

FS Consulting Blog

By PwC Deutschland | 14.03.2023

Das PwC IT Transformation Framework Teil 3: “IT-Architektur & Entwicklungssteuerung”

Der dritte wesentliche Baustein unseres IT Transformation Frameworks ist die IT-Architektur in Kombination mit der Entwicklungssteuerung, die wir im Rahmen dieser Blog-Serie vorstellen werden.



Content

IT-Architektur	3
Entwicklungssteuerung	5



Abbildung 1: Das IT Transformation Framework.

IT-Architektur

Transformationen im Financial Services Umfeld haben die Eigenschaft, sich in direkten Anforderungen an die IT-Systeme zu materialisieren. Daher ist ein wesentlicher Erfolgsfaktor für erfolgreiche Transformationen das klare Ziel sowie eine präzise Vorstellung der Zielarchitektur.

Basis für die Zielarchitektur-Diskussion sind immer klar formulierte (Geschäfts-)Ziele. Diese werden typischerweise in der übergreifenden Transformationsstrategie festgelegt (siehe hierzu auch Teil 1 und 2 unserer Blog Serie). Nur damit ist es möglich, die aktuelle Güte der Anwendungslandschaft zu messen und das Ziel zu bestimmen.

Als Methodik eignen sich die Schritte der strategischen Bebauungsplanung.

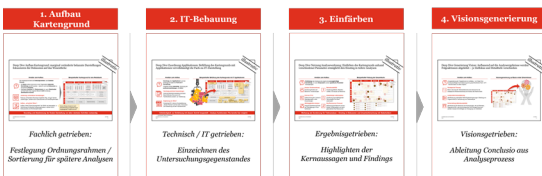
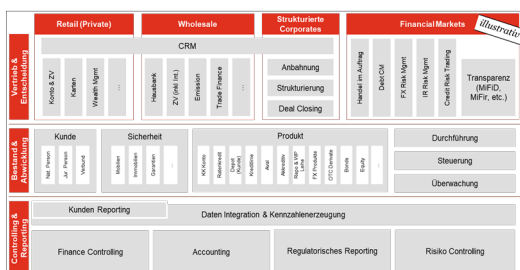


Abbildung 2: Unsere PwC-Bebauungsplanung als strukturierter Ansatz zur Zielbilderstellung.

Schritt 1: Als erstes erfolgt der Aufbau eines Kartengrunds.

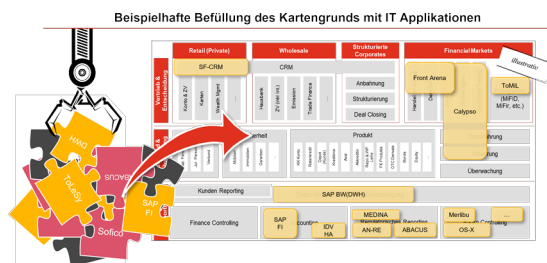
Beispielhafter Kartengrund für eine Retailbank



Dieser dient als fachlicher Ordnungsrahmen und als Einstiegspunkt für spätere Analysen. Was eignet sich als Kartengrund? Eine gute Ausgangsbasis bietet eine Business Capability Map, welche auch in den

meisten Organisationen existiert. Wichtig bei der Darstellung ist, dass sich alle Stakeholder wiederfinden und “ihren Bereich” identifizieren können. Durch die Verwendung einer bekannten Darstellung ist es möglich, sich auf die eigentliche Diskussion zu fokussieren, anstatt die Darstellung selbst zu diskutieren.

Schritt 2: Im zweiten Schritt wird auf den Kartengrund die **IT-Bebauung** aufgetragen.



Konkret wird jede Anwendung auf die jeweilige (fachliche) Fähigkeit des Kartengrunds “geklebt”, welche mittels dieser unterstützt oder ausgeübt wird. Dies kann in erster Iteration tatsächlich physisch mit Post-its erfolgen, um die relevanten Stakeholder für die momentane Bebauung zu sensibilisieren und an die IT-Systeme heranzuführen. Für die weitere Verarbeitung empfiehlt es sich natürlich, das strukturiert abzulegen. In einer ersten Form kann das PowerPoint sein; um einen effizienten und wiederholbaren Prozess zu gewährleisten, ist die Verwaltung in einem (oder vorhandenen) Enterprise Architektur Management Werkzeug sinnvoll.

Schritt 3: Der dritte Schritt dient der **Analyse und Bewertung** der Ist-Landschaft.



Der Bewertungsmaßstab sind die zu anfangs erwähnten Geschäftsziele und daraus abgeleiteten fachlichen Anforderungen an die IT. Vielmehr leiten sich daraus auch die durchzuführenden Analysen ab. Doch welche Analysen werden hier durchgeführt? Typischerweise wird geprüft an welchen Stellen der IT-Bebauung “Lücken” oder “Überlappungen” auftreten. Lücken stellen dabei fehlende IT-Unterstützung von Fachfunktionen dar. Überlappungen bedeuten hingegen, dass mehrere IT-Systeme (ergänzend oder redundant) eine Fachfunktion abbilden. Weitere typische Analysen sind die Untersuchung des Datenflusses oder auch der in Produktion aufgetretenen Incidents zentraler Anwendungen.

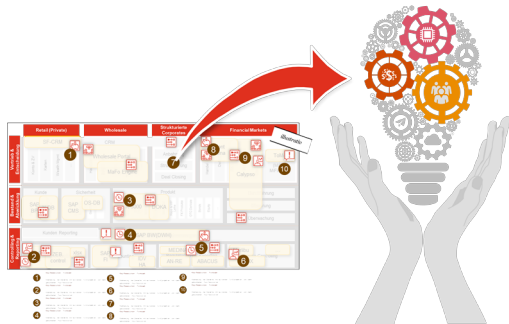
Die Ergebnisse werden dann in Form von “Highlights” in unserer Landkarte mit Applikationen aufgetragen. Diese Abstraktionsebene eignet sich gut, um mit allen relevanten Stakeholdern nun in die Diskussion der

Zielvision einzusteigen. Natürlich könnten weitere Detailanalysen gefahren werden, aus Kosten/Nutzen Perspektive ist dies an der Stelle im Prozess jedoch meist nicht notwendig.

Schritt 4: Im vierten und letzten Schritt folgt die Diskussion der Zielbebauung und der **Visionsgenerierung**.

Wichtig ist, dass alle relevanten Stakeholder und Entscheidungsträger an der Diskussion beteiligt sind und das Zielbild mittragen. Dies sichert ausreichenden Rückhalt während der Umsetzungsphase und ermöglicht überhaupt erst die Realisierung. Doch wie lässt sich eine Zielarchitektur nun entwerfen?

Visionsgenerierung auf Basis erster Erkenntnisse



Priorisierung der Bereiche ist hier das A und O. Neben den identifizierten Auffälligkeiten sollten zudem noch die - nach heutigem Kenntnisstand - zukünftigen Neuanforderungen der betroffenen Bereiche berücksichtigt werden. Legt man beides übereinander und sortiert jeweils nach Priorität, so färben sich die Bereiche, die am dringlichsten neu gestaltet werden müssen.

Ergebnis der Diskussion je Bereich muss - wie anfangs erwähnt - die für alle Stakeholder tragbare Lösung sein und nicht die aus architektonischer Sicht "theoretisch beste". Natürlich müssen alle Varianten hinsichtlich der angestrebten Ziele sachlich eingewertet werden und die Lösung muss insgesamt eine Verbesserung darstellen.

Entwicklungssteuerung

Nachdem wir mit dem Zielbild unseren eigenen Gipfel definiert haben, beginnt nun der Aufstieg dorthin. Die Entwicklungssteuerung ist mindestens ebenso wichtig und herausfordernd wie ein gutes Zielbilddesign. Im Folgenden beleuchten wir kritische Erfolgsfaktoren:

1. **"Planänderungen sind die Regel, nicht die Ausnahme"**: Justierungen am Gesamtplan und einzelnen Releases sind üblich. Über den typischerweise mehrjährigen Planungshorizont ist es schlichtweg nicht möglich alle Eventualitäten ex ante bereits korrekt abzuschätzen. Daher: Die übergreifende Releaseplanung muss mit geringem Aufwand Umplanungen vornehmen können. Nur so lässt sich die Gesamtkomplexität adäquat beherrschen.
2. **"Der Plan muss verständlich sein & Hüter des Plans muss die ganze Transformation begleiten"**: Das Zielbild sowie die Zwischentappen müssen im Programm bekannt sein und durch einen fachkundigen Dritten verstanden werden können. Das beginnt bereits bei der prominenten

Platzierung des übergeordneten Ziels. Nur so wird gewährleistet, dass über die typischerweise mehrjährige Entwicklungszeit das ursprüngliche Ziel beibehalten wird und Entscheidungen "richtig" getroffen werden.

3. **„Beherrschbarer Umfang und Komplexität für die Organisation“**: Häufig wird rein aus fachlicher Sicht ein Release geschnitten und als "Atomar" bezeichnet. Dabei wird außen vorgelesen, ob die Organisation Vorhaben solcher Größe gewohnt sind und stemmen können. Sollte letzteres nicht üblich sein, ist es sinnvoll vor Umsetzungsstart den Umfang (nach fachlichen) Gesichtspunkten zu verkleinern. Das spart „ad hoc“ Repriorisierungen in späten Umsetzungsphasen und oft auch unschöne (ewig währende) und teure Provisorien.
4. **„Der Test ist allgegenwärtig und nicht nur das Ende“**: Die Zeiten der ausschließlichen zentralen Integrationstests kurz vor Release sind lange vorbei. Kernbestandteil aller modernen (iterativen) Entwicklungsvorgehen sind Testaktivitäten. Darüber hinaus sollte bereits im ersten Testdesign ein hoher Grad an Automatisierung angestrebt werden. Das sichert eine hohe Entwicklungsgeschwindigkeit in allen Phasen und vermeidet böses Erwachen beim Einstieg in zentrale Testphasen. Das Thema Testautomatisierung wird ausführlicher im nächsten Artikel unserer Serie am 11.04.2023 beschrieben.
5. **„Weniger ist mehr: Simple Reportingdesign & automatisierte Datenerhebung“**: Beim Aufsatz des Projekt- & Programmreportings sollten möglichst einfach zu messende Kennzahlen erhoben werden. Beispielsweise kann der Vergleich von inhaltlichem zu monetären Fortschritt auf einem Liefergegenstand in Kombination mit qualitativen Aussagen der (Teil-)Projektleiter bereits ausreichend Transparenz für eine effiziente Steuerung bieten. Auch hier hilft ein hoher Automatisierungsgrad zur Konzentration der Kapazität auf die Steuerung selbst zu lenken.
6. **„Keine langen Konzeptionsphasen – MVP-Gedanke spart Geld“**: Der iterative Gedanke sollte bereits während der Konzeptionsphase verfolgt werden. Ein gutes Grobkonzept, das die wesentlichen Leitplanken legt und Raum für die prinzipiell notwendigen Funktionen oder Erweiterungen bietet, stellt eine ausreichende Basis für erste Umsetzungen dar. Hier gilt: strebe im ersten Schritt nicht die „rundumsorglos-Lösung“ an, sondern gehe iterativ je Einzel-Usecase vor mit: Konzeptionierung, Implementierung, Testen und Verfeinerung.

Gerne unterstützen wir Sie mit unseren umfangreichen, interdisziplinären Erfahrungen und Praxisbeispielen und stehen bei weiteren Fragen gerne zur Verfügung.

Am 18.04.2023 folgt unser nächster Artikel zum Thema Testmanagement.

[Zu weiteren PwC Blogs](#)

Keywords

[Digitalisierung](#), [Framework](#), [IT-Systeme](#)

Contact



Frank Wagershauser

Frankfurt am Main

frank.wagershauser@pwc.com