HOLD MY CODE

Angular 8 auf 20 in einer laufenden Enterprise-Vertriebsplattform

Ein globaler Software-Dienstleister hat seine interne Vertriebsplattform in einem Jahr von Angular 8 auf Angular 20 gebracht. Über 5.000 Nutzer arbeiteten durchgehend weiter, ohne Downtime.

5.000+ Nutzer	**Ohne Downtime**	**−65 % Ladezeit**

Ausgangslage

Der Vertrieb arbeitet täglich mit der Plattform: Produkte suchen, Warenkorb füllen, Angebote und Bestellungen verwalten. Angular 8 war ohne Support, und die Anwendung war langsam und umständlich geworden.

Risiko

Bei einem Ausfall hätte der Vertrieb keine Angebote erstellen und keine Bestellungen aufgeben können. Außerdem musste das Upgrade laufen, während das Team weiter Features auslieferte.

Eine komplette Neuentwicklung stand zur Diskussion. Die Entscheidung fiel für schrittweises Refactoring.

Vorgehen

Das Upgrade lief der Reihe nach durch jede Major-Version, im laufenden System, über ein Jahr. Die Migrationsänderungen gingen in eigenen Releases live, abgesichert durch Tests.

Die Ladezeit sank um 65 %, durch Architektur-Refactoring, Code Splitting und Lazy Loading, gemessen mit den Chrome DevTools.

4 Entwickler wurden so begleitet, dass sie Angular-Upgrades und Feature-Entwicklung eigenständig weiterführten.

Dieses Projekt wurde ohne KI-Agenten umgesetzt.

Ergebnis

- Angular 20 statt Angular 8, in einem Jahr.
- Keine Downtime für über 5.000 Nutzer.
- 65 % weniger Ladezeit.
- 4 Entwickler führten Upgrades und Feature-Entwicklung nach dem Projekt ohne externe Hilfe weiter.

Was das für Sie bedeutet

Wenn Ihre Angular-Anwendung geschäftskritisch ist und mehrere Major-Versionen zurückliegt, prüfen Sie ein schrittweises Upgrade, bevor Sie sich für eine Neuentwicklung entscheiden. Dieses Projekt hat zwölf Major-Versionen im laufenden System hinter sich gebracht, während die Nutzer weiterarbeiteten.

Für neue Kunden beginnt diese Arbeit mit dem Modernisierungs-Review: 10 Arbeitstage, Festpreis und eine Roadmap mit Optionen und Aufwandsschätzung.