICAL
`CoreEngine/CoreEngine.csproj`	154	5,620	2	Low	CRITICAL
`CoreEngine/ProcessDisclosure.cs`	114	22,514	1	CRITICAL	CRITICAL
`CoreLend/CoreLend.csproj`	131	14,346	2	Low	CRITICAL

The Security-Instability Death Spiral

The most frequently changed file in the codebase (`ProcessDisclosure.cs` , 114 commits, 22,514 lines of churn) also contains 4+ critical security findings including hardcoded cloud credentials. Frequent changes to compliance-critical code are *introducing* hardcoded credentials — a security-stability death spiral where every fix creates new exposure.

Risk Scorecard

FILE	BUS FACTOR	BLAST RADIUS	COMPLEXITY	SECURITY	OVERALL
`MainSection.aspx.cs`	1	High	High	Medium	CRITICAL
`ProcessDisclosure.cs`	1	High	CRITICAL	CRITICAL	CRITICAL
`CoreEngine.csproj`	2	CRITICAL	Medium	Low	CRITICAL

> **Compound Risk:** `CoreEngine.csproj` is referenced by ClientAccess, PartnerPortal, WebPortal, and the data warehouse. A single configuration change in this file breaks *multiple subsystems*. It has been changed 154 times — and every change is a roll of the dice.

SECTION 07

Strategic Options

Based on the consolidated findings, we present four paths forward — each with a clear cost, timeline, risk reduction profile, and ROI projection.

Option A: Targeted Fix — "Stop the Bleeding"

PARAMETER	VALUE
Total Cost	**$31K**
Timeline	4–6 weeks
Risk After	~5.5 (MEDIUM)

Fix only the critical security vulnerabilities and immediate compliance failures. Best for organizations that need to address imminent audit/breach risk while planning a larger effort.

Option B: Comprehensive Remediation — "Fix What We Have"

PARAMETER	VALUE
Total Cost	**$720K**
Timeline	12 months
Risk After	~2.0 (LOW)

Full remediation within the existing architecture: zombie cleanup, package standardization, security hardening, compliance program, and framework migration. Best for teams committed to keeping the current system 3+ years.

Option C: Platform Rebuild — "Start Fresh"

PARAMETER	VALUE
Total Cost	**$1.83M**
Timeline	18 months
Risk After	~1.0 (MINIMAL)

Greenfield rebuild on a modern stack using the existing system as a specification. Security and compliance built in from day one. Best when the current codebase is too far gone to modernize incrementally.

Option D: Hybrid — "Fix Critical, Rebuild Core" (Recommended)

PARAMETER	VALUE
Total Cost	**$1.02M**
Timeline	12 months
Risk After	~1.5 (LOW)

Immediate security fixes + targeted rebuild of highest-risk modules, while hardening lower-risk modules on the existing stack. Best balance of investment, risk reduction, and timeline.

Financial Comparison

OPTION	INVESTMENT	TIMELINE	RISK AFTER	5-YEAR ROI	BREAK-EVEN
Do Nothing	$0	—	7.5 (HIGH)	-$4.2M to -$13M	—
A: Targeted Fix	$31K	6 weeks	~5.5	7,000%–22,500%	< 1 month
B: Comprehensive	$720K	12 months	~2.0	440%–1,600%	4–11 months
C: Rebuild	$1.83M	18 months	~1.0	120%–590%	9–27 months
D: Hybrid (Rec.)	$1.02M	12 months	~1.5	290%–1,130%	5–15 months

> **Our Recommendation:** Option D (Hybrid) delivers the best risk-adjusted return. Start with Option A critical fixes immediately ($31K, 6 weeks) to stop the bleeding, then execute the core rebuild in parallel. The combined approach reduces composite risk from 7.5 to 1.5 within 12 months.

SECTION 08

What Accelion Delivers

This sample report represents *one section* of a complete Accelion Intelligence engagement. In two weeks, we deliver five stories backed by evidence from your codebase — not opinions, not assumptions. Here's the full picture:

The Five Stories

STORY	WHAT IT ANSWERS	REPORTS INCLUDED
Architecture Story	What does the system actually look like vs. what people think?	Ghost Feature Detection, Schema Impact Analysis, API Surface Assessment, Logic Location Map, Vendor Lock Inventory
Knowledge Story	Who knows what? Where is institutional memory concentrated?	Hero Dependency Map, Bus Factor Analysis, Archaeology Ratio, Feature Business Logic Health
Risk Story	Where is risk concentrated and what's the business impact?	Security Assessment, Hotspot Risk Analysis, Explosion Radius, No-Fly Zones, Test Coverage Gaps
Velocity Story	Where do engineering cycles actually go vs. the roadmap?	Cost of Waiting, Team Performance, Velocity & Roadmap Analysis, Blocked Initiative Analysis
Investment Story	Is budget going where the strategy says it should?	Revenue Risk Mapping, Modernization Scorecard, Strategic Options & Risk Consolidation

Accelion Intelligence Platform

Beyond the reports, every engagement includes access to the Accelion platform — a searchable, interactive knowledge base built from your codebase:

Interactive Knowledge Base

- AI-powered codebase exploration
- Entity drill-down and relationship mapping
- Contextual chat with code-aware AI
- Saved queries and shareable insights

Risk & Priority Dashboard

- Real-time risk scoring by module
- Compliance gap tracking (SOC 2, PCI, GLBA, NIST)
- Dependency vulnerability monitoring
- Strategic options modeling

How It Works

PHASE	TIMELINE	WHAT HAPPENS
Connect	Week 0	Read-only access to repos, tickets, docs + one 30-min stakeholder call
Map	Week 1	Correlate commits to tickets to requirements. Build knowledge graphs, topology analysis, risk models.
Deliver	Week 2	Executive briefing (60–90 min), five narratives with evidence, risk map, strategic options, searchable knowledge base.

Two weeks. Five stories. One decision.

Fixed-price engagement scoped to repository complexity. No retainers, no subscriptions. All deliverables remain client-owned.

foundersled.studio