

Software Ecosystems

Tobias Schrade
tobias@familie-schrade.de

Abstract: Viele große Unternehmen bedienen sich bereits selbst aufgebauter Software Ökosysteme. Doch was genau ist ein Software Ökosystem und wie entsteht es? Welchen Nutzen kann man daraus ziehen und was gilt es zu beachten? Diese Fragen werden in der folgenden Ausarbeitung behandelt.

1 Überblick Ökosysteme

Ein Ökosystem ist eine Einheit aller biotischer (Pflanzen, Tiere und Mikroorganismen) und abiotischer (nicht-lebenden) Faktoren und deren Wechselwirkungen. Da diese Definition für ein Software Ökosystem noch nicht ganz passend ist, kann man sie weiter unterteilen. Als erster Schritt in natürliche und menschliche Ökosysteme wobei ein Software Ökosystem in letztere Kategorie fällt. Ein menschliches Ökosystem ist dann schon eine Einheit aus Akteuren, deren Beziehungen und den Transaktionen entlang dieser Beziehungen. Von einem menschlichen Ökosystem ausgehend kann man noch eine weitere Unterteilung in wirtschaftliche und soziale Ökosysteme vornehmen. Bei einem wirtschaftlichen Ökosystem sind die Akteure nun Firmen, Zulieferer und Kunden und die Transaktionen sind Geld-, Güter- und Informationsflüsse. Dies ist nun auch die passende Oberkategorie für ein Software Ökosystem. Die hierarchische Struktur der Ökosysteme wird auch nochmal in Abb. 1 verdeutlicht. Im Folgenden werden dann noch neben den Gründen für Software Ökosysteme deren verschiedenen Bereiche mit den Elementen Merkmale, Erfolgsfaktoren und Herausforderungen erläutert. Abschließend wird noch auf den Umstieg zu einem Software Ökosystem, die Auswirkungen eines solchen auf die Software-Programmierung und die Frage Was hat ein Software Ökosystem mit IT-Transformation zu tun? eingegangen.

2 Software Ökosysteme

„Ein Software Ökosystem besteht aus einer Anzahl an Softwarelösungen welche die Aktivitäten und Transaktionen der Akteure des zugehörigen sozialen oder wirtschaftlichen Ökosystems und den Organisationen, welche die Software bereitstellen, ermöglicht, unterstützt und automatisiert.“ (Nach: Bosch, Jan; 2009) Um diese Definition zu verdeutlichen dient Abbildung 2. Hierbei ist zu erwähnen, dass die Anzahl der Akteure nicht festgelegt und nur beispielhaft zu verstehen ist.

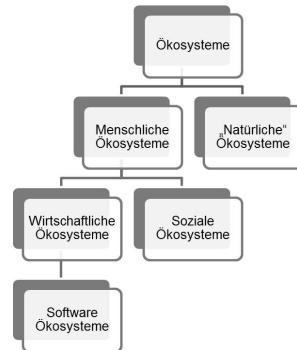


Abbildung 1: Überblick Ökosysteme [BO09]

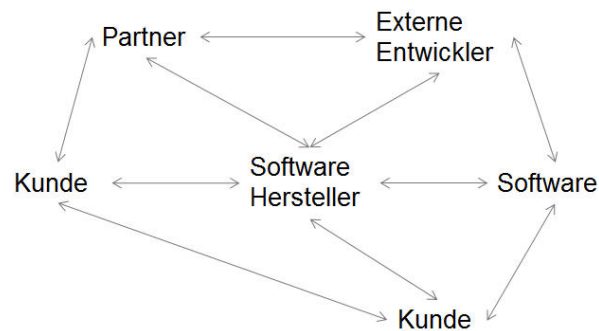


Abbildung 2: Software Ökosystem (Eigene Abb.)

2.1 Was führt zu einem Software Ökosystem?

Einzelne Software Pakete werden immer größer und werden von immer mehr Kunden genutzt. Mehr Kunden haben nun auch mehr individuelle Ansprüche an die Software. Um diese Ansprüche alle zu befriedigen ist nun ein sehr hoher Research and Development Aufwand nötig. Das übersteigt oft die Möglichkeiten einer einzelnen Firma und sie muss sich neue Partner suchen. Hier ist nun ein gängiger Ansatz um die Software ein Ökosystem aufzubauen, wodurch die Kundenwünsche von mehreren Firmen bedient werden können und jede einzelne Firma trotzdem noch Gewinne erzielt. Ein weiterer Punkt ist, dass im Software-Bereich meistens "The winner takes it all" gilt. Dies bedeutet, dass sich nur die beste und am weitesten verbreitete Software durchsetzt. Dadurch sind Firmen gezwungen ihre Produkte in einem rasanten Tempo zu entwickeln, zu verbessern und zu verbreiten. Auch hierfür ist ein Software Ökosystem eine adäquate Lösung.

2.2 Verschiedene Bereiche der Software Ökosysteme

Es gibt verschiedene Arten von Software Ökosystemen, die sich nach folgender Tabelle in verschiedene Bereiche einteilen lassen:

Bereich \ Kategorie	Desktop	Web	Mobil
Betriebssystem	Windows, Apple OS X, Linux	Google AppEngine, Yahoo! developer	Android, Palm, iPhone
Anwendung	Microsoft Dynamics NAV, SAP, MS Office	eBay, Amazon	AppStore, Android Market
Benutzer-Programmierung	MS Excel, Lego Mindstorms	Yahoo! Pipes	-

Abbildung 3: Verschiedene Bereiche der Software Ökosysteme [BO09]

Zum einen erfolgt die Einteilung in die Bereiche Desktop, Web und Mobil, zum anderen in die Kategorien Betriebssystem (OS), Anwendung und Benutzer-Programmierung. Jeder Bereich hat seine individuellen Merkmale, Erfolgsfaktoren und Herausforderungen, welche in den folgenden Abschnitten näher erläutert werden.

2.2.1 Betriebssysteme

Merkmale:

Betriebssysteme benötigen Programme, welche für das Betriebssystem entwickelt wurden. Ohne diese Programme wird auch niemand das Betriebssystem verwenden. Daher werden für Betriebssysteme vom Software-Hersteller Entwicklungsumgebungen bereitgestellt. Diese sind ein sehr wichtiges Merkmal. Wenn ein Betriebssystem keine gute Entwicklungsumgebung besitzt werden sich kaum Programmierer die Mühe machen und Anwendungen für dieses Betriebssystem programmieren, was die Attraktivität des OS für Kunden erheblich senkt. Ein weiteres Merkmal ist es, dass der Erfolg des OS sehr eng mit dem Erfolg der für das OS entwickelten Geräte verknüpft ist. Wenn sich das Gerät schlecht verkauft, verkauft sich auch automatisch das darauf installierte OS schlecht.

Erfolgsfaktoren:

Die Erfolgsfaktoren hängen alle sehr eng miteinander zusammen und lassen sich aus den Merkmalen abbilden (siehe Abb. 4).

Durch die Grafik wird verdeutlicht, dass ein Erfolgsfaktor eine Voraussetzung für den nächsten ist. Dadurch entsteht eine Art Teufelskreis, den man in Gang setzen muss, was aber nicht einfach ist.

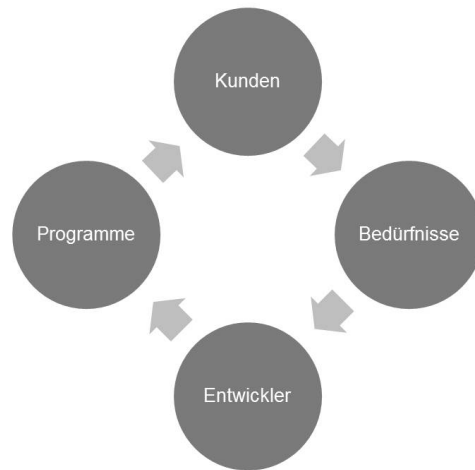


Abbildung 4: Erfolgsfaktoren für ein Software Ökosystem im Bereich Betriebssysteme [BO09]

Herausforderungen:

Eine wichtige und nicht einfach zu lösende Herausforderung ist die Spanne zwischen Abwärtskompatibilität und das nutzen neuester Errungenschaften und technischen Möglichkeiten. Wenn man nämlich immer die neuesten Features einbaut kann es schnell passieren, dass man ältere Geräte überfordert. Andererseits will niemand ein OS mit der Technik von gestern. Hier muss man also das richtige Maß finden und beide Bereiche ausreichend bedienen. Eine weitere Herausforderung die hauptsächlich den Bereich Mobil betrifft, ist der enorme Erfolg des OS Microsoft Windows im Bereich Desktop. Jetzt will nämlich jeder einen Teil des Erfolges im Bereich Mobil abbekommen, was den Konkurrenzdruck erhöht. (Nokia sah sich z.B. schon gezwungen sein OS Symbian komplett zu öffnen).

2.2.2 Anwendungen

Merkmale:

Bei Anwendungen ist es meist der Fall, dass erst eine erfolgreiche Anwendung existiert, welche dann in ein Software Ökosystem eingebettet wird. Dadurch sind schon wichtige Voraussetzungen, wie z.B. ein großer Kundenstamm erfüllt. Ein zweites wichtiges Merkmal ist es, dass die Dritten möglichst unbemerkt am Programm mitwirken, dass beim Kunden immer noch das Gefühl besteht, dass alles aus einer Hand kommt.

Erfolgsfaktoren:

Der Erfolg von Software Ökosystemen im Bereich Anwendungen hängt zum Einen von guten Entwicklungsumgebungen für Applikationen für die Anwendung ab. Denn wie auch im Bereich OS werden die neuen Entwickler lieber in einer guten Entwicklungsumgebung Applikationen schreiben, als wenn sie sich erst in eine komplizierte Umgebung einarbeiten.

ten müssen. Zum Anderen ist auch die nahtlose Integration der neuen Funktionen erfolgsentscheidend. Wenn das Programm am Schluss dem Kunden total zusammengestückelt vorkommt, wird er es nicht so gern verwenden und auch zum Teil frustriert sein, wenn er Dinge wie z.B. die Hilfe nicht dort findet wo er sie erhofft.

Herausforderungen:

Die Einbettung der Anwendung in ein Software Ökosystem stellt gleich die erste Herausforderung an das Unternehmen da. Teile des Codes müssen zugänglich gemacht werden und somit früher gut gehütete Geheimnisse nun offen dargelegt werden. Darüber hinaus verliert man auch an Freiheiten, was die Weiterentwicklung des Programmes darstellt. Man kann nämlich nicht einfach Code abändern, da dadurch die Entwicklung von Applikationen beeinflusst werden kann und auch eventuell Projekte von Dritten nicht mehr funktionieren. Eine weitere Herausforderung ist es ein neues Geschäftsmodell zu entwickeln, da das alte ja meistens nur auf die eigene Firma ausgelegt war und nun aber viele weitere Firmen mit eingebunden werden müssen. Hierzu zählen auch neue Geschäftsprozesse. Nun folgt der Abschnitt über die Benutzer-Programmierung.

2.2.3 Benutzer-Programmierung

Merkmale:

Bei der Benutzer-Programmierung handelt es sich um eine sehr einfache Weise der Programmierung, mit der sich Endbenutzer eigene Programmteile mit Hilfe einer eigenen Programmiersprache entwickeln können (z.B. Makros in MS Excel). Ein wichtiges Merkmal ist, dass diese Programmiersprache sehr einfach gehalten werden muss, da auch "Nicht-Programmierer" damit umgehen können müssen. Des Weiteren wird mit dieser Programmiersprache meist nichts fundamental Neues erschaffen sondern nur kleine Programmteile, die sehr spezifisch einzelne Kunden unterstützen.

Erfolgsfaktoren:

Hier hängt der Erfolg vor Allem vom Verhältnis zwischen Nutzen und Aufwand der neuen Features ab. Niemand wird sich die Mühe machen und ein Makro erstellen, das er von Hand schneller eingeben könnte.

Herausforderungen:

Die wohl wichtigste Herausforderung ist die Einfachheit der Programmiersprache und der anschließenden Nutzung der neuen Features. Denn wie schon oben genannt muss der Nutzen für jeden einzelnen Kunden schnell ersichtlich sein und er muss die Programmteile einfach erstellen können, sonst wird er sich nicht der Benutzer-Programmierung bedienen, sondern die Funktionen einfach jedes mal händisch wieder eingeben, wenn er sie braucht. Eine weitere Herausforderung ist es auch, dass für die Entwicklung der eigenen Programmiersprache meist nur ein sehr geringes Etat zur Verfügung steht und dadurch nicht auf Einzelheiten geachtet werden kann, was die Sprache im Normalfall wieder verkompliziert.

3 Umstieg zu einem Software Ökosystem

Die Art des Umstieges ist sehr stark abhängig vom Bereich der Software. Vor allem in den Bereichen OS und Benutzer-Programmierung ist der Umstieg aus verschiedenen Gründen sehr schwierig. Bei den OS muss man, wie oben genannt, den Kreis der Erfolgsfaktoren „ins laufen bringen“ und bei der Benutzer-Programmierung ist meist das geringe Etat schon Grund um einen Umstieg zu verhindern. Also bleiben als letztes nur noch die Anwendungen übrig. Hier spielt sich dann meist ein Prozess ähnlich Abb. 5 ab. Die einzelnen Stufen sind dann aber nicht so ersichtlich sondern die Grenzen überschneiden sich. Ein Beispiel hierfür wäre: Software = MS DOS, Software Paket = MS Windows XP und Software Ökosystem = MS Windows 7. MS DOS beschränkt sich fast nur auf Basisfunktionen. Bei MS Windows XP sind diese Funktionalitäten schon sehr erweitert und bei Windows 7 hat man neben den ursprünglichen Funktionen auch noch zahlreiche zusätzliche Applikationen (z.B. Sidebar).



Abbildung 5: Umstiegsprozess zu einem Software Ökosystem (Eigene Abb.)

Für den Übergang zwischen Software Paket und Software Ökosystem gibt es nun zwei extreme Strategien wie aus Abb. 6 entnommen werden kann. Beim gezielten Vorgehen werden nur für bestimmte Bereiche der Software gezielt Partner gesucht. Diese erhalten dann hohe Zugriffsrechte auf den für sie relevanten Code und dies wird in Verträgen festgehalten. Ein Beispiel hierfür ist MS Office. Beim freien Vorgehen, werden alle Bereiche der Software geöffnet und es findet keine Partnersuche statt. Der Code wird im Internet zur freien Verfügung gestellt wobei er als Basis dient. Hier bestimmt dann vor allem der Wettbewerb, welche Applikationen genutzt werden und welche nicht. Dieser Vorgang ist dann quasi Open Source. Hierfür ist Open Office ein geeignetes Beispiel (vor allem vor der Übernahme von Sun durch Oracle).

Ein Software Ökosystem durchläuft mehrere Stadien, bis es das theoretisch effizienteste Stadium erreicht hat. Dies ist aber de-facto noch nicht der Fall. Im ersten Stadium sucht der Softwarehersteller Partner und bindet sie dann meist durch Verträge oder andere Bedingungen, wobei zwischen den Partnern noch keine Kommunikation stattfindet und nur der Hersteller eine aktive Rolle hat. Im 2. Stadium nutzen die Partner untereinander ihre einzelnen Kompetenzen um so ein noch besseres Produkt zu erstellen. Ein Beispiel hierfür wäre die Applikation zum Drucken von Tabellen. Dort können Partner die sich gut mit Tabellen auskennen mit Partner die sich im Bereich Drucken spezialisiert haben zusammenschließen um die Qualität zu steigern. Aus diesen Verbindungen lernt dann der Software Hersteller und verwendet diese Erkenntnisse um auch die Beziehung zu und zwischen den anderen Partnern zu verbessern. Optimal wäre es dann wenn, wie im 4. Stadi-

Gezielt	Frei (Open Source)
Bestimmte Bereiche	Alle Bereiche offen
Gezielte Partnersuche	Keine Partnersuche
Hohe Zugriffsrechte	Plattform als Basis
Verträge	Wettbewerb

Abbildung 6: Zwei extreme Strategien [BO09]

um gezielt, alle Partner zusammenarbeiten um den bestmöglichen Outcome zu erzeugen. Hierbei ist zu erwähnen, dass es sich nur um ein theoretisches Modell handelt, für das es in der Praxis noch kein Beispiel gibt.

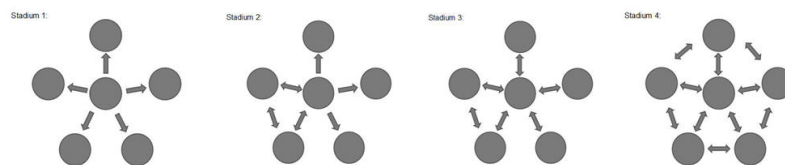


Abbildung 7: Stadien eines Software Ökosystems [FA01]

4 Auswirkungen auf die Software-Programmierung

Ein Software Ökosystem hat sehr viele Auswirkungen auf die Software-Programmierung, dabei ist fast jeder Bereich einer Firma betroffen. Erstens die Koordinationsmechanismen. Die alten Prozesse der Firma sind optimiert für interne Abläufe und sehen daher meist Mitwirkung von Dritten nicht vor, was sie jetzt zum Teil ineffektiv oder gar unbrauchbar macht. Es müssen also neue Prozesse entwickelt werden. Mit der Anzahl der Partner steigt auch der Koordinationsaufwand dieser Partner, wodurch hier meist nun mehr Leute beschäftigt werden müssen. Zweitens sind auch die Ingenieursfähigkeiten betroffen. Zum einen soll die Software in regelmäßigen Abständen weiterentwickelt werden und zum anderen werden aber immer mehr Kunden angesprochen. Hier müssen die Entwickler also sehr flexibel bleiben und die Terminvorgaben einhalten. Ein weiterer Punkt ist die Software-Zusammensetzung, siehe Abb. 8.

Dies hier ist nur ein sehr kleiner Ausschnitt aus einem Produkt, aber veranschaulicht schon welche Schwierigkeiten auftreten werden. Der Kunde kann aus sehr vielen verschiede-

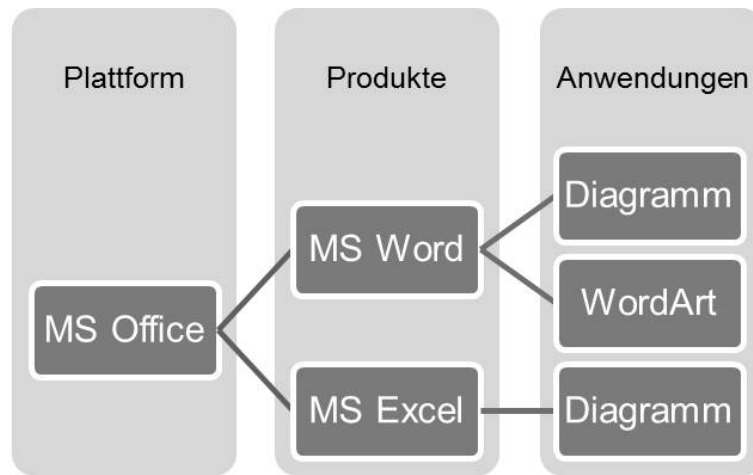


Abbildung 8: Produktzusammensetzung [BO09]

nen Variationen wählen und diese müssen nun alle getestet werden. Diese Tests sind sehr aufwändig, da die Kombinationsmöglichkeiten mit der Größe der Software rasant ansteigen. Das Diagramm in Excel und Word müssen für den Kunden gleich zu bedienen sein, dass er immer das Gefühl hat, als käme alles aus einer Hand. Des Weiteren sollten auch alle Kunden immer die gleichen Erfahrungen mit dem Produkt machen, sowohl im Bereich Funktionsweise als auch Support und Aussehen, um nur ein paar zu nennen.

5 Software Ökosysteme führen zu IT-Transformationen

Durch Software Ökosystem entstehen neue Schnittstellen zu anderen Firmen und es werden neue Prozesse benötigt, welche auch unterstützt werden müssen. Dies sind sehr viele Anforderungen mit der die IT in einem Unternehmen zurecht kommen muss. Um das zu bewerkstelligen bedarf es einer Unternehmensübergreifenden IT-Transformation.

6 Quellen

[BO09] From Software Product Lines to Software Ecosystems (Bosch, Jan, 2009)

[FA01] Software Acquisition: a business strategy analysis (Farbey, Barbara; Finkelstein, Anthony, 2001)

[JA09] A Sense of Community: A Research Agenda for Software Ecosystems (Jansen, Slinger; Finkelstein, Anthony; Brinkkemper, Sjaak, 2009)