Vergleich: BOI FreeDa vs. Eigenentwicklung

Kriterium	BOI FreeDa	Eigenentwicklung
Time-to-Value	Produktiver Einsatz in wenigen Wochen möglich	Monate bis Jahr(e) für Konzeption, Implementierung und Härtung
Revisionssicherheit & Compliance	Out of the box: Audit Trail, Versionierung, historische Datenstände, nachvollziehbare Änderungen inkl. Benutzer-/Zeitstempel	Revisionsnachweise selbst implementieren und auditfest dokumentieren
Governance & Rollen	Granulare Rollen- und Rechteverwaltung	Eigenes Rollenmodell samt Administration aufbauen und pflegen
Pflege-Prozesse	Pflegeprozesse gemäß 2-, 4- und 6-Augen-Prinzip inkl. Änderungsansicht	Konzeption und Implementierung individuell erstellen
Validierungslogik auf Datenbank-Ebene	Automatische Sicherstellung der Datenbankkonformität aller Änderungen	Prüfungen selbst programmieren, testen, dokumentieren
Validierungslogik auf Tabellen-Ebene	Durch User konfigurierbare Regel- und Prüfmechanismen auf Tabellen-Ebene	Konzeption und Implementierung von Prüfungen selbst erstellen
Excel-Integration	Nahtlose Import/Export-Workflows	Individuelle Parser, Mapping, Fehlerbehandlung entwickeln
Datenbank-Anbindung	Alle gängigen relationalen Datenbanken werden unterstützt	Meistens nur für eine Datenbank geeignet
APIs & Integration	Standardisierte Schnittstellen für automatisierte Prozesse	Schnittstellen-Design, Versionierung, Rückwärtskompatibilität selbst umsetzen
UI/UX für Fachbereiche	Hohe Usability der Benutzeroberfläche, optimiert für Tabellenpflege für User ohne Datenbankkenntnisse	UI/UX Konzept und Komponenten selbst entwickeln, Barrierefreiheit sicherstellen

BOI BETTER ORGANIZED INFORMATION
BOI Software Entwicklung und Vertrieb GmbH

Office Linz
Spazgasse 4
4040 Linz, Austria

Office Graz
Friedrichgasse 30/1
8010 Graz, Austria

office@boi.at
www.boi.at
+43 (0) 732 736 423 - 0

Vergleich: BOI FreeDa vs. Eigenentwicklung

Skalierbarkeit & Performance	Bewährte Architektur für hohe Datenmengen (>10.000 Tabellen) und viele Concurrent User (>100)	Nicht funktionale Anforderungen (Last, Latenz) zusätzlich absichern
Betriebssystem	Plattformunabhängig	Meistens plattformabhängig umgesetzt, da kostengünstiger
Support & Wartung	Herstellersupport inkl. regelmäßige Produkt-Updates	Eigenes Team für Bugfixes, Feature Requests, Rufbereitschaft
Total-Cost-of-Ownership	Planbar: Lizenz + Einführung + Betrieb	Variabel/steigend: Entwicklung + Pflege + Wissensbindung
Risiko	Gering: produktiv erprobt, revisionssicher und stabil	Hoch: Termin-, Budget-, Qualitäts- und Personalausfallrisiken
Flexibilität	Hohe Konfigurierbarkeit: Erweiterbarkeit über APIs	Maximale Freiheit, aber hoher Implementierungsaufwand