

Bewertung von Open Source Frameworks als Ansatz zur Wiederverwendung

Christian Neumann
Abteilung für Informationswirtschaft
Wirtschaftsuniversität Wien
Augasse 2-6
1090 Wien, Austria
christian.neumann@ai.wu-wien.ac.at

Abstract: In diesem Artikel wird die Verwendung von Real Options als Modell für die Entscheidung über alternative Strategien in der Softwareentwicklung vorgestellt. Der Fokus liegt hierbei auf einer Bewertung der Vor- und Nachteile der Verwendung von Open-Source-Software Frameworks.

1 Einleitung

Die Wiederverwendung von Software ist ein integraler Bestandteil des modernen Software-Engineering und kann sich positiv auf Qualität und Produktivität auswirken [Lim94] sowie in unterschiedlichsten Ausprägungen betrieben werden. In der Literatur wird Wiederverwendung zumeist auf technischer Ebene diskutiert [PD93], wonach eine mögliche technische Umsetzung der Einsatz eines Frameworks ist. Ein Framework besteht aus einer Menge von zusammenarbeitenden Klassen, die einen wiederverwendbaren Entwurf für einen bestimmten Anwendungsbereich implementieren. Ein Framework ist somit eine Referenzarchitektur für eine Anwendung, die grundlegende Strukturen und Kommunikationswege vorgibt, jedoch die Details des konkreten Entwurfes offen lässt [Bal00]. Die Verwendung von Frameworks reduziert den Entwicklungsaufwand und kann sich positiv auf die Qualität auswirken, erfordert allerdings im Gegenzug erhöhte Aufwände für das Erlernen und Warten [BMMB99]. Der Einsatz von Frameworks ist besonders bei der Entwicklung von Produktlinien interessant, wo in der Zukunft wiederkehrende, ähnliche Funktionalitäten erwartet werden.

Ist die Integration eines Frameworks in eine zukünftige Produktlinie auf technischer Ebene sinnvoll, müssen strategische Entscheidungen getroffen werden, ob das Framework intern entwickelt oder extern beschafft werden soll. Die interne Entwicklung eines Framework hat den Vorteil, dass der Source-Code verfügbar ist und daher an zukünftige Anforderungen angepasst werden kann. Dem gegenüber steht der gesteigerte initiale Aufwand bei der Erstellung des Frameworks. Eine andere Möglichkeit ist der Einsatz von kommerziellen Frameworks (commercial-off-the-shelf, COTS), die von externen Anbietern erworben werden. Auf Grund von Economies-of-Scale sind die Lizenzkosten in der Re-

gel geringer als eine vergleichbare Eigenentwicklung. Der nicht vorhandene Source-Code führt allerdings zu einigen Nachteilen: Beschränkung auf die vorhandene Funktionalität, Abhängigkeit vom Hersteller, mangelnde Flexibilität sowie ungewisse Qualität [BA99]. In letzter Zeit hat sich eine weitere Möglichkeit etabliert, die als Symbiose der oben genannten Entscheidungen zu sehen ist - die Verwendung von Open-Source-Software (OSS). OSS zeichnet sich durch Offenheit des Source-Codes und eine verteilte Entwicklung aus. Inzwischen existiert zum einen OSS-Projekte, die sich als eigenständige Programme zu einem de facto Standard entwickelt haben (z.B. Linux, Apache) und zum anderen Projekte, die für die Integration in den Entwicklungsprozess konzipiert wurden (z.B. diverse Java-Bibliotheken). Die Offenheit des Source-Codes macht den Einsatz von OSS im Software Engineering Prozess sehr interessant, da die Vorteile der Verwendung fertiger Komponenten (geringer Aufwand bei der Implementierung) und die Möglichkeit den Source-Code zu verändern, kombiniert werden können. OSS Frameworks sind vor allem im Bereich von Web-Applikationen zu finden, können aber auch komplexe Funktionalitäten von Oberflächen-Elementen bereitstellen (z.B. Netbeans, Eclipse).

2 Investitionsentscheidungen in der Softwareentwicklung

Der Einsatz eines Frameworks birgt die gleichen Kosten und Risiken wie jede andere finanzielle Investition auch. Eine Investition lohnt nur, wenn die Einsparungen durch die Wiederverwendung größer als die Ausgaben sind, also einen positiven *Return-on-Investment* haben. Insbesondere für derartige Investitionsvorhaben bieten sich dynamische Bewertungsmodelle an, die den zeitlichen Verlauf eines Projektes berücksichtigen. Weit verbreitet ist die Kapitalwertmethode, die die erwarteten addierten Cash-Flows CF_t zu einem Zeitpunkt t mit einem risikolosen Zinssatz r zum Zeitpunkt $t = 0$ diskontiert:

$$K = \sum_{t=0}^n \frac{CF_t}{(1+r)^t} \quad (1)$$

Die Kapitalwertmethode bietet zwar die Möglichkeit, den Zinsfaktor r um einen Risikozuschlag zu erweitern, strategische Managemententscheidungen können mit diesem Modell aber nur unzureichend abgedeckt werden. Aus diesem Grunde wurde in neuester Zeit ein Instrument aus der Finanztheorie in die Investitionsrechnung integriert. Diese so genannten Real Options leiten sich aus Finanzoptionen ab und bewerten monetär ein Recht, aber keine Verpflichtung, eine Aktion in der Zukunft auszuführen [AK99]. In einem Software-Projekt können zu jedem Zeitpunkt Managemententscheidungen getroffen werden, die sich auf den Verlauf auswirken. So kann das Projekt z.B. verschoben, eingestellt oder auch ausgeweitet werden. Das Real Options Modell bietet die Möglichkeit diese Flexibilitäten zu bewerten und in die Entscheidungsfindung mit ein zu beziehen. Somit ergibt sich für den Kapitalwert

$$K = \sum_{t=0}^n \frac{CF_t}{(1+r)^t} + \Omega \quad (2)$$

wobei Ω den Wert der Flexibilität darstellt.

Die Integration von Frameworks in den Entwicklungsprozess ist eine weit reichende Investitionsentscheidung, deren Verlauf von einer Vielzahl von Faktoren abhängt. Real Options bieten einen sinnvollen Ansatz, um den Wert dieser Form der Wiederverwendung und korrespondierende Managemententscheidungen zu bewerten. Es existieren einige Studien die Real Options im IT-Bereich [Ben02], bei Wiederverwendung von Frameworks [FF99] und bei OSS Projekten [OP02] einsetzen. Eine ökonomische Betrachtung der Integration von OSS in den Entwicklungsprozess ist bis jetzt noch nicht mit Real Options modelliert worden.

3 Bewertung von OSS Frameworks mit Real Options

Ist der Einsatz eines Frameworks auf technischer Ebene sinnvoll, gilt es drei verschiedene strategische Alternativen zu evaluieren: Entwicklung und Verwendung eines eigenen Frameworks, Verwendung eines COTS Frameworks und Verwendung eines OSS Frameworks. Für diese Szenarien müssen initiale und laufende Investitionen, Risiken und mögliche Entscheidungen bestimmt und verglichen werden.

3.1 Vorgehensweise

Aus aktueller Sicht eignet sich die Real Options Theorie sehr gut, um die verschiedenen Investitionsentscheidungen zu bewerten, die mit einem hohen Risiko und/oder Unsicherheit behaftet sind. Die Entwicklung eines Modells basierend auf dieser Theorie gliedert sich in vier aufeinander aufbauende Stufen:

1. **Festlegen der Investitionen, Risiken und möglichen Entscheidungen.** In dieser Phase werden alle potentiellen Risiken erfasst, die während des Projektes auftreten können. Eine Klassifizierung der Risiken ist ebenfalls Bestandteil dieses Abschnittes.
2. **Abbilden der Risiken auf Real Options.** Die in Schritt 1 erkannten Risiken werden auf die Grundtypen der Real Options abgebildet.
3. **Entwickeln alternativer Konfigurationen der Investition.** Auf Basis der Real Options aus Schritt 2 werden verschiedene Szenarien modelliert, die unter Umständen sehr komplex werden können. Daher ist die Reduktion der Konfigurationen ein wichtiger Punkt innerhalb dieser Phase.
4. **Auswerten der Optionen und Konfigurationen.** Die in Schritt 3 erstellten Konfigurationen werden mit verschiedenen mathematischen Verfahren ausgewertet.

3.2 Investitionen

Unter Investitionen werden alle Aufwände verstanden, die während der Projektlaufzeit auftreten. Dies sind zum einen initiale Aufwände, die nur einmalig auftreten (Entwicklung eines eigenen Frameworks, Erlernen bestehender Frameworks) und zum anderen Aufwände die bei der Implementierung der Folgeprojekte entstehen (Integration). Obwohl bei der Verwendung von OSS Frameworks weder Kosten für die Entwicklung des Frameworks noch Lizenzgebühren anfallen, entstehen Aufwände durch die Adaption und Wartung des Frameworks [MRS02]. Für dessen erfolgreichen Einsatz, bedarf es einer Einarbeitungsphase, in der die Handhabung und die Funktionsweise erlernt werden müssen. Der Aufwand für das Erlernen und die Adaption eines Frameworks unterliegt einer Lernkurve und ist zudem von der Qualität der bereitgestellten Software abhängig. Innerhalb der Wartungsphase konkurriert die Weiterentwicklung der OSS Gemeinschaft mit einer eventuellen eigenen Weiterentwicklung, die zusätzliche Kosten verursachen kann, insbesondere wenn diese parallelen Entwicklungsrichtungen kompatibel gehalten werden sollen.

3.3 Entscheidungen und Risiken

Der Identifizierung möglicher Entscheidungen zu verschiedenen Zeitpunkten eines Projektes kommt eine besondere Bedeutung zu. Grundlegende Entscheidungen sind das Einstellen, Fortführen, Ausdehnen oder Anhalten des Projektes, die durch Real Options modelliert werden können. Wird zum Beispiel eine OSS Technologie zu einem de facto Standard, können sich für Unternehmen, die schon frühzeitig auf die Integration von OSS gesetzt haben, Wettbewerbsvorteile ergeben. Die Anwendung einer Wachstums-Option kann den möglichen zusätzlichen Nutzen monetär bewerten. Weiterhin ermöglicht der verfügbare Source-Code den Einsatz von OSS-Frameworks in einem breiten Umfeld. Diese Flexibilität kann mit einer Wechsel-Option modelliert werden. Das Risiko, das ein Projekt nicht mehr weiterentwickelt wird, kann mit einer Anhalte-Option bewertet werden.

3.4 Modellierung

Die angesprochenen Aufwände können mit einem Schätzmodell (z.B. Cocomo II [BAC00]) modelliert werden. Die Risiken und Entscheidungen werden dann auf Real Optionen abgebildet und die drei verschiedenen Szenarien verglichen. Hieraus ergeben sich Handlungsempfehlungen, ob ein Framework eingesetzt, und wenn, ob es selbst entwickelt, zugekauft oder OSS adaptiert werden sollte. Auswirkungen beim Wechsel auf alternative Strategien der Entwicklungen können ebenfalls modelliert werden. Die Auswirkungen der verschiedenen OSS Lizenzen und resultierende mögliche Geschäftsmodelle sind ebenfalls ein Punkt, der modelliert werden könnte.

4 Zusammenfassung

Bei der Entwicklung von Produktlinien ist die Verwendung von Frameworks eine sinnvolle Alternative, um langfristig die Entwicklungskosten zu senken. Am Anfang steht die Überlegung, ob ein eigenes Framework entwickelt werden soll, oder aber fremde Produkte integriert werden können. Die Verwendung von OSS bietet im Rahmen dieses Entscheidungsprozesses eine besonders flexible Alternative. Der unsichere Projektverlauf bedingt allerdings Managemententscheidungen monetär zu bewerten, um eine ideale Strategie auszuwählen.

Literatur

- [AK99] Martha Amram und Nalin Kulatilaka. *Real Options - Managing Strategic Investment in an Uncertain World*. Harvard Business School Press, Boston, 1999.
- [BA99] Barry Boehm und Chris Abts. COTS Integration: Plug and Pray. *IEEE Computer*, 32(1):135–138, 1999.
- [BAC00] Barry Boehm, Chris Abts und Sunita Chulani. Software Development cost destination approaches - A survey. *Annals of Software Engineering*, 20:177–205, 2000.
- [Bal00] Helmut Balzert. *Lehrbuch der Software-Technik: Teil 1: Software-Entwicklung*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2. Auflage, 2000.
- [Ben02] Michel Benaroch. Managing Information Technology Investment Risk: A Real Options Perspective. *Journal of Management Information Systems*, 19(2):43–84, 2002.
- [BMMB99] Jan Bosch, Peter Molin, Michael Mattsson und PerOlof Bengtsson. Object Oriented Frameworks - Problems and Experiences. In Mohamed E. Fayad, Douglas Schmidt und Ralph E. Johnson, Hrsg., *Building Application Frameworks: Object Oriented Foundations of Framework Design*, Seiten 55–82. John Wiley and Sons, 1999.
- [FF99] John M. Favaro und Kenneth R. Favaro. Strategic Analysis of Application Framework Investments. In Mohamed E. Fayad, Douglas Schmidt und Ralph E. Johnson, Hrsg., *Building Application Frameworks: Object Oriented Foundations of Framework Design*, Seiten 565–597. John Wiley and Sons, New York, 1999.
- [Lim94] Wayne C. Lim. Effects of Reuse on Quality, Productivity, and Economics. *IEEE Software*, 11(5):23–30, September 1994.
- [MRS02] Maurizio Morisio, Daniele Romano und Ioannis Stamelos. Quality, Productivity and Learning in Framework-Based Development: an Exploratory Case Study. *IEEE Transactions on Software Engineering*, 28(8):340–357, August 2002.
- [OP02] Jan Opstrup Poulsen. Open-source software in e-government. Bericht, Danish Board of Technology, 2002.
- [PD93] Ruben Prieto-Diaz. Status report: software reusability. *IEEE Software*, 10(3):61–66, May 1993.