

Grundstein für eine solide .NET Enterprise-Architektur

Der Microsoft „Patterns & Practices“-Leitfaden hilft auf anschauliche und praxisorientierte Weise, den Grundstein für eine tragfähige Unternehmenslösung zu legen. Dank des reichhaltigen Erfahrungsschatzes kompetenter Experten des Microsoft Developer Networks (MSDN) können Entwickler das dort gesammelte Wissen für die Architektur und Implementierung ihrer Anwendungen nutzen.

Mit der Entwicklung von Unternehmenslösungen ist es wie beim Hausbau. Ohne sorgfältige Planung droht die gesamte Konstruktion einzustürzen. Mehr noch: Bei jeder Architektur ist es unerlässlich, dass der Grundriss zu jeder Zeit den sich wandelnden Anforderungen gerecht wird. So müssen schon während der Bauzeit, ja sogar nach dem Richtfest noch Änderungen vorgenommen werden können, ohne das gesamte Projekt zu gefährden. Auch bei Unternehmenslösungen steht die Planung an erster Stelle. Besonders bei neuen Technologien fehlen dem Entwickler allzu oft die nötigen Erfahrungswerte, die er sich erst mühsam aneignen muss. Das kostet den Anwender nicht nur viel Zeit und Mühe, sondern gefährdet oft auch das laufende Projekt. Um seinen Kunden diese schmerzliche und teure Einarbeitungsphase weitgehend zu ersparen, hat Microsoft für die einzelnen Problembereiche mehrere Programme entwickelt, die den Einstieg in neue Technologien erleichtern und von Anfang an für einen möglichst verlustfreien Projektablauf sorgen.

Mit „Patterns & Practices“ zur tragfähigen Unternehmenslösung

Wichtige Hilfen bei der Planung und Entwicklung einer Lösung gibt der vom Microsoft Developer Network (MSDN) in der Reihe „Patterns & Practices“ erschienene Leitfaden „Application Architecture for .NET: Designing Applications and Services“. Er beschreibt sehr detailliert und praxisorientiert, zum Teil bis hinunter in Beispielscodes, wie der Grundstein zu tragfähigen und robusten Systemen in Softwareprojekten gelegt werden kann.

Doch neben den technischen Fragen entscheidet nicht selten der richtige Einsatz von so genannten „weichen Faktoren“ über Erfolg oder Misserfolg eines Projekts. Darunter fallen zum Beispiel die Rollenverteilung, das Abbilden von Prozessen und nicht zuletzt der Umgang mit den menschlichen Aspekten im Team. Hier bietet das Microsoft Solutions Framework (MSF) einen sehr praxisnahen, aus den täglichen Erfahrungen während der Durchführung von Softwareprojekten entwickelten Leitfaden, der von Anfang an die Problemstellen eines Projekts erkennt und diese erfolgreich zu umgehen hilft.

Dennoch: Eine Anwendung kann noch so gut geplant, entwickelt und erfolgreich getestet worden sein, am Ende ist es der Benutzer, der sie einsetzen und bedienen muss. Das Microsoft Operations Framework (MOF) nimmt sich der bei Entwicklung und Betrieb auftretenden Probleme an und beschreibt Systemarchitekturen, die auch komplexeren Anforderungen standhalten können. Ferner gibt es Ratschläge, Prozesse zu entwickeln, an denen sich Hacker die Zähne ausbeißen, falls sie versuchen sollten, in ein Rechenzentrum einzudringen. Ergänzend zu MSDN, MSF und MOF enthält die Microsoft Systems Architecture (MSA) alle Infrastrukturrichtlinien zum Aufbau unterschiedlicher Datacenter –



Die Planung ist das Wichtigste für eine solide Unternehmenslösung. Microsoft und MSDN bieten Hilfe für einen reibungslosen Projektablauf.

entweder als intern genutztes Corporate, Department oder Branch Office Datacenter, als Extranet Datacenter für die Einbindung von Partnern und Lieferanten oder als Internet Datacenter für kundenorientierte E-Business-Anwendungen.

Kriterien für bewährte Lösungen

Obwohl keine Unternehmenslösung der anderen gleicht, können erfolgreiche Lösungen sehr wohl an grundsätzlichen und gemeinsamen Merkmalen erkannt werden. Zu den wichtigsten zählen:

- Die Lösung passt genau auf die Anforderungen des Geschäftsprozesses, den sie abbildet. Das klingt zwar trivial, ist jedoch in der Praxis ein häufig fehlendes Merkmal.

- Sicherheit wird von Anfang an in die Planungen integriert. Dabei geht es nicht nur um eine sichere Authentifizierung am System, sondern auch um die Gewährleistung der Datensicherheit während der Übertragung beziehungsweise der Datenentnahme.

- Gute Geschwindigkeit bei allen gängigen Funktionen und Konfigurationsszenarien.

- Hochverfügbarkeit und Robustheit sowie die Fähigkeit, sich nach Fehlern schnell in einen konsistenten Zustand bringen zu lassen.

- Die Möglichkeit der verteilten Installation in unabhängigen Rechenzentren.

- Hohe Skalierbarkeit: Eine hohe Anzahl von Funktionen und Benutzern werden mit einem minimalen Einsatz von Ressourcen bedient.

- Gutes Anwendungsmanagement: Die Anwendung lässt sich von Administratoren in jedem benötigten Szenario konfigurieren, überwachen und nach Fehlern überprüfen.

- Hohe Wartbarkeit und Flexibilität: Das System lässt sich schnell und zuverlässig an verschiedene Anwendungs- und Installationsszenarien anpassen.

So plausibel die oben stehenden Punkte erscheinen, so schwierig sind sie in der Praxis umzusetzen. Insbesondere wenn neue Technologien zum Einsatz kommen, türmen sich vor den Entwicklern Berge von Problemen. Gerade in diesen Fällen ist die umfangreiche Unterstützung der Microsoft .NET-Entwickler von großem Vorteil. Bei dem Hilfeangebot wurde viel Wert darauf gelegt, dass die bereitgestellten Informationen nicht auf theoretischen Planungen basieren, sondern von Microsoft .NET gewonnene Erkenntnisse sind. So werden durch das Microsoft Developer Network die von Experten der Microsoft Consulting Services, den Produkt-Support-Mitarbeitern und den Produktentwicklern in der Praxis gesammelten Inhalte dem Anwender zur Verfügung gestellt.

Auf diese Weise kann jeder Entwickler aus einem reichhaltigen Erfahrungsschatz kompetenter Experten schöpfen und diesen für die Architektur und Implementierung seiner Anwendung nutzen.

Schritt für Schritt zu einer komponentenorientierten, mehrschichtigen Unternehmensanwendung

Doch bevor der Anwender die Hilfe der Experten in Anspruch nehmen kann, muss er zunächst eine große Hürde überwinden: den Einstieg in die neue .NET-Technologie. Der „Patterns & Practices“-Leitfaden für die Anwendungsarchitektur hilft dabei, diese Ängste zu überwinden. Anhand einer Beispielanwendung werden dem Entwickler die grundlegenden Sachverhalte schnell und praxisnah vermittelt.

Zum Inhalt: Zunächst erklärt der Leitfaden die Unterschiede zwischen Anwendungen und den relativ neuen Diensten, oft auch Webservices genannt, da gerade diese beiden Typen von Applikationen derzeit die Hauptrolle in der Softwareentwicklung und -integration für Unternehmen spielen.

Der nächste Schritt ist die Diskussion des Designs einer mehrschichtigen, komponentenorientierten Anwendung oder eines Dienstes: Ausge-

hend von den Nutzungsfällen nehmen die darauf angepassten Komponenten ihren Platz in den verschiedenen Schichten einer solchen verteilten Applikation ein. Dies beginnt bei der Präsentationsschicht, welche vorwiegend die für den Endbenutzer sichtbaren Bestandteile und deren Hilfskomponenten als Benutzerschnittstelle enthält. Danach werden ebenso die Business- und Datenzugriffsschicht mit Geschäfts- und Datenzugriffskomponenten quasi „gefüllt“.

Jeder einzelne Gestaltungsschritt wird von konkreten Empfehlungen zu Design und Architektur der Anwendung, Grafiken und sogar Codebeispielen begleitet.

Hat der Anwender die grundlegende Struktur seiner Lösung ermittelt, sollten so früh wie möglich, drei weitere Themenbereiche integriert werden: Sicherheit, Wartung und Betrieb sowie die Festlegung der Kommunikationsmechanismen zwischen den einzelnen Schichten.

Sicherheit beginnt bei prinzipiellen Vorsichtsmaßnahmen wie beispielsweise dem Entwicklergrundsatz, niemals externen Eingaben zu vertrauen und diese entsprechend vorsichtig zu behandeln. Dazu kommen bewährte Vorgehensweisen für Authentifizierung und Autorisierung von Benutzern, sichere Verständigung im Netzwerk, das Management von Sicherheitsprofilen und natürlich die Überwachung (Auditing) der gesamten Sicherheitsinfrastruktur.

Eine ebenso wichtige Rolle spielen das Fehlermanagement, die Konfiguration und Überwachung der Anwendung zur Laufzeit sowie die Wartung des fertigen Systems. Auf diese Fragen gibt der Abschnitt über die Kommunikationstechniken die entscheidenden Antworten und Empfehlungen.

Um das Gesamtbild bei der Implementierung einer leistungsfähigen Anwendung abzurufen, widmet sich das letzte Kapitel des Architekturleitfadens den operationalen Anforderungen wie Hochverfügbarkeit, Skalierbarkeit und nicht zuletzt dem Testen.

„Patterns & Practices“ statt „Trial and Error“

Wie das Beispiel der Anwendungsarchitektur für .NET zeigt, kann mit Hilfe der Microsoft „Patterns & Practices“ auch ein relativer .NET-Neuling schnell und erfolgreich zu einer Unternehmenslösung kommen. Und Dank der fundierten und praxisnahen Erklärungen ist für den Umgang mit Microsoft .NET auch kein Doktorhut erforderlich.

Die Empfehlungen des Leitfadens verhindern außerdem eine zeitraubende und kostspielige „Trial and Error“-Phase. Dadurch ist es dem Entwickler möglich, seiner eigentlichen Aufgabe, der Entwicklung von Anwendungen, nachzugehen.

Für smarte Entwickler stellt sich „Patterns & Practices“ als Fundgrube dar mit einer Fülle von wertvollen Informationen und Codebausteinen rund um die Entwicklung unternehmenskritischer Anwendungen mit der Microsoft .NET-Technologie.

