

TOP 5 MISTAKES YOU WANT TO AVOID WHEN DEVELOPING A BESPOKE SOFTWARE:



Leif-Arild Åsheim, CEO
leifa@promineo.com
+47 901 55 853



Tor Åge Leknes, Product Manager
tor.aage.leknes@promineo.no
+ 47 982 48 220

1. Not utilizing AI to supercharge your software development

Not utilizing AI to supercharge your software development is a critical mistake. AI is transforming how software projects are designed, built, and delivered. Among all industries, software development is currently the field where AI is driving the greatest automation gains. The pace at which new AI tools and capabilities are being developed and adopted is breathtaking, reshaping the industry week by week. Failing to take advantage of what AI can offer your software development projects means missing one of the most powerful productivity shifts in modern technology.

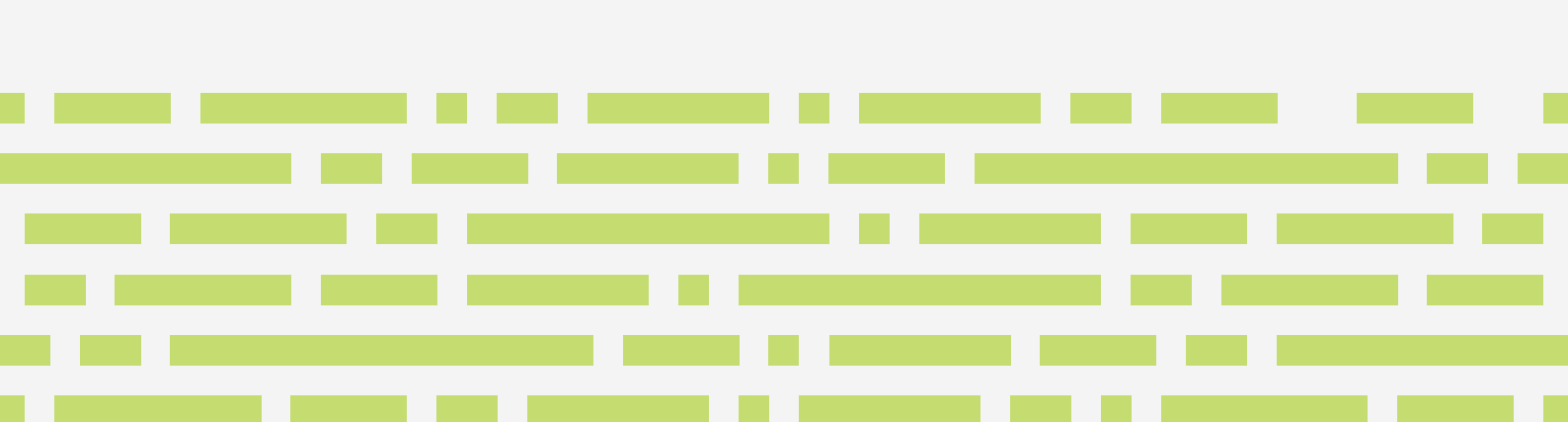
2. Lack of a shared objective for the software

A common pitfall in software development is when team members pull the project in different directions,

ultimately working against each other. While each person brings their own expertise and perspective—a key strength of any bespoke software team—these differences can also become a liability. That's why it's essential to invest time early on in aligning everyone around a shared vision. Without this, differing opinions can lead to a fragmented and incoherent solution.

3. Ignoring non-functional requirements

Ignoring non-functional requirements in software development can lead to serious consequences, even when all functional features are delivered as planned. One major risk is poor performance—the software may become slow, unresponsive, or unable to handle expected user loads, frustrating users and undermining its value. Another critical concern is security. Without clear security requirements, the system may be vulnerable to breaches or data loss, with



potentially severe legal and financial consequences. Additionally, reliability and availability often suffer when non-functional aspects are overlooked; frequent crashes or downtime can erode user trust and disrupt business operations. That's why we strongly recommend that your development team includes an experienced software architect who can ensure that non-functional requirements are identified and resolved.

4. Not creating an MVP first

Skipping the development of a Minimum Viable Product (MVP) is a common - and often costly - mistake in software projects. In bespoke software development, the consequences can be even more severe.

The goal of bespoke software is to deliver measurable business benefits and a strong return on investment. Achieving this doesn't require the same level of polish or feature completeness as commercial off-the-shelf software. Instead, a strict focus on the MVP - the smallest possible version of the product that still delivers core value - allows teams to build, test, deploy, and begin generating business impact as early as possible.

This early deployment makes it possible to collect real user feedback and iterate toward greater value. However, if development efforts stray beyond the MVP scope - adding features that are "nice to have" but not

essential to the value proposition - the result is often delayed delivery, budget overruns, or even project failure before reaching production.

On the other hand, trimming the product too far can lead to a version that is no longer viable - failing to meet business needs or user expectations.

Our recommendation: treat both minimum and viable with equal importance. Doing so ensures the MVP delivers real value quickly and cost-effectively, while laying a solid foundation for future growth.

5. Allow super users to take control of what you are developing

Allowing super users to drive software development is a sure way to make the product complex and inaccessible for first-time and everyday users. Super users tend to focus on enhancing advanced features that improve efficiency from their own perspective—but this often leads to a level of sophistication that overwhelms or alienates less experienced users. To avoid this, it's essential to have an experienced UX designer on the team—someone who understands these pitfalls and can strike the right balance between the needs of expert users and those of new or typical users.