



## SOFTWARE-ÖKOSYSTEME

### Zusammenfassung des Interviews mit Slinger Jansen

Welche Anforderungen ergeben sich für Unternehmen, die insbesondere in Software-Ökosystemen aktiv werden wollen? Auf welche Herausforderungen müssen sich etablierte Softwarehersteller einstellen und wo liegt eigentlich der Vorteil eines Software-Ökosystems? Diese und andere Fragen hat uns Slinger Jansen im Interview beantwortet.

Slinger Jansen ist Assistant-Professor an der Universität von Utrecht und Mitautor von grundlegenden Werken zur Erforschung von Software-Ökosystemen. Einer seiner Forschungsschwerpunkte liegt insbesondere auf den Möglichkeiten, die Software-Ökosysteme Unternehmen und anderen Organisationen bieten.

#### Das Software-Ökosystem

Von einem wirtschaftlichen Ökosystem wird gesprochen, wenn sich um ein Produkt oder einen Standard herum ein loses Netzwerk entwickelt, in dem ergänzende Produkte – auch Dienstleistungen – angeboten werden. Die Akteure in einem Ökosystem sind in der Regel unabhängig und recht heterogen. Unter ihnen können Großunternehmen und Mittelständler, aber auch Freelancer sein.

Gerade bei Software-Ökosystemen sind oft auch nicht-kommerzielle Akteure vorhanden. Dabei gibt es oft, aber nicht immer, einen „Keystone Player“, der mit seinen zentralen Produkten die Entwicklung des Ökosystems wesentlich beeinflussen kann. Sein Einfluss auf die restlichen Akteure ist jedoch normalerweise indirekt. Die Akteure können sogar teilweise im Wettbewerb mit dem Keystone Player stehen.

Software-Ökosysteme sind ein rein abstraktes Modell. Einer sichtbaren Repräsentation, etwa eines Konsortiums, Interessensverbandes oder dergleichen, bedarf es nicht. Sofern solche Organisationen existieren, bilden sie in der Regel nur eine Teilmenge des Ökosystems ab.

In einem Software-Ökosystem können zum Beispiel mehrere Unternehmen als Keystone Player eine Plattform anbieten, die andere Unternehmen wiederum für ihre speziellen Produkte nutzen. Als Beispiel können hier Apples App Store, der Android Market aber auch das Ökosystem rund um SAP gelten. In allen drei Software-Ökosystemen geben die Keystone Player – also Apple, Google und SAP – die Rahmenbedingungen vor. Die anderen Unter-

nehmen wiederum können die Plattformen gegen Gebühren u. a. als etablierte Distributionskanäle für ihre spezialisierten Produkte nutzen und profitieren obendrein vom qualitativen Status der Plattform wie auch des Plattformbetreibers.

Die Keystone Player wiederum profitieren neben den Einnahmen aus der Bereitstellung ihrer Plattform auch noch davon, dass ihr Produkt, zum Beispiel Android oder iOS, durch eine Vielzahl von fremden Produkten den unterschiedlichsten Anforderungen genügt, ohne dass die Keystone Player alle Anforderungen selbst evaluieren müssen. Das wiederum sorgt für eine hohe Verbreitung der Plattform.

### Vorteile von Software-Ökosystemen

Software-Ökosysteme sind letztendlich wirtschaftliche Ökosysteme, nur dass hier „eine softwarezentrierte Perspektive eingenommen wird“, wie Slinger Jansen erklärt. Diese Perspektive kann auch noch weiter gefiltert werden, „zum Beispiel in eine kommerzielle und eine Forschungsperspektive.“ Die Mechanismen des Ökosystems bleiben dabei die gleichen: Die Wirtschaftsgemeinschaft wird durch einzelne Unternehmen, Organisationen und Einzelpersonen, also Organismen im Sinne eines Ökosystems, unterstützt. Die Entwicklung im Laufe der Zeit wird dabei in der Regel von einem oder mehreren zentralen Unternehmen, den Keystone Players, vorgegeben.

Die Formen der Unterstützung durch Dritte sind dabei vielseitig, im Bereich der Software-Ökosysteme sticht allerdings ein Bereich besonders heraus: „Alle Anforderungen an das eigene Softwareprodukt im eigenen Unternehmen im Laufe der Zeit erfüllen zu wollen ist kritisch. Um sich das zunehmend aufwändige Domain-Engineering zu ersparen, können Dritte hier unterstützen und vom Pro-

dukt des Initiators profitieren, während das eigene Unternehmen als Keystone Player die generelle Richtung vorgeben kann“ erläutert Jansen. Wenn es das wesentliche Ziel eines Softwareunternehmens ist, dass seine eigenen Produkte und Services möglichst oft Verwendung finden, können hier also Dritte spezielle Anforderungen bedienen und so die Zielgruppe entsprechend erweitern, während das eigene Unternehmen als Keystone Player zum Beispiel als Betreiber der Plattform die generelle Richtung vorgeben kann. Im Ökosystem Software werden so Nischen von Dritten besetzt, während die Plattform als Basis für Entwicklungen dient.

### Offenheit als Motor für Innovationen

Allerdings ist in diesem System die Offenheit im Bezug auf eigene Produkte und Standards eine „essentielle Voraussetzung für ein erfolgreiches Software-Ökosystem, da die Partner sonst keine Basis haben, auf welcher sie ihre Entwicklungen aufbauen können“, wie Jansen meint. An diesem kritischen Punkt, gerade was die Sorge um die Qualität der eigenen Produkte angeht, bleibt es erst einmal jedem Unternehmen selbst überlassen, welche Regeln es für sein Ökosystem aufstellt. Doch letztendlich bestimmen die Regeln, die man dem eigenen Software-Ökosystem zu Grunde legt, wie leistungsstark das eigene Unternehmen ist. Das hängt wiederum im hohen Maße mit der Exzellenz des betreffenden Produkts zusammen und ist auch ausschlaggebend dafür, wie erfolgreich das eigene Produkt sein wird.

Slinger Jansen meint dazu: „Es ist vielleicht ein bisschen altmodisch, in nur einer Domäne durch einen sehr restriktiven Umgang mit der Offenheit der eigenen Produkte unbedingt der König bleiben zu wollen“, wenn es doch genug Beispiele dafür gebe, wie man trotz einer Öffnung seine Geschäftsfelder dominieren kann – siehe Apple. Außerdem komme

„im Laufe der Zeit jedes Produkt an den Punkt, an dem nicht mehr alle Anforderungen ohne Probleme im eigenen Haus erfüllt werden können.“

### Herausforderungen beim Einsatz

Durch die Art der Zusammenarbeit in einem Software-Ökosystem, das nach Jansen „zumindest mehr als zwei Teilnehmer umfassen sollte“, entstehen allerdings neue Herausforderungen, um die Partner, die für das eigene Produkt oder die Plattform arbeiten, an eben dieses zu binden. Insbesondere in den Bereichen Architektur, Prozesse und Organisation sollten einige Leitlinien beachtet werden, um aus einer temporären Kooperation ein florierendes Ökosystem wachsen zu lassen.

Im Bereich der Architektur sollte das besondere Augenmerk beim Entwickeln in einem Software-Ökosystem auf „der Erweiterbarkeit, der Robustheit sowie der Offenheit im Bezug auf die Möglichkeiten der Zusammenarbeit mit Dritten liegen“, wie Slinger Jansen aus seiner Forschung weiß. Anderenfalls kann es bedingt durch die unabhängigen Entwicklungen der Partner früher oder später zu Instabilitäten und somit zu einem Qualitätsverlust des Produkts oder der eigenen Plattform führen. Dies wiederum kann zur Folge haben, dass sich Partner schnell anderen Software-Ökosystemen zuwenden. Auch in den Prozessen wird eine Neuausrichtung insbesondere „des Softwareproduktmanagements unumgänglich sein, da die Anforderungen an das Produkt oder die Plattform eher von der Community bzw. den Partnern abhängen als von potentiellen Endkunden.“

### Die Community als kritische Masse

Der letzte und vielleicht wichtigste Aspekt betrifft die Organisation in einem Software-Ökosystem. „Da die Community – also die Partner im weitesten Sinn – in einem Software-Ökosystem eine kritische Masse ist, müssen die Anforderungen der Partner an das eigene Produkt oder die Plattform in einem stärkeren Maß berücksichtigt werden, als es außerhalb dieses Systems nötig wäre“, wie Jansen erläutert. Darüber hinaus ist die Möglichkeit der Mitbestimmung im Software-Ökosystem für die verschiedenen Partner ein klarer Anreiz für die Entscheidung zu Gunsten der eigenen Plattform.

Als ein negatives Beispiel nennt Slinger Jansen in diesem Zusammenhang Twitter: „Twitter hat immer wieder bestimmte Entwicklungen von Dritten aufgekauft und anschließend als De-facto-Standard behandelt, was zur Folge hatte, dass sämtliche konkurrierenden Entwicklungen damit abgewürgt wurden.“ Diese Art der Organisation, bei welcher die Community wenig bis gar nicht an relevanten Entscheidungen beteiligt ist, hemmt Investitionen in innovative Lösungsansätze und steht somit einem florierenden Software-Ökosystem im Wege.

Allgemein lassen sich in der Organisation von Software-Ökosystemen vor allem die Aspekte der Zentralisierung bzw. Dezentralisierung unterscheiden. In dezentral organisierten Ökosystemen, zum Beispiel Microsofts CRM, erfolgt nur ein kleiner Teil der Entwicklung direkt im Unternehmen. Große Teile des Systems entstehen in Partnerunternehmen. Ein anderes Beispiel für die komplette Dezentralisierung ist das Software-Ökosystem rund um Debian. Hier sind auch die Möglichkeiten der Mitbestimmung durch die Community viel stärker ausgeprägt als in zentralisierten Systemen.

Googles Android dagegen ist ein Beispiel für die zentralisierte Organisation eines Software-Ökosystems, da hier die meisten Entwicklungen im Unternehmen erarbeitet werden und eher weniger von Dritten. Google ist in diesem Sinne nicht nur der Keystone Player im eigenen Ökosystem, sondern entwickelt eben auch selbst in den unterschiedlichsten Domänen. „Allerdings kann das auch an dem Umstand liegen, dass Googles Android und die Community rund herum noch recht jung sind“, wie Slinger Jansen anführt.

### **„Get big, get niche or get out“**

„Auf längere Sicht jedenfalls werden die Communities rund um erfolgreiche Plattformen diese in stärkerem Maße tragen als es die Unternehmen,

welche die Plattformen anbieten, selbst tun“, prognostiziert Slinger Jansen. Dabei wird auch das Selbstverständnis der Partner als Teil eines Software-Ökosystems zum Erfolg des Nischenunternehmens wie auch des Plattformbetreibers als Keystone Player beitragen, da zunehmend auf das jeweilige Ökosystem abgestimmte innovative Geschäftsideen etabliert werden können. „Generell wird das Sprichwort ‚Get big, get niche or get out‘ jedenfalls in Zukunft noch an Relevanz gewinnen“ meint Slinger Jansen — nicht zuletzt sicherlich durch die zunehmende Orientierung an Ökosystemen als Geschäftsmodell.

---

Vincent Wolff-Marting, Philipp Rothmann