

Vorwort

Erfahren Sie, was in diesem Buch auf Sie zukommt, welches Wissen wir Ihnen vermitteln werden und wie das Buch aufgebaut und gegliedert ist.

Da wir uns jeden Tag aufs Neue mit denselben Themen auseinandersetzen und die gleichen Diskussionen mit Kunden und Kollegen führen – nämlich wie eine Lösung in SharePoint auszusehen hat und welche Kriterien sie erfüllen muss –, haben wir unsere Erfahrungen in diesem Bereich zusammengefasst und auf Basis der wiederkehrenden Anforderungen ein Konzept entwickelt, das eine ordnungsgemäße Planung und Entwicklung ermöglicht und nahezu alle fachlichen Anforderungen mit den technischen Möglichkeiten von SharePoint 2019 abdeckt.

Die Planung und Entwicklung im SharePoint-Umfeld stellt den Consultant, aber auch den Entwickler oft vor spezielle Herausforderungen. Diese Herausforderungen wirken sich vor allem auf die Erstellung standardisierter Softwareprodukte aus.

Die Anforderungen, die ein Kunde an eine gewöhnliche Software stellt, sind simpel:

Eine Lösung, die als Produkt entwickelt wird, sollte

- ▶ einfach zu installieren sein,
- ▶ supportet werden,
- ▶ über Updatepfade verfügen,
- ▶ ein Handbuch enthalten
- ▶ und gegebenenfalls über definierte Releasezyklen verfügen.

Diese Anforderungen sind klar verständlich und nachvollziehbar. Unsere Erfahrung zeigt allerdings, dass im SharePoint-Umfeld noch weitere Anforderungen hinzukommen, die sowohl die Planung als auch die Entwicklung erschweren.

So sollte eine Lösung im SharePoint-Umfeld zusätzlich

- ▶ individuell anpassbar sein,
- ▶ möglichst auf allen Plattformen (Standard, Enterprise) lauffähig sein,
- ▶ mit anderen Drittherstellern kompatibel sein
- ▶ sowie vollständig in das Trägersystem integriert sein.

Was heißt das genau?

Der Kunde möchte innerhalb der Software möglichst nicht auf die Flexibilität von SharePoint verzichten. Er will also weiterhin in der Lage sein, neue Spalten in Listen anzulegen, WebPart-Seiten zu gestalten, Ansichten zu erstellen und vieles mehr. Dies ist zwar nicht immer zu gewährleisten, an der einen oder anderen Stelle können Sie

dem Kunden diese Flexibilität jedoch erhalten. Wie dies geht, werden wir im Umsetzungsteil an einigen Beispielen erläutern.

Das wirkt sich zum einen auf die Entwicklung von individuellen New-, Disp- und Edit-Forms, aber auch auf Updatepfade aus. Wenn der Kunde einer Liste eine neue Spalte hinzufügt, soll sie natürlich auch in seinen Formularen verfügbar sein. Somit ist der Entwickler in der Verantwortung, bei individuell gestalteten Formularen darauf zu achten, dass automatisch auch Felder angezeigt werden, die in der initialen Planung noch nicht vorgesehen waren.

Im Bereich des Softwareupdates entstehen weitere Probleme. Wenn ich eine neue Version mithilfe eines automatisierten Updates ausrollen möchte, ist es erforderlich, zu wissen, welche Teile der Anwendung zum Produkt gehören und welche vom Kunden angepasst wurden. Ist eine solche Unterscheidung nicht möglich, besteht die Gefahr, dass Anpassungen des Kunden durch ein Update überschrieben werden oder ebendiese Anpassungen ein sauberes Update verhindern.

Natürlich soll eine standardisierte Lösung auf allen verfügbaren Plattformen lauffähig sein. Da sich der Funktionsumfang von SharePoint Server und SharePoint Enterprise gravierend unterscheidet, ist es ratsam, eine Produktentwicklung, sofern eben möglich, auf Basis des SharePoint-Server-Lizenzmodells durchzuführen. Dadurch vergrößern Sie die Zielgruppe der Kunden, die Ihr Produkt einsetzen können, und steigern so Ihren Profit. Eine Tabelle zum Funktionsvergleich von SharePoint 2019 Standard und SharePoint 2019 Enterprise finden Sie in Anhang B, »Berechtigungsstufen«. In älteren SharePoint-Versionen sollten Sie hierzu möglichst auf SharePoint Foundation zurückgreifen, da diese Lizenz die kostengünstigste Variante für SharePoint als Trägersystem darstellt. Im weiteren Verlauf des Buchs werden wir, um die Verwendung der Ansätze unter SharePoint 2013 weiter zu ermöglichen, die Foundation-Version grundsätzlich mit berücksichtigen. Ist eine Funktion erst ab SharePoint Enterprise verfügbar, ist es gegebenenfalls sinnvoll, sie in Teilen oder vollständig für das aktuelle Projekt nachzuprogrammieren, wobei Sie hier sehr vorsichtig agieren und die Kosten im Blick halten sollten. Der Versuch, Funktionen nachzuprogrammieren, kann sich als absolute Kostenfalle und Fass ohne Boden herausstellen.

Als offenes System bietet SharePoint die Möglichkeit, Software verschiedener Dritthersteller in einem Portal zusammenzuführen – sei es die Verwendung verschiedener WebParts auf einer Seite, die Installation unterschiedlicher Komplettlösungen in einer *SiteCollection* oder der Einsatz einer individuellen *Masterpage*. Dabei besteht das Risiko, mit Software anderer Hersteller zu kollidieren. Hier müssen Sie Wege finden, diese Kollisionen so weit wie möglich zu verhindern.

Mit SharePoint als einheitlichem Portal stellt der Kunde selbstverständlich auch an unsere Lösung die Anforderung, dass sie sich vollständig in sein System integriert. Dieser Punkt bezieht sich weniger auf die Technik als vielmehr auf die optische Inte-

gration der Anwendung. Gerade die optische Integration einer Lösung ist mit hohem Aufwand und Liebe zum Detail verbunden. Trotz allem lohnt sich dieser Aufwand, da die optische Harmonie einer Anwendung einen großen Teil der Außenwirkung ausmacht und der ausschlaggebende Punkt bei der Kaufentscheidung des Kunden sein kann.

Diese Punkte stellen nur einen Teil der Herausforderungen dar, mit denen sich Consultants und Entwickler während der Erstellung einer SharePoint-Lösung auseinandersetzen müssen. Dieses Buch soll als Leitfaden dienen, eine stabile Lösung zu planen, zu entwickeln und dabei allen Anforderungen gerecht zu werden.

Das Buch ist in zwei Bereiche geteilt und spricht Projektleiter, Softwarearchitekten, SharePoint-Consultants sowie SharePoint-Entwickler an. Der erste Teil des Buchs behandelt die Planung einer Lösung und ist für alle Zielgruppen gleichermaßen interessant. Der zweite Teil richtet sich in erster Linie an SharePoint-Entwickler und erläutert die Umsetzung der geplanten Anwendung unter Berücksichtigung der vorab erstellten Lösungsarchitektur. In diesem Teil werden wir Ihnen anhand klarer Entwicklungsbeispiele nützliche Tipps und Tricks für die Entwicklung an die Hand geben. Die gezeigten Programmierbeispiele des zweiten Teils werden in der Programmiersprache C# abgebildet, beziehen sich auf die On-Premises-Entwicklung für SharePoint Classic Pages und sind nicht für das Hosting in SharePoint Online geeignet.

Begriffe

Eine *SiteCollection* (eine Websitesammlung) ist die Menge an logisch zusammenhängenden Seiten, die ein Portal bilden. Die am höchsten gegliederte Seite dient als Einstiegspunkt in das durch diese SiteCollection abgebildete Portal und wird *Root-Web* genannt.

Der Begriff *Masterpage* kommt aus der ASP.NET-Entwicklung. Unter einer Masterpage versteht man eine Vorlagenseite, mit deren Hilfe sich grundlegende UI-Strukturen und globale Komponenten für alle Detailseiten vorgeben lassen, die auf diese Masterpage referenzieren.