



ProtoComet

INDEPENDENT TECHNOLOGY CONSULTING

MODERNIZATION READINESS TOOL

The .NET Modernization Readiness Checklist

20 questions to find out how close — or how far — your legacy .NET estate is from being modern, cloud-ready, and AI-ready.

1. Who This Is For

CTOs, VPs of Engineering, and IT Directors at organizations still running .NET Framework, WebForms, WCF, or other legacy Microsoft stacks who need a fast, honest read on modernization readiness before committing budget.

2. How To Use It

Work through each section with your engineering leads. Check every box that's true today. Tally your score at the end. It tells you which modernization tier to start with.

3. The Checklist

3.1 Application & Architecture Health

Where your codebase and architecture actually stand today.

- ☐ **Framework currency:** We know exactly which .NET Framework / .NET Core / .NET version(s) every production application runs on.
- ☐ **EOL exposure:** None of our business-critical applications run on a .NET Framework version that is past (or nearing) end-of-life support.
- ☐ **Architecture clarity:** We have an up-to-date architecture diagram for our core applications (not just tribal knowledge in someone's head).
- ☐ **Monolith awareness:** We know which applications are monoliths, which are already service-oriented, and where the tightly-coupled risk areas are.
- ☐ **Legacy patterns:** We've identified where WebForms, WCF, ASMX, or other legacy presentation/service patterns are still in active use.
- ☐ **Technical debt visibility:** We have a documented (even informal) list of known technical debt and the risk each item poses.
- ☐ **Dependency mapping:** We understand the third-party libraries, NuGet packages, and COM dependencies our applications rely on and which are outdated or unsupported.

3.2 Data & Infrastructure

The foundation everything else gets built on.

- ☐ **Database platform:** We know whether each application's database can move to Azure SQL / Managed Instance with minimal rework, or needs significant re-engineering.
 - ☐ **Data coupling:** We understand how tightly our application logic is coupled to stored procedures, SSIS packages, or on-prem-only data services.
-

- ☐ **Hosting footprint:** We have a clear inventory of on-prem servers, VMs, and IIS instances currently hosting these applications.
- ☐ **Hybrid readiness:** We've assessed whether a hybrid (on-prem + Azure) model is needed during transition, and what that would require.
- ☐ **Backup & DR:** Our current disaster recovery and backup strategy is documented and tested — not assumed.

3.3 Security, DevOps & Observability

The operational maturity that determines how safely you can modernize.

- ☐ **Identity modernization:** Authentication is (or can be) moved to a modern identity provider (Entra ID / OAuth2 / OIDC) rather than legacy Windows auth alone.
- ☐ **CI/CD maturity:** We have automated build and deployment pipelines for our core applications (not manual deploys).
- ☐ **Test coverage:** Our legacy applications have meaningful automated test coverage, or we know exactly where coverage is missing.
- ☐ **Monitoring & logging:** We have centralized logging and monitoring in place today, or know it would need to be built during modernization.
- ☐ **Security posture:** We've had a recent security/vulnerability review of these applications and their dependencies.

3.4 AI & Future Readiness

Whether your platform could actually take advantage of AI once modernized.

- ☐ **API surface:** Core business logic is exposed (or exposable) through APIs that an AI agent or service could call — not buried inside a UI layer.
- ☐ **Data accessibility:** Our business data is structured and accessible enough to feed AI/ML pipelines, not siloed in proprietary or undocumented formats.
- ☐ **Governance readiness:** We have (or are building) a framework for responsible AI use, data privacy, and model governance.

4. Score Yourself

For each section above, count the boxes you checked — "Yes" — and tally a total below. Use this scale per section: 0 = none apply, 1 = a few apply, 2 = most apply, 3 = all apply.

Section	0	1	2	3
Application & Architecture Health	☐	☐	☐	☐
Data & Infrastructure	☐	☐	☐	☐
Security, DevOps & Observability	☐	☐	☐	☐
AI & Future Readiness	☐	☐	☐	☐

Total Score: _____ / 12

5. What Your Score Means

Score	Tier	What It Means
0–3	Early-Stage Risk	Your estate carries real EOL, security, and operational risk. Start with a structured Legacy Health Check before any migration conversation — you need visibility before you can plan.
4–7	Modernization-Ready, Unplanned	You have partial visibility and some modern practices in place, but no costed roadmap. A Modernization Assessment will turn what you know into a prioritized, fundable plan.
8–10	Architecture-Ready	Your foundation is solid enough to move into target-state architecture and design work — the blueprint stage before implementation.
11–12	AI- and Cloud-Ready	Your platform is in strong shape. Focus shifts to governance, FinOps, and unlocking AI capabilities on top of what you've already modernized.

NEXT STEP

This checklist gives you a directional read, not a verified score. The fastest way to get a precise, evidence-based picture of your .NET estate is ProtoComet's AI Readiness & Legacy Health Check, a **1–3 week engagement** that combines automated codebase scanning with manual review to produce a cloud-readiness score, an AI-readiness score, and a prioritized modernization backlog.

6. About Protocomet

Protocomet is led by Yogesh Verma, who brings nearly two decades of hands-on .NET and Azure experience — he's written the kind of code you're now trying to modernize. Every engagement is guided by someone who has lived every stage of that journey, from legacy development through modern cloud-native architecture.

Write to us or book your free 30-minute discovery call — protocomet.com