UKAD

.NET Application Health Checklist

Practical checks for long-term reliability

Prepared by UKAD

WA, USA
+1 425 382 1212
hi@ukad-group.com

Introduction

An existing .NET application can appear stable while maintenance risks build quietly in the background. This checklist helps you identify where attention is needed first.

Who this checklist is for

- CTOs and engineering leads responsible for existing .NET applications
- Product owners planning ongoing maintenance or modernisation
- Digital agencies supporting long-running client solutions
- Teams maintaining legacy .NET Framework or ASP.NET codebases
- Organisations without dedicated in-house .NET expertise
- Businesses preparing for growth or technical due diligence

Why maintenance risks build up

Common patterns in established .NET solutions:

- Framework support ends without a clear upgrade plan
- Dependencies remain outdated between releases
- Recurring incidents hide deeper causes
- Monitoring gaps delay issue detection

What this checklist helps you do

Use it to:

- Identify areas that need attention
- Set priorities for the next maintenance cycle
- Prepare a focused technical discussion
- Review progress over time

WA, USA
+1 425 382 1212
hi@ukad-group.com

The Checklist

1. Framework & Security

- ☐ Confirm the current .NET / .NET Framework / ASP.NET version and support status
- ☐ Identify runtime and OS end-of-life risk
- ☐ Check target framework compatibility across the solution
- ☐ Scan NuGet packages for known vulnerabilities
- ☐ Review exposure to the OWASP Top 10 vulnerabilities
- ☐ Confirm security patches are applied
- ☐ Review authentication and data-protection settings

2. Performance & Data

- ☐ Profile response times and locate bottlenecks
- ☐ Review database query performance
- ☐ Check index usage for frequently accessed data
- ☐ Assess caching where repeated work affects response time
- ☐ Review memory usage under normal load and check for memory leaks
- ☐ Confirm capacity for expected traffic

3. Codebase & Delivery

- ☐ Identify tightly coupled or hard-to-test modules
- ☐ Review automated test coverage
- ☐ Flag areas of technical debt
- ☐ Check build reliability
- ☐ Review release frequency and failure patterns
- ☐ Compare development, staging, and production settings
- ☐ Confirm rollback procedures are documented

4. Operations & Maintainability

- ☐ Confirm logging and error tracking are in place
- ☐ Check that alerts reach a named owner
- ☐ Test backup and recovery procedures
- ☐ Review cloud resource usage and cost
- ☐ Confirm technical documentation is current
- ☐ Identify knowledge-transfer risk in legacy modules
- ☐ Assign ownership for maintenance decisions

WA, USA
+1 425 382 1212
hi@ukad-group.com

How to use the checklist

Rate each area

Review every item against evidence from the current environment.

- Green - working as expected and monitored
- Amber - needs planned improvement
- Red - affects operations or increases delivery risk
- Add a short note for every Amber or Red item
- Assign an owner before setting a target date
- Review the checklist after major releases

What good looks like

Healthy maintenance practices are visible in daily work:

- The framework remains within support
- Critical dependencies stay current
- Production issues are visible through monitoring
- Recovery procedures are tested
- Releases do not depend on one person

Common warning signs

These signals deserve closer review:

- The runtime is approaching end of support
- Security updates are repeatedly delayed
- The same incidents return without a root-cause fix
- Releases depend on manual steps
- Backups exist but recovery has not been tested

WA, USA
+1 425 382 1212
hi@ukad-group.com

Turning findings into a plan

The checklist gives you a starting point. The next step is to turn the findings into clear maintenance priorities.

Start with what affects operations

- Address Red items before work that can wait
- Group related issues into one manageable workstream
- Confirm what evidence is still missing
- Agree who will own each decision
- Set a date for the next review
- Review progress after the next release

When to ask for expert input

A health check is useful when:

- Several areas are marked Amber or Red
- The current framework is no longer supported
- Recurring incidents have no clear cause
- Planned changes carry too much uncertainty

What you should receive

A useful assessment should explain:

- Which issues need attention first and why they matter
- What can be handled through regular maintenance
- Where specialist input is required
- What the practical next step should be

WA, USA
+1 425 382 1212
hi@ukad-group.com

Need a clearer view of your .NET application?

UKAD's experienced .NET engineers review the current setup and identify where attention is needed first.

Request a free .NET health check

- A focused review of the current application
- A concise summary of the main findings
- Practical recommendations for the next step
- No obligation to continue

Contact us:

hi@ukad-group.com

Book a free consultation call with Alexander

Alexander Razvalynov, CTO

- LinkedIn
- Book a call