NexusBond

How to avoid project failure

THE 2026 SOFTWARE PROJECT BUYER'S GUIDE

Applications / Websites / Internal Systems

2 IN 3 SOFTWARE, APP & WEBSITE PROJECTS FAIL.

That's not the exception. That's the default

66% of software projects fail or fall short. Only 31% succeed.

Standish Group CHAOS Report. Based on 50,000+ projects globally.

45% over budget. 56% deliver less value than promised.

Oxford & Mckinsey, analysis of 5,400+ large IT projects.

Two-thirds of large-scale tech programmes miss targets on time, budget, or scope.

BCG, 2024. Survey of 1,000+ C-suite executives.

So why does this keep happening?

Teams never align on what they want until they see it and test it.

Briefs get written. Diagrams get drawn. **Assumptions** get made. Then, six months later and with no budget left, the first working version arrives, and it's not what anyone actually wanted.

This isn't new. Software, websites and apps have failed this way since long before AI.

M&S spent £150M on a botched eCommerce relaunch. TSB's migration disaster cost over £400M. Birmingham City Council's Oracle implementation overran by more than £100M in the last two years.

Unlimited budgets, top-tier agencies, experienced leaders, and still, **projects fail at every scale**.

For the first time, AI makes that solvable. Teams can now see, click and test what's being built in weeks, before committing to the full build.

Most businesses aren't taking advantage of this. The capability is here. The workflow often isn't. And when that gap shows up, it's the decision maker who commissioned the work who pays the price, the launch, the budget, the credibility, sometimes the job.

This playbook exists so it doesn't happen to you.

What follows is what every business leader should know before starting a project: what's actually changing in 2026, what to look for in a development partner, which AI tools matter, and how our own process makes project failure far less likely.

AI IS CHANGING HOW BUSINESSES BUILD SOFTWARE.

Despite the failure rate, AI is genuinely transforming how software gets built when it is properly applied. Here is what the new landscape actually looks like.

If you have a software project on the horizon, an app idea taking shape, or a website that needs rebuilding, **the landscape has shifted underneath you**.

AI is now part of how modern development teams work. It helps with everything from gathering requirements to writing code to running tests. The process looks very different to how it did even two years ago.

For business leaders, this matters for two reasons:

First, the speed. Projects that used to take six months can now be delivered in weeks. Custom software, cloud platforms, and digital tools that were once only viable for large enterprises are now within reach for growing businesses.

Second, the risk. Businesses that do not adapt to this shift will fall behind. The gap between those who embrace AI-powered development and those who wait is widening every month.

AI is a game-changer for how software gets built.

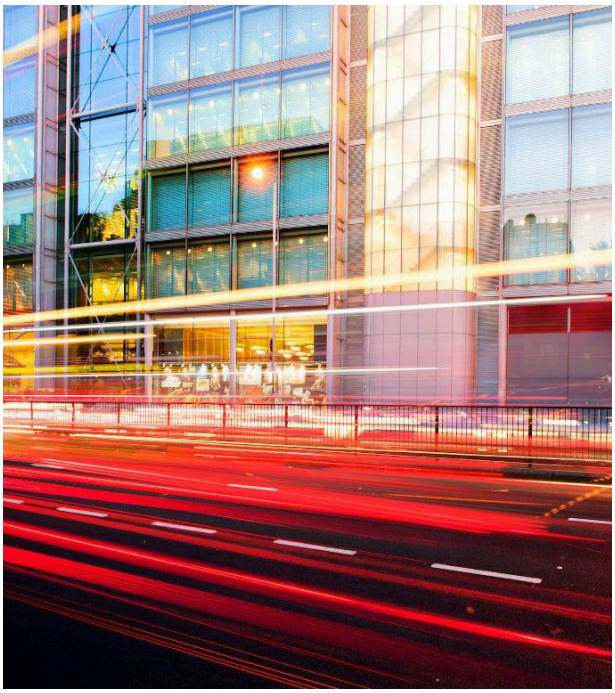
For business leaders, this means: faster delivery, **better results**, and more **predictable costs**. Projects that used to be out of reach are **now realistic**.

On the following pages, you will learn how **AI is reshaping development**, what it means for **your next project**, and what to look out for.

WHAT'S INSIDE

"Over 12 years, we have delivered hundreds of digital projects for established SMEs to enterprise brands like Bridgestone and CH&Co. This guide shares what we have learned about how AI is reshaping the way software gets built, and what it means for your next project."

Marco Navarro, Managing Partner, NexusBond



FASTER TIME TO MARKET

Many businesses struggle with software, website, and app projects that **drag on for months**. Deadlines slip. Budgets get exceeded. The launch date keeps moving. And by the time the product finally goes live, the market opportunity has already shifted.

A central reason: too many assumptions made too early, too many manual steps, and too much time spent building things that turn out not to be what anyone actually wanted.

How AI changes this

AI is a **power tool** in the hands of a skilled developer. It can scaffold in hours what used to take weeks of manual coding. The developer then refines and integrates that into a working version that can be seen, clicked, and **tested in weeks, not months**.

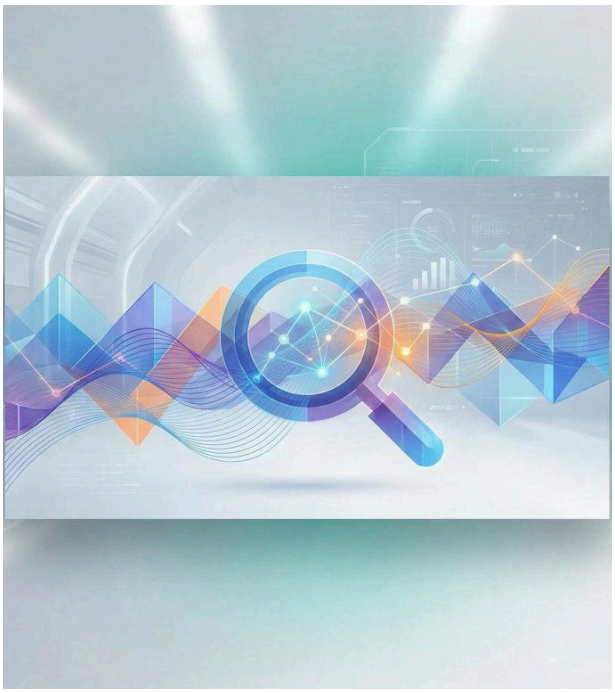
This matters for two reasons.

First, **less wasted work**. Teams catch misalignment while it is still cheap to change. If a feature does not match what was intended, it gets corrected in days, not after months of full development.

Second, **more time on what matters**. Developers focus on architecture, quality, and the business-critical decisions, while AI handles the repetitive scaffolding work.

> KEY TAKEAWAYS
>
> - AI reduces development time by removing manual, repetitive work
> - Smaller teams can match or exceed the output of larger traditional teams
> - Projects become more predictable and are completed faster
> - The advantage goes to teams who know how to use AI properly, not to AI alone

If your last project took six months, the same scope can likely be **delivered in a fraction of that time** today.



BETTER REQUIREMENTS ANALYSIS

In most businesses, the people who know the business and the people who build the software do not speak the same language. Whether it is a website, a mobile app, or a custom cloud platform, requirements get written in documents that **nobody fully reads**. Assumptions get made. Details get lost. And by the time the first version is delivered, it does not match what anyone had in mind.

This is one of the biggest reasons software projects go wrong. Not because the developers are bad, but because the brief was never properly understood by anyone.

How AI bridges the gap

AI tools allow business requirements to be expressed in plain English and **rapidly translated** into functional designs or working code. You describe what you want. AI builds a first version. You see it, click through it, and say "yes, that is right" or "no, change this."

This creates a feedback loop that used to take weeks of meetings and revision rounds. With AI, it can happen **in hours**.

Even more powerful: interactive, working versions of a product can now be produced in the **early stages** of a project. Ideas can be tested with real users, feedback gathered, and concepts validated before committing serious budget.

This gives early confidence and **eliminates the expensive correction cycles** that plague traditional projects.

> KEY TAKEAWAYS
>
> - AI translates requirements into working software faster
> - Faster feedback loops prevent expensive misunderstandings
> - Ideas can be validated before committing full budget

Better communication, fewer surprises, and significantly higher confidence that what gets built is what was actually wanted

THE PROBLEM WITH AI IN DEVELOPMENT

AI in software development is real, and the productivity gains are genuine. But there's a story nobody's telling the boardroom.

The same tools that can deliver in weeks what used to take months can also produce code that looks professional, appears to work and run, but **quietly falls apart** in production. The difference isn't the AI agent used. It's whether a team of seniors are actually on the driver's seat.

How AI gets misused in 2026

Briefs get fed into Claude or GPT and the output sent back as requirements. Code gets generated and committed **without senior review**. Junior developers accept AI output they don't fully understand. The work looks like progress.

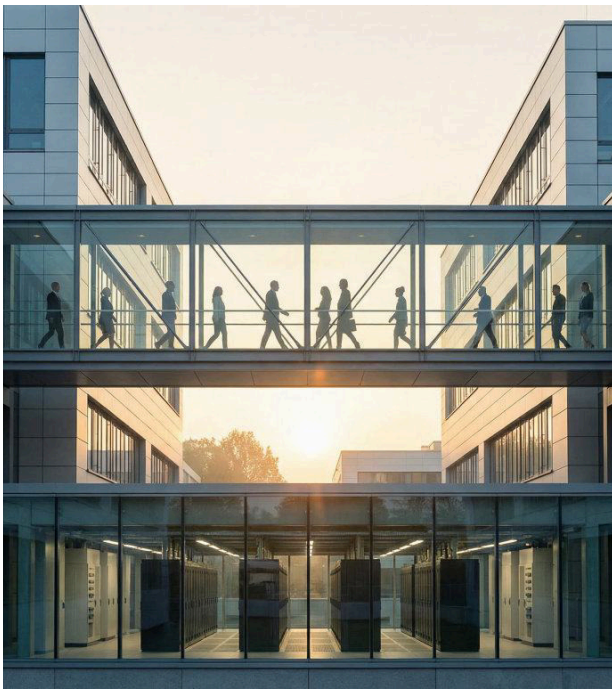
Then the cracks appear. AI hallucinates functions that don't exist. It writes code with security holes; SQL injection, exposed credentials, authentication bypasses, that pass review when nobody's actually looking. Performance falls over at scale. Architectural decisions get outsourced to a language model that **doesn't understand your business**.

A senior developer catches what AI misses: the security flaw, the scalability wall, the assumption that holds today and breaks next quarter. AI accelerates the work. The senior developer makes the calls and **uses AI as a power tool**.

KEY TAKEAWAYS

- Unsupervised AI creates expensive, delayed failure modes
- Hallucinations, security flaws, and architectural drift are the top risks
- Senior developer review on every AI output is non-negotiable
- Relying on AI solely increases project failure risks drastically

Developers who understand your business stay in the driver's seat. AI is the power tool, fast, sharp, only as **useful as the person holding it**.



LOWER RISK, HIGHER QUALITY

In practice, proper testing is often the first thing to get cut when budgets run tight or deadlines loom. Teams skip it because it takes time and costs money. But this always comes back to bite.

Bugs slip through. Features break when new ones are added. The site that worked perfectly in staging falls apart in production. Fixes become emergency patches that **cost three times** what proper testing would have cost in the first place.

How AI changes the equation

AI can now generate test cases automatically, based on the requirements of a project. It checks code for errors before they reach production. It identifies edge cases that manual testing might miss. And it does all of this **faster and cheaper than manual QA**.

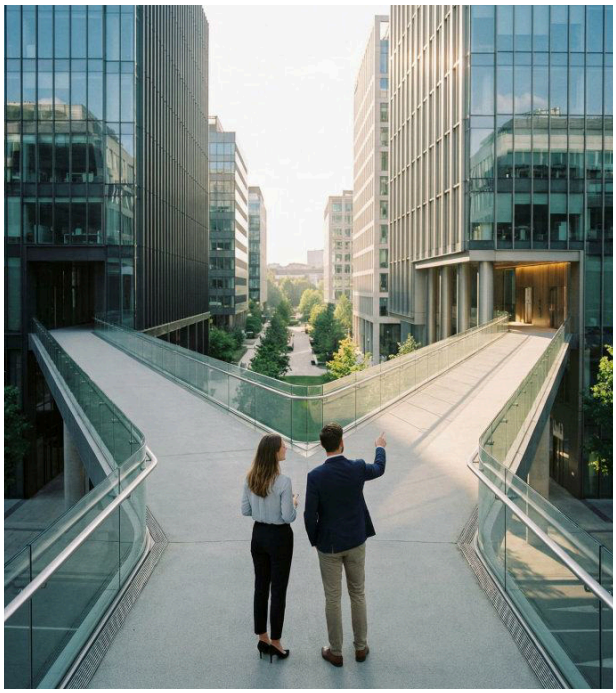
The result is software that is more stable, more reliable, and significantly cheaper to maintain over time. Changes become less risky because automated test suites catch problems immediately.

For any business investing in software, this matters. It means a website, app, or custom system does not become a **maintenance nightmare** six months after launch. It stays clean, stable, and easy to extend when business needs change.

KEY TAKEAWAYS

- AI-powered testing provides more coverage with less effort
- Bugs are caught early, before they become expensive to fix
- Software stays maintainable and stable long-term
- Future changes and updates become lower risk

Quality becomes built-in rather than bolted-on. **Maintenance becomes predictable** rather than a constant source of surprises.



HOW TO CHOOSE THE RIGHT AGENCY

The agencies that will deliver the best results over the next few years are the ones that have **genuinely integrated AI** into how they work. Not as a marketing buzzword, but in their actual day-to-day process.

When evaluating a development partner for your next project, **here is what to look for**.

✓ They show you results early

A modern development partner should be able to show you something **tangible within the first few weeks**. Something you can see, click through, and test. Early visibility means fewer surprises down the line.

✓ They speak your language

Your development partner should **understand your business**, not just your technology. It is not enough to write good code. They need to understand what that code is supposed to achieve for your customers and your bottom line.

✓ They offer fixed pricing

AI has reduced development risk significantly. This makes **fixed-price projects realistic** again. Predictable costs, clear scope, and defined timelines should be the standard, not the exception.

✓ They can explain their process

A clear, **repeatable process** is a sign of a mature operation. Defined stages, clear outputs at each step, and regular checkpoints. **AI should be part of how they work**, not the whole answer.

KEY TAKEAWAYS

- Look for partners that show working results early
- Choose partners who understand your business goals, not just technology
- AI makes fixed-price projects realistic. Expect predictable costs.
- A clear, repeatable process is a sign of quality

NexusBond

INTRODUCING

ignite by NexusBond

A structured, **four-phase process**, Discovery, Alpha, Beta, Live, with defined outputs at every stage. AI powered for maximum efficiency. No surprises.

PHASE 1 OF 4

DISCOVERY CALL

Where we learn about your project, and you decide if we're the right fit.

01

Discovery Call

We learn about your project, goals, and challenges. Together we decide if Ignite is **the right fit**.

Output: *Exact timeframe and fixed price to get started*

PHASE 2 OF 4

ALPHA

3 sub-stages
Define, design, and prove it works, before you commit to the full build.

02

Requirements Analysis

We go deep into your pages, user journeys, and business logic. **Everything is documented. Nothing is assumed.**

Output: *Detailed requirements document*

03

Prototype

We build a **fully interactive, clickable version** of your product. Every key screen, every core flow, desktop and mobile. See what you're building **before committing** to the full development.

Output: *Working prototype*

04

Test, Review & Adjustments

We present the prototype to you and your stakeholders. You test it, share it, and request changes. Revisions are turned around in **hours, not weeks**. You also receive a **fixed-price proposal** for the full build.

Output: *Approved prototype + build proposal with fixed pricing*

PHASE 3 OF 4

BETA

Build the production-ready product. Fixed scope. Fixed price.

05

Full Build

With the prototype approved, we develop the **production-ready product**. Fixed scope, fixed price, exactly what you signed off on.

Output: *Production-ready product*

PHASE 4 OF 4

LIVE

Deploy, hand over, and go live. It's yours.

06

Launch & Handover

We deploy, test, and hand over everything. Documentation, code, assets. **It's yours**.

Output: *Live product + Training + Documentation*

Learn more about Ignite and see if it's right for your project.
www.NexusBond.com/ignite

WHICH AI TOOL FITS YOUR PROJECT?

The AI development landscape is evolving fast. New tools appear every month, each promising to change how software gets built. But not all tools are equal, and the right choice depends on your project, your team, and your goals.

Before evaluating tools, consider where your biggest challenge is:

- **Speed**: Do you need to get an idea to market fast?
- **Quality**: Do you need production-grade, maintainable code?
- **Design**: Does visual precision matter more than functionality right now?
- **Simplicity**: Do you need something non-technical people can use?

What to look for when choosing:

1. **Team fit**: Can your team use it, or does it require experienced developers?
2. **Scalability**: Will it carry you past prototype into production?
3. **Integration**: Does it work with your existing systems and workflows?
4. **Cost:** What is the licensing model as your needs grow?

On the following pages, we break down the five most relevant tools in the market right now, by use case, strengths, and limitations.

Practical Tip

Don't commit to one tool based on a demo or a blog post. **Test two or three** in parallel on a small, low-stakes project. You will quickly see which one fits your team and your workflow best.

Claude

by Anthropic • Production-grade AI development

Best for: Complex applications, large codebases, architectural decisions

Strengths

+ Exceptional reasoning and problem-solving ability
+ Handles large, complex codebases with ease
+ Excellent at understanding business logic and translating it into architecture
+ Produces clean, maintainable, production-ready code
+ Strong at debugging and refactoring existing systems

Limitations

– Command-line interface requires technical comfort
– Not a visual tool. No drag and drop.
– Requires experienced developers to guide it effectively
– Best results come from precise, well-structured prompts

Takeaway:

The go-to for complex, production-grade projects. If you need custom business logic, complex integrations, or a system that will need to scale, this is the engine behind it. Not flashy, but incredibly capable. The workhorse of our stack.

Who should care: CTOs, technical founders, and any business building something that needs to be robust, scalable, and maintainable long-term.

CURSOR

AI-powered code editor • The developer's companion

Best for: Day-to-day development, code editing, real-time AI assistance

Strengths

+ Integrated directly into the coding environment
+ Real-time AI suggestions as you type
+ Excellent at refactoring and improving existing code
+ Smart error detection and auto-fixes
+ Multi-file editing across entire projects

Limitations

– Requires solid technical knowledge to use effectively
– Learning curve for teams new to AI-assisted development
– Can occasionally suggest incorrect patterns if not supervised
– Best with a developer in the driving seat

Takeaway:

The most widely adopted AI code editor on the market. It makes good developers faster and helps them catch mistakes they would otherwise miss. Think of it as a co-pilot that never sleeps. Essential for any team that writes code regularly.

Who should care: Technical teams, in-house developers, and agencies that want to accelerate their existing workflow without changing their tools completely.

Windsurf

by Codeium • Rapid full-stack development

Best for: Fast prototyping, full-stack applications, getting from idea to working product quickly

Strengths

+ Extremely fast at scaffolding entire applications
+ Great for building working versions quickly
+ Handles both frontend and backend in one flow
+ Integrated terminal and file management
+ Strong for iterative development with rapid feedback

Limitations

– Still maturing as a platform
– Complex integrations may need manual intervention
– Less established ecosystem than Cursor
– Requires developer oversight for production deployments

Takeaway:

The strongest option when speed is the priority. Need a working version of your product in days instead of weeks? This is what makes that possible. Particularly strong for early-stage builds where you want to validate an idea fast before investing heavily.

Who should care: Founders, product managers, and business leaders who want to see something working quickly. Ideal for the "build it and test it" mindset.

Figma

Industry standard • Pixel-perfect design

Best for: Projects where visual precision is critical. Branding, luxury, retail, design-led products.

Strengths

+ Unmatched design precision and control
+ Industry standard for collaboration between designers and developers
+ Interactive prototyping with realistic user flows
+ Complete design system management (typography, colours, components)
+ Clean handoff to any development team

Limitations

− Does not generate working code
− Requires skilled designers to use effectively
− The output still needs to be built by developers
− Slower than AI code tools for functional validation

Takeaway:

Figma is irreplaceable when the visual standard needs to be perfect. If your brand demands pixel-level precision, or if stakeholders need to see exactly how the final product will look before any code is written, Figma is the answer. We use it for projects where design quality is the top priority and offer it as a dedicated option within our Ignite process.

Who should care: Marketing directors, brand managers, and any business where the visual experience is a core part of the product value.

Lovable

No-code AI builder • Simple use cases

Best for: Quick internal tools, simple landing pages, basic apps with limited functionality

Strengths

+ No coding knowledge required
+ Very fast for simple, straightforward use cases
+ Built-in database integration (Supabase)
+ Good for internal tools and basic dashboards

Limitations

– Limited flexibility for custom requirements
– Output is template-driven, not truly custom
– Not suitable for production-grade, scalable applications
– Struggles with unique design or complex user flows
– You will quickly hit a ceiling if your project grows

Takeaway:

Lovable fills a niche for very simple use cases where speed matters more than quality or customisation. Think of it as a smarter website builder. If all you need is a basic landing page or a simple internal tool, it can get you there quickly. But if your project requires custom functionality, unique design, real integrations, or will need to scale over time, you will outgrow Lovable fast. For anything business-critical, we recommend the tools above.

Who should care: Non-technical teams who need something basic up and running quickly, and do not need it to scale or customise beyond the defaults.

Questions or feedback?
Drop us an email at
contact@nexusbond.com

Want to discuss your project?
Book Your Discovery Call here:
Book Discovery call →

NexusBond

NexusBond LTD • Company No: 09001165 • VAT: 206275524