

Agile Migration

Michael Pisula

TNG Technology Consulting GmbH
Beta-Str. 13A
85774 Unterföhring
mihcael.pisula@tngtech.com

Abstract: Die Migration komplexer Systeme ist oft teuer, langwierig und birgt große Risiken. Das liegt häufig daran, dass ein System nicht einzeln migriert werden kann, weil starke Abhängigkeiten bestehen. Nicht selten führt dies dazu, dass die Migration als nicht durchführbar angesehen wird. Wir haben für unsere Kunden ein Verfahren entwickelt, um die Migration zu entkoppeln. Statt die Migration in einem Big Bang durchzuführen, werden die Systeme zuerst entkoppelt, um dann Schritt für Schritt migriert zu werden. Somit kann man die Risiken erheblich senken und die Komplexität verringern. Durch die Entkopplung der Migration wird es möglich, agil auf Verzögerungen im Projektplan und Änderungen der Anforderungen einzugehen.

1 Einleitung

Die Situation dürfte den meisten bekannt vorkommen. In der Systemlandschaft schlummert eine Zeitbombe, ein System, vor langer Zeit entstanden, kaum noch wartbar, fehleranfällig. Neue Features in das System zu bringen ist so gut wie unmöglich. Ganz klar, das System muss migriert werden. Doch damit fangen die Probleme oft erst an. In diesem Beitrag möchte ich die Probleme mit Migrationen von komplexen Systeme aufzeigen, und eine Lösung vorstellen, mit der man einige dieser Probleme umgehen kann.

2 Die Migration

Die Migration eines Systems ist immer dann problematisch, wenn dieses System in einer Systemlandschaft eingebettet ist. Denn so ein System kommuniziert mit anderen Systemen. Somit muss man sich darüber Gedanken machen wie diese Kommunikation nach der Migration aussehen soll. Alle Schnittstellen des alten Systems 1:1 zu übernehmen, ist normalerweise keine ideale Option. Zum einen soll das neue System mehr können, zum anderen sind die alten Schnittstellen auch oft nicht mehr im besten Zustand, die eingesetzte Technologie ist veraltet, es wurden Workarounds eingebaut, die man loswerden will etc. Bei einigen der anderen Systeme kann man die Schnittstellen anpassen, gerade dann wenn diese Systeme noch jung sind, und es einen aktiven Support

gibt. Das trifft aber nicht auf alle zu, meist sind einige der anderen Systeme auch schon veraltet, und eine Ablösung ist schon geplant oder zumindest angedacht.

Migriert man jedes dieser Systeme einzeln, hat man das Problem, dass die noch nicht migrierten Systeme und ihre Schnittstellen weiter unterstützt werden müssen. Genauso muss aber auch schon die Unterstützung der neuen Systeme vorhergesehen und eingebaut werden. Dies macht die Entwicklung der neuen Systeme deutlich teurer und langwieriger. Oftmals möchte man ein altes System auch in mehrere neue Systeme unterteilen, damit wird es deutlich komplexer die gleichen Schnittstellen zu unterstützen, die das alte System angeboten hat. Oder es gibt Workflows die in den neuen Systemen nicht umgesetzt werden, die aber trotzdem für die Übergangszeit eingebaut werden müssen. In anderen Fällen ist die neue Lösung ein Standardprodukt, welches die alten Systeme nicht unterstützt und erst kostspielig angepasst werden müsste.

Alles in allem ist eine Schritt-für-Schritt Migration nur selten sinnvoll, wenn zum Beispiel nur sehr wenige Systeme betroffen sind oder keine Änderungen an den Schnittstellen gewünscht sind.

3 Der Big Bang

Eine häufig verwendete Lösung der Big Bang Ansatz. Bei der Big Bang Migration werden alle Systeme zum gleichen Zeitpunkt migriert. Dieser Ansatz beseitigt die eben beschriebenen Probleme, es müssen in allen neuen Systemen nur die Schnittstellen der Systeme unterstützt werden, die nach der Migration übrig bleiben. Der Preis dafür ist aber eine hohe Komplexität bei der Planung und Umsetzung, sowie große Risiken, gerade in Bezug auf Verzögerungen im Projektverlauf. Da die Systeme miteinander kommunizieren müssen, ist eine sehr detaillierte Abstimmung zwischen den einzelnen Projekten unabkömmlich. Verzögerungen bei nur einem System führen zu einer Verzögerung der kompletten Migration. Auch die Umschaltung der Systeme selbst ist sehr riskant. Zum einen wegen der hohen Komplexität, da viele Systeme gleichzeitig umgeschaltet werden, zum anderen weil die einzige Möglichkeit auf schwere Probleme bei der Migration zu reagieren, ein komplettes Rollback aller Systeme ist. Die Risiken sind schwer zu kalkulieren und je mehr Systeme betroffen sind, desto aufwändiger wird die Migration.

Oft genug führen diese Gründe dazu, dass die Migration immer wieder verschoben wird, weil niemand das Risiko auf sich nehmen möchte. Das führt oft dazu, dass Systeme weit über ihr Haltbarkeitsdatum erweitert und gewartet werden, bis unweigerlich der Zeitpunkt kommt, an dem man um die Migration nicht mehr umhin kommt.

4 Agile Migration

4.1 Ansatz

Für unsere Kunden in solchen Situationen haben wir einen Ansatz entwickelt, der die meisten der Probleme löst und komplexe Migrationen deutlich erleichtert.

Während die Big Bang Migration dadurch kompliziert wird, dass sie die Migration aller Systeme synchronisiert, hat die Migration einzelner Systeme das Problem, dass jedes neue System deutlich komplexer wird, da es sowohl Schnittstellen zu den alten als auch zu den neuen Systemen unterstützen muss. Unser Verfahren kombiniert die Vorteile: die neuen Systeme müssen nur die Schnittstellen unterstützen die nach der kompletten Migration verwendet werden sollen, die Migration der einzelnen System ist aber trotzdem entkoppelt, jedes System kann unabhängig von den anderen migriert werden.

Bei der Big Bang Migration wird den einzelnen Migrationsprojekten ein starres Korsett aufgezwungen, alle müssen zum gleichen Zeitpunkt testbar sein und für die Migration fertig sein. Darin ähnelt die Big Bang Migration dem Wasserfallmodell, dass ebenfalls eine starre Abfolge von Phasen hat. Wie beim Wasserfall muss bei Problemen in eine frühere Phase zurückgefahren werden. Treten beim Testen der Systeme Probleme auf, wirkt sich das auf alle Systeme aus. Gibt es bei der Umschaltung eines Systems Probleme, müssen alle Systeme wieder in die Testphase zurück. Wie beim Wasserfallmodell auch ist die Flexibilität im Umgang mit Änderungen und Problemen auch bei der Big Bang Migration sehr eingeschränkt. Der hier vorgestellte Ansatz ist im Gegensatz dazu agil. Die starren Phasen werden aufgelöst, während ein System entwickelt wird, kann ein anderes schon getestet und ein drittes bereits in Produktion gebracht werden. Treten bei der Migration eines Systems Probleme auf, wirkt sich dies nicht auf die Projektplanung der anderen Systeme aus. Deshalb haben wir unseren Ansatz Agile Migration getauft.

Die Kernidee bei dem Verfahren ist es, die einzelnen Systeme voneinander zu entkoppeln. Sind die Systeme entkoppelt, können sie in einer beliebigen Reihenfolge migriert werden. Um die Entkopplung zu erreichen wird ein sogenannter Adapter zwischen die Systeme eingebaut. Dieser Adapter wächst mit der Migration und verbindet die bereits migrierten Systeme und die Altsysteme die sich noch in Produktion befinden. Er fungiert sozusagen als Übersetzer zwischen neuen und alten Systemen, die Daten kommen immer noch von den Systemen selbst, der Adapter enthält die Logik die notwendig ist, um für jede Anfrage die notwendigen Daten zu holen und ins richtige Format zu bringen. Wenn nötig kann der Adapter auch Daten verwalten, die von Altsystemen noch gebraucht werden, nach der kompletten Migration aber wegfallen. Indem man die Aufgabe zwischen neuen und alten Systemen zu vermitteln in eine externe Komponente verlagert, muss beim Entwurf der neuen Systeme nur der Endzustand betrachtet werden. Sobald die Migration abgeschlossen ist, kann der Adapter abgeschaltet werden. Es verbleibt somit kein Code in Produktion, der nur dazu dient, mit abgelösten Systemen zu kommunizieren. Durch den Adapter wird auch der Druck genommen die Migration in einem engen Zeitrahmen umzusetzen: jedes System kann

migriert werden, sobald es fertig implementiert wurde. Dadurch werden die Projektpläne der einzelnen Systeme voneinander entkoppelt und Verzögerungen wirken sich somit nicht mehr projektübergreifend aus. Es ist ebenfalls möglich neue und alte Systeme parallel laufen zu lassen, etwa weil die Daten nicht auf einen Schlag, sondern nach und nach migriert werden. Insofern kann die Agile Migration auch bei der Datenmigration den Big Bang Ansatz unnötig machen.

4.2 Umsetzung

Bei der Agilen Migration werden zuerst die Systeme identifiziert, die migriert werden sollen. Im Normalfall migriert man zuerst das größte System, welches oft das zentrale System mit den meisten Schnittstellen ist. Im weiteren Verlauf nennen wir dieses System Hauptsystem. Dieses System ist oft auch der eigentliche Fokus der Migration, es hat die meisten Abhängigkeiten und ist am schwierigsten zu migrieren. Die Systemlandschaft vor und nach der Migration wird festgelegt. Die Umsetzung des neuen Hauptsystems und des Adapters kann zeitgleich starten. Als ersten Migrationsschritt kann der Adapter in das intiale System integriert werden und dort bereits die zu migrierenden Systeme entkoppeln. Gerade wenn die Dokumentationslage der abzulösenden Systeme dürftig ist, gibt dieser Schritt die Möglichkeit die Daten zu analysieren die zwischen den Systemen fließen. Zu diesem Zweck können Methoden aus der Big Data Analyse im Adapter verwendet werden um die Kommunikation zwischen den Systemen zu kategorisieren. Damit kann sichergestellt werden, dass wichtige Use Cases nicht übersehen werden und dass die Anfragen, die der Adapter an die alten Systeme schickt auch von diesen verstanden werden. Diese Informationen sind mitunter für das neue Hauptsystem elementar. Oft ist die Dokumentation von alten Systemen in einem schlechten Zustand, so dass es keine gesicherten Erkenntnisse gibt, welche Daten benötigt werden oder welche Abhängigkeiten bestehen. In diesem Modus beobachtet der Adapter nur, und leitet die Daten 1:1 weiter.

Im nächsten Schritt erfolgt dann die Migration des Hauptsystems. Das neue Hauptsystem kommuniziert nun direkt mit den von der Migration nicht betroffenen Systemen und über den Adapter mit den noch zu migrierenden Systemen. Auch in diesem Schritt können die Daten, die durch den Adapter fließen, analysiert werden, um frühzeitig Probleme zu finden. Zu diesem Zeitpunkt sind keine weiteren Arbeiten am Adapter notwendig, wenn nicht weitere Systeme in den Fokus der Migration rücken.

Als nächstes folgen nun Schritt für Schritt die weiteren Systeme. Migrierte Systeme werden direkt an das neue Hauptsystem angeschlossen. Nach der Migration des letzten Systems kann der Adapter abgeschaltet werden. Wenn die Migration eines Systems erst viel später erfolgen kann, kann der Adapter auch über längere Zeit in Produktion eingesetzt werden.

4.3 Umgang mit Verzögerungen

Der Big Bang Ansatz ist besonders empfindlich wenn bei der Implementierung eines der neuen Systeme Verzögerungen entstehen. Da die Migration viele Systeme umfasst, ist

eine Verzögerung teuer. Die alten Systeme müssen laufen, bis alle neuen Systeme fertig umgesetzt sind, eventuell vorhandene neue Features können nicht genutzt werden. Oft genug endet dies in vielen Eskalationsmeetings. Bei der Agile Migration ist die Lage etwas entspannter. Da jedes System einzeln migriert wird, wirken sich Verzögerungen nur auf dieses System aus. Sobald ein System fertig entwickelt wurde kann es produktiv gebracht werden. Auch hier können natürlich Probleme entstehen, gerade wenn zwischen den neuen Systemen Abhängigkeiten bestehen. Angenommen zwei Systeme A und B sollen noch migriert werden, dabei greift B auf Schnittstellen von A zu. Deshalb war vorgesehen, zuerst A, dann B zu migrieren. Durch eine Verzögerung bei der Implementierung wird aber System A nicht vor System B fertig. Statt mit der Migration von B zu warten, kann man die Schnittstellen von A im Adapter bereitstellen, und so bereits früher B in Produktion bringen. Dieser flexible Ansatz ist besonders dann sinnvoll, wenn die Verzögerungen groß oder die Schnittstellen einfach zu implementieren sind. Auch im Falle von Abhängigkeiten die spät entdeckt werden, kann der Adapter schnelle Abhilfe schaffen, bis eine Lösung im eigentlichen System bereitgestellt werden kann. Da der Adapter ein Wegwerf-Projekt ist, können hier auch Migrations-Workarounds, die man im fertigen System nicht haben möchte, untergebracht werden.

4.4 Der Adapter

Offensichtlich ist das beschriebene Verfahren sehr abhängig von der Qualität des Adapters. Viele der Unsicherheiten des Projekts lasten auf dieser Komponente, da er die Kommunikation mit den Legacy-Systemen leisten muss, für die es unter Umständen keine Dokumentation mehr gibt. Da alle Kommunikation über den Adapter läuft, muss er auch sehr performant sein, und darf die anderen Systeme nicht ausbremsen. Da er eine zentrale Rolle innerhalb der Infrastruktur einnimmt, ist es wichtig, dass diese Komponente besonders ausfallsicher ist. Ein weiterer Punkt den man bedenken sollte, ist dass der Adapter viele verschiedene Schnittstellen unterstützen können muss, selten sind Legacy-Systeme homogen in der Verwendung von Schnittstellen-Technologien. Der Adapter sollte auch sehr leicht zu erweitern und zu warten sein. Durch die Natur des Anwendungsfalls werden während der Migration sichere Annahmen über Workflows über den Haufen geworfen, und der Adapter muss schnell in der Lage sein, sich auf neue Gegebenheiten anzupassen. Die Agile Migration steht und fällt mit einer professionellen Umsetzung des Adapters.

5 Zusammenfassung

Wenn es um die Migration von komplexen Systemen geht, wird häufig der Ansatz der Big-Bang Migration eingesetzt. Dieser Ansatz bringt aber viele Probleme mit sich, und ist sehr unflexibel. In vielen Situationen kann hier die Agile Migration helfen, die viele Probleme des Big Bang Ansatzes löst und sehr flexibel mit Risiken und Problemen umgeht. Bei der Agilen Migration wird die Migration dadurch vereinfacht, dass Systeme mittels einer Adapter-Komponente voneinander entkoppelt werden, was eine flexible Migration erlaubt. Der Adapter verbindet bereits migrierte Systeme mit noch nicht

migrierten Systemen und fungiert als Schnittstellen-Übersetzer. Nach der fertigen Migration wird der Adapter abgeschaltet, so dass keinerlei Spuren der alten Schnittstellen in der Systemlandschaft verbleiben.