

# Applikations-Rationalisierung

Und jetzt sind Sie dran: Sie und Ihr Team sind die Enterprise-Architekten der kleinen Beispielgeschichte.

Ermitteln Sie, welche Applikationen Sie stilllegen wollen.

1. Stellen Sie einen einfachen Bebauungsplan auf. Wählen Sie dafür aus den in der Vorlesung vorgestellten Softwarekarten die passende aus.
2. Entwerfen Sie jeweils eine einfache Balanced Scorecard (höchstens je 3-4 Kriterien) für Strategiebeitrag und Wertbeitrag. **Sie können das vorgegebene Template nutzen!**
3. Tragen Sie die Daten mit TCO und FI/FO in ein Ward-Peppard-Diagramm ein.
4. Werten Sie das Diagramm aus und empfehlen Sie Kandidaten für eine Transformation. Priorisieren Sie, und geben Sie ggfs. Hinweise auf noch notwendige Architektur-Prüfungen.

## Hinweise:

- Sie brauchen nur Kandidaten für eine Transformation zu identifizieren!
  - Diese werden dann im nächsten Schritt (nicht Ihre Aufgabe) näher untersucht
- Die strategischen Ziele wurden in der Geschichte genannt: Multi-Channel und Self-Service. Wenn Sie das technisch umsetzen wollen, dann brauchen Sie offene Schnittstellen.
- Die strategische SOLL-Technologieplattform der Bank ist in aller Kürze:
  - JEE auf Linux Blades, Oracle, Webclients
  - Mainframe ist Legacy, alles andere ist Nischentechnologie (kann ok sein)

## 1.1 Betrachtete Geschäftsprozesse

**Hinweis:** Sie brauchen für diese Übung die Kritikalität der Geschäftsprozesse nicht zu gewichten. Jeder Prozess ist hier also gleich wichtig für die Bank.

Prozess	Kürzel	Beschreibung
Basel II Reporting	B2R	Alle Aspekte des Basel II Reportings für die Bank, d.h. gibt eine Aussage über Compliance mit dieser regulativen Vorgabe.
Gesamtbilanzierung	GBZ	Stellt die laufende Gesamtbilanz der Bank auf
Bilanzierung der Beteiligungen	BB	Gibt eine Übersicht über alle Unternehmens-, Wertpapier- und sonstige Beteiligungen der Bank. Erlaubt auch Historisierung und Trendanalyse.
Kreditrisiko-Berechnung	KRB	Berechnung und Reporting des Kreditrisikos, und damit Instrument für die Entscheidung zur Kreditvergabe und zur Höhe des Zinssatzes

## 1.2 Applikationen

Applikation	Kürzel	Kurzbeschreibung	Alter (Jahre)	Total Cost of Ownership (TCO)	Bedient Prozesse	Wird verwendet in Region	Fan-In / Fan-Out
Universelle Reporting-Plattform	URP	Selbstentwickelte, Mainframe-basierte Reporting-Plattform, geschrieben in COBOL. Klassisches Beispiel einer kaum dokumentierten Stove-Pipe-Architektur. Vor einigen Jahren mit sehr hoher Investition um Basel II-Reporting Fähigkeiten erweitert. Demnächst steht Erweiterung auf Basel III an. Eine Einbindung in Self-Service- und Multichannel-Angebote wird schwierig, da die Service-Schnittstellen fehlen und erst aufwändig an die veraltete Technologie und die schlechte Architektur angebunden werden müssen. Läuft stabil, bietet aber wenig Funktionalität für die Fachseite. Nutzer klagen.	15	940	B2R, GBZ	West, Ost, Süd	15
JBasel2	JB2	Kommerziell eingekaufte, JEE-basierte Basel 2 Reporting Engine mit gut konfigurierbaren Schnittstellen und APIs, also auch brauchbar für Einbindung in eine Web-Informationenplattform der Bank. Die Herstellerfirma ist bekannt für gute und schnelle Reaktionen auf neue Anforderungen. Hohe Anwenderzufriedenheit. Upgrade auf Basel III ist angekündigt.	4	150	B2R	Nord	8
BilanzClient	BC	Selbstentwickeltes Reporting-Modul auf Basis von MicroStrategy (einem kommerziellen BI-Tool). Hinzufügen neuer Reports geht sehr schnell. Gut dokumentiert, da noch relativ neu. Läuft auf der Standard-JEE-Linux-Plattform. Wird von den Nutzern als bedienungsfreundlich allgemein geschätzt, deckt aber nicht alle nötigen Informationen ab, so dass die Anwender zusätzlich manuelle Schritte tun müssen. Aufgrund der kommerziellen Plattform gute Webanbindung möglich, um Reports für Bankkunden und Außendienstmitarbeiter bereitzustellen.	3	80	GBZ	Nord	13

Applikation	Kürzel	Kurzbeschreibung	Alter (Jahre)	Total Cost of Ownership (TCO)	Bedient Prozesse	Wird verwendet in Region	Fan-In / Fan-Out
Berichtsmodul für Beteiligungen	BMB	Selbstentwickelte, Windows Client / Windows Server / MS SQL Server basierte Berichtssoftware zur Übersicht über Beteiligungen. Schlecht dokumentiert. Miserable Wartbarkeit, da alle Geschäftsregeln hartcodiert sind. Basis-Schnittstellen existieren, neue wären wohl nur aufwändig zu implementieren. Hohe Unzufriedenheit der Anwender über Usability. Der Funktionsumfang ist mittlerweile OK, aber neue Reports und Regeln sind nur sehr mühsam hinzuzufügen. Die Fachseite braucht immer die IT dazu.	7	240	BB	Nord, West, Süd	12
Ruby-Bet	RBET	Ruby-on-Rails-basierte Reporting Engine für Beteiligungen, hochkonfigurierbar über eine intuitive Regelengine, in der die Fachanwender selbst Regeln einfügen und ändern können. Begeisterte Nutzer und ein hochkompetentes, engagiertes Entwicklerteam. Architektur ist der Traum aller EAM-ler: gut dokumentiert, hoch integrierbar. Allerdings noch nicht völlig ausgereift, und Regelsatz noch sehr lückenhaft.	1	75	BB	Ost	3
Kredit-Engine 1	KR1	Mainframe-basierte Kredit-Engine, Eigenentwicklung, auf die Eigenheiten der Bank zugeschnitten. Hohe Zufriedenheit einer kleinen, eingeschworenen Nutzer- und Entwicklergemeinschaft. Allerdings besteht das Risiko, dass Nutzer und Entwickler miteinander altern und in Rente gehen. Eine vernünftige Systemdokumentation liegt nicht vor. Auch die Anfragen nach neuen Schnittstellen werden meist „ausgesessen“.	14	320	KRB	West, Süd	5
Kredit-Engine 2	KR2	Mainframe-basierte Kredit-Engine, kommerzielles Produkt, Portierung nach Java/Linux für 2016 angekündigt. Dann sollen auch sehr viel mehr Schnittstellen für Batch-orientierte Massenanfragen und Reporting zur Verfügung stehen. Mittlere Nutzerbasis. Mittlere Nutzerzufriedenheit, deckt nicht alle Spezialfälle der Bank ab, einige manuelle Schritte bleiben. Allerdings wirkt die Herstellerfirma bemüht, die Mängel abzustellen.	8	240	KRB	Ost, Nord	12