
Department Informatik

Technical Reports / ISSN 2191-5008

Dirk Riehle und Detlef Kips

Geplanter Inner Source: Ein Weg zur Profit-Center-übergreifenden Wiederverwendung

Technical Report CS-2012-05

Mai 2012

Please cite as:

Dirk Riehle und Detlef Kips, "Geplanter Inner Source: Ein Weg zur Profit-Center-übergreifenden Wiederverwendung,"
University of Erlangen, Dept. of Computer Science, Technical Reports, CS-2012-05, Mai 2012.



Geplanter Inner Source: Ein Weg zur Profit-Center-übergreifenden Wiederverwendung

Dirk Riehle
dirk@riehle.org
Department Informatik
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Detlef Kips
detlef.kips@cs.fau.de
Department Informatik
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Abstract: Wiederverwendung von Softwarekomponenten verspricht, Softwareentwicklung schneller und günstiger zu machen und die Ergebnisqualität zu steigern. Trotz diverser methodischer Ansätze ist es für viele Softwareentwicklungsorganisationen schwierig geblieben, diese Ziele auch nur ansatzweise zu erreichen. Vor diesem Hintergrund bietet „Inner Source“, die Verwendung von Open-Source-Praktiken in der firmeninternen Softwareentwicklung, neue Chancen. Inner-Source-Software ist Software, die innerhalb eines Unternehmens über Profit-Center-Grenzen hinweg in Gemeinschaftsarbeit entwickelt wird und von allen Abteilungen genutzt werden kann. In diesem Artikel stellen wir die bisher gewonnenen Erfahrungen mit Inner-Source-Entwicklung dar, definieren organisatorische Gestaltungsmöglichkeiten und prognostizieren die Entstehung von Inner-Source-Organisationen, einer neuen Form der Organisation für die Wiederverwendung.

1. EINFÜHRUNG

Die Wiederverwendung von Komponenten bei der Entwicklung von Software verspricht viele Vorteile für Entwicklungsorganisationen. Bei entsprechender Qualität der wiederverwendbaren Komponenten wird das Zielprodukt schneller fertig, in höherer Qualität und zu geringeren Kosten. All dies ermöglicht es der Entwicklungsorganisation, sich verstärkt auf die wettbewerbsdifferenzierenden Merkmale der eigenen Softwareentwicklung zu konzentrieren und effektiver am Markt zu agieren.

Was theoretisch naheliegend ist, ist praktisch nicht immer einfach zu realisieren. Berichte aus Forschung und Praxis zeigen unterschiedliche Probleme auf, die sich einer Softwareentwicklungsorganisation bei dem Versuch, systematische Wiederverwendung einzuführen, in den Weg stellen [11]. Die Probleme reichen von der eigentlichen Entwicklung der Komponenten bis zu deren Anwendung.

Dieser Artikel stellt das Konzept des „Inner Source“ vor. *Inner-Source-Software* ist Software, die innerhalb eines Unternehmens über Profit-Center-Grenzen hinweg in Gemeinschaftsarbeit entwickelt wird und von allen Abteilungen genutzt werden kann¹. Ggf. kann der Nutzerkreis auf die beitragenden Entwicklungsabteilungen eingeschränkt werden. Die Unterstützung von Inner-Source-Entwicklung ist eine Maßnahme, mit der Wiederverwendung gezielt gefördert werden kann. Inner-Source-Entwicklung erweitert traditionelle organisatorische Maßnahmen (etwa die Schaffung einer zentralen Abteilung für wiederverwendbare Komponenten) um neue Praktiken.

Unsere Erfahrungen zeigen, dass Profit-Center-übergreifende Zusammenarbeit zur Entwicklung wiederverwendbarer Komponenten in großen Entwicklungsorganisationen keineswegs selten geschieht. In der Forschung hat Inner Source jenseits der Darstellung von Einzelfallbeispielen aber bisher keine größere Berücksichtigung gefunden.

Dieser Artikel leistet die folgenden Beiträge:

1 Hier und im weiteren Text verwenden wir die Begriffe "Profit Center" und "Abteilung" synonym.

- Aufbereitung der Literatur zum Thema Inner-Source-Entwicklung
- Bestimmung und Anwendung eines Modells zur Klassifikation der Fallbeispiele
- Definition von Gestaltungsoptionen für die Organisation von Wiederverwendung
- Diskussion von Problemen bei der Wiederverwendung und von deren Behebung
- Vorstellung einer daraus abgeleiteten Organisationsform, der „Inner-Source-Organisation“.

Der Artikel ist wie folgt strukturiert: Abschnitt 2 fasst die existierende Literatur zur Organisation von Wiederverwendung zusammen. Abschnitt 3 rezipiert die existierende Literatur zu Inner-Source-Ansätzen und entwickelt ein Modell der Gestaltungsoptionen. Abschnitt 4 stellt ein verallgemeinertes Modell der Organisation von Wiederverwendung, inklusive Inner-Source-Entwicklung, vor und erfasst existierende Beispiele. Abschnitt 5 leitet aus dem Modell eine neuartige firmeninterne Organisationsform ab, die bisher nur hypothetisch existiert, aber zumindest in der Theorie signifikante Vorteile gegenüber bekannten Organisationsformen aufweist. Abschnitt 6 beschließt den Artikel.

2. FORSCHUNG ZUR SOFTWARE-WIEDERVERWENDUNG

Im Folgenden beziehen wir uns auf einen Überblicksartikel von Frakes und Kang [4], der die Forschung zum Thema „Softwarewiederverwendung“ zusammenfasst und das Ergebnis eines Konsensprozesses im Nachklang zur *ICSR (International Conference on Software Reuse)* 8 darstellt. Allen Betrachtungen liegt die Annahme zugrunde, dass die Bestimmung, Erstellung, Wartung und Anwendung von wiederverwendbaren Softwarekomponenten im Rahmen von Produktlinienentwicklung stattfindet. Eine Produktlinie liegt vor, wenn Softwareunternehmen oder Entwicklungsabteilungen in einem Anwenderunternehmen verschiedene Produkte entwickeln, die alle der gleichen Anwendungsdomäne zuzurechnen sind [8].

Frakes und Kang unterscheiden zwei Ansätze zur Bestimmung und Erstellung wiederverwendbarer Komponenten: den *proaktiven* und den *reaktiven Ansatz*. Im *proaktiven Ansatz* wird per Vorabinvestition die Anwendungsdomäne untersucht, es werden etwaige Komponenten identifiziert und bewertet und dann ggf. entwickelt. Im *reaktiven Ansatz* wird eine wiederverwendbare Komponente nur dann erstellt, wenn es sich während der normalen Produktentwicklung als vorteilhaft erweist, diese Komponente zu entwickeln. Einen Mittelweg stellt der „*extraktive*“ *Ansatz* dar, bei dem existierende Software untersucht wird, um etwaige redundante Komponenten zu erkennen [8]. Diese redundanten Komponenten werden dann ggf. in wiederverwendbare Komponenten umgewandelt.

Weiterhin unterscheiden Frakes und Kang einen *zentralisierten* und einen *verteilten Ansatz*, wiederverwendbare Komponenten zu entwickeln. Im *zentralisierten Ansatz* wird eine Organisationseinheit für die anfallenden Wiederverwendungsaufgaben etabliert. Im *verteilten Ansatz* werden die Aufgaben auf existierende Organisationseinheiten verteilt, ohne dass es einen zentralen organisatorischen Anlaufpunkt gäbe. Im zentralisierten Ansatz nimmt eine zentrale Organisationseinheit sowohl die Bestimmung, Entwicklung und Wartung, als auch die Verwaltung und Betreuung der Anwendung von wiederverwendbaren Komponenten wahr. Eine zentrale Organisationseinheit verlangt Vorabinvestitionen, hat aber den Vorteil, dass die Zuständigkeiten klar definiert sind. Bei verteilter Entwicklung werden dagegen wiederverwendbare Komponenten in den einzelnen Projekten der Produktlinie entwickelt und später zu einer gemeinsam genutzten Bibliothek beigesteuert. Wesentlicher Vorteil der verteilten Entwicklung sind die reduzierten Kosten, da keine separate Organisationseinheit etabliert werden muss und die Entwicklung der Komponenten reaktiv im Rahmen der laufenden Projekte geschieht.

Es fallen die folgenden Probleme ins Auge:

- Ohne zentrales Wiederverwendungskonzept ist die Koordination schwierig.
- Komponenten passen ggf. nicht für Projekte, die an ihrer Entwicklung nicht beteiligt waren.
- Projektverantwortliche investieren keinen Aufwand in Wiederverwendung, wenn die Kosten/Nutzen-Analyse für ihr Projekt kein klar positives Ergebnis zeigt.

	Zentrale Einheit (E1)	Gemeinschaftsarbeit (E2 - En)
Bestimmung	x	
Entwicklung	x	
Wartung	x	
Verwaltung	x	
Betreuung	x	

Abb. 1: Traditionelle Organisation mit zentraler Einheit für Wiederverwendung

	Zentrale Einheit (E1)	Gemeinschaftsarbeit (E2 - En)
Bestimmung		x
Entwicklung		x
Wartung		x
Verwaltung	x	x
Betreuung		x

Abb. 2: ISE bei HP, IBM, SAP und Microsoft [2] [3] [13] [10]

	Zentrale Einheit (E1)	Gemeinschaftsarbeit (E2 - En)
Bestimmung	x	x
Entwicklung	x	x
Wartung	x	x
Verwaltung	x	
Betreuung	x	

Abb. 3: Geplante ISE und Wiederverwendung bei Philips Healthcare [14]

	Zentrale Einheit (E1)	Gemeinschaftsarbeit (E2 - En)
Bestimmung	x	x
Entwicklung	x	x
Wartung	x	x
Verwaltung	x	
Betreuung	x	x

Abb. 4: Naheliegende Organisation für Wiederverwendung mit Inner Source

Frakes und Kang beschränken ihren Überblick zum Stand der Kunst im Bereich der Organisation von Wiederverwendung auf die wenigen o.g. Gesichtspunkte. Alles Weitere handelt von technischen Aspekten wie Metriken, Sprachen und Rahmenwerken.

3. FORSCHUNG ZU INNER SOURCE

3.1 Definition von Inner Source

Inner-Source-Software ist Software, die innerhalb eines Unternehmens über Profit-Center-Grenzen hinweg in Gemeinschaftsarbeit entwickelt wird und allen Abteilungen zur Verfügung steht (unsere Definition, abgeleitet von [2]). Ggf. kann der Nutzerkreis auf die beitragenden Entwicklungsabteilungen eingeschränkt werden.

Wesentliche Eigenschaft von *Inner-Source-Entwicklung (ISE)* ist ein transparenter, der Open-Source-Entwicklung entlehnter Entwicklungsprozess. Diese Transparenz hat zur Folge, dass sich Projekte und Organisationseinheiten innerhalb eines Unternehmens für andere Projekte und Organisationseinheiten öffnen müssen: Jeder Mitarbeiter kann Artefakte von Projekten, an denen er qua Organisation nicht beteiligt ist, einsehen und ggf. konstruktiv beitragen. Ein wesentliches Mittel zur Realisierung dieser Transparenz ist die Verwendung einer zentralen *Software-Forge*. Eine Software-Forge ist eine web-basierte Werkzeugplattform für die Softwareentwicklung. Die meisten großen Softwarefirmen (IBM, Microsoft, SAP, etc.) verfügen über eine Software-Forge [3] [10] [13].

Die durch ISE und eine entsprechende Werkzeugunterstützung herbeigeführte Transparenz birgt große Chancen: Sie ermöglicht effizientere Selbstorganisation, besseres Wissensmanagement und allgemein mehr Wiederverwendung [13]. Andererseits löst diese Transparenz bei vielen Projektmanagern und Abteilungsleitern Angst vor Kontrollverlust aus. Wir adressieren Chancen und Probleme weiter unten.

	ZE	A1	A2	A3	An
Anforderungsermittlung	x	x	x		
Projektleitung	x				
Softwareentwicklung	x	x	x	x	

Abb. 5: Beispielhafte Rollen- und Aufgabenverteilung in einem IS-Projekt

Wir unterscheiden zwischen *geplanter* und *ungeplanter ISE*. Ersteres bezeichnet Inner-Source-Entwicklungen, die als traditionell geplante Projekte der Organisation durchgeführt werden. *Ungeplante ISE* ergibt sich dagegen durch Eigeninitiative von Mitarbeitern ohne initiale Managementeinwirkung und häufig sogar ohne Managementgenehmigung. Ungeplante ISE kann von Managementseite aus unterstützt werden, wenngleich primär durch statistische Methoden, die dazu dienen, dem Zufall „auf die Sprünge zu helfen“. Googles „20% Zeit“ [6] und SUSEs „hack week“ [1] sind Beispiele für solche Bemühungen. In diesem Artikel verfolgen wir den Ansatz der ungeplanten ISE nicht weiter, sondern verweisen auf unsere früheren Arbeiten [13].

3.2 Grundlegende Gestaltungsoptionen

Die Entscheidung für eine gezielte Förderung von Wiederverwendung geht in großen Unternehmen oder Behörden organisatorisch oft mit der Entscheidung einher, ob eine dediziert zuständige, zumeist zentrale und von spezifischen Produktlinien unabhängige Organisationseinheit für Wiederverwendung geschaffen werden soll. Der Inner-Source-Gedanke erweitert den Möglichkeitenraum um die Entscheidung, wiederverwendbare Komponenten Profit-Center-übergreifend zu entwickeln, mit oder ohne Involvierung einer zentralen Organisationseinheit. Somit stehen zwei Entscheidungen an:

- E1. Etabliert man eine zentrale Organisationseinheit für Wiederverwendung?
- E2. Entwickelt man wiederverwendbare Komponenten Profit-Center-übergreifend?

Diese beiden Entscheidungen sind voneinander unabhängig und schließen sich nicht gegenseitig aus. Sie werden jeweils auf die verschiedenen Aktivitäten resp. Aufgaben in der Organisation für Wiederverwendung angewendet. Die wesentlichen Aktivitäten sind:

- A1. Bestimmung von wiederverwendbaren Komponenten
- A2. Entwicklung von wiederverwendbaren Komponenten
- A3. Wartung von wiederverwendbaren Komponenten
- A4. Verwaltung von wiederverwendbaren Komponenten
- A5. Betreuung der Anwendung von wiederverwendbaren Komponenten

Es ergibt sich eine 2x5-Matrix, welche die Möglichkeiten der Zuweisung von Aufgaben an Organisationseinheiten erfasst. Der traditionelle Ansatz, in dem eine zentrale Wiederverwendungseinheit alle Aufgaben exklusiv übernimmt, ist in Abbildung 1 dargestellt.

3.3 Erfahrungen mit Inner-Source-Entwicklung

Existierende Literatur zum Thema Inner Source ist durchgängig beispielhaft; eine grundlegende repräsentative Untersuchung fehlt bisher. In diesem Unterabschnitt rezipieren wir diese Fallbeispiele nach dem eben eingeführten Modell und diskutieren im nächsten Unterabschnitt die Probleme, die in der Praxis mit ISE-basierten Ansätzen auftreten.

Dinkelacker et al. haben den Begriff „Inner Source“ geprägt [2] [9]. Es handelt sich um Software, welche innerhalb eines Unternehmens allen interessierten Parteien zur Verfügung steht und auch gemeinschaftlich entwickelt wird. Einem Projektleiter steht es hierbei frei, ob er die Mitarbeit anderer Abteilungen an seinem Projekt erlaubt. Diese Offenheit ermöglicht u.a. die schnellere Verbreitung von

Softwarekomponenten und verbessert ihre Qualität durch Beiträge aus einem weiten Nutzerkreis. Wesentlich für den Erfolg von Inner Source ist, dass alle Komponenten an einem Ort gefunden werden können. Dieser Ort, die Software-Forge, wird von einer zentralen Gruppe bereitgestellt.

Abb. 2 stellt die organisatorische Wahl von Hewlett Packard dar [2]. IBM verfügte noch vor HP über ein nicht formalisiertes Inner-Source-Programm, hat darüber aber nichts veröffentlicht. Unsere Information stammt aus einem Interview mit Steve Fox, dem Gründer von „*IBM's Internal Open Source Bazaar*“ (IIOB) [3]. Der IIOB ist eine Forge, die allen Entwicklungseinheiten innerhalb von IBM die freie Entwicklung von Softwarekomponenten ermöglicht. Dies führte zum selben organisatorischen Aufbau wie bei HP, nämlich der zentralen Verwaltung der Komponenten und der verteilten ggf. gemeinschaftlichen Bestimmung, Entwicklung, Wartung und Betreuung der Komponenten. Dies ist auch die Konfiguration, mit der weitere Softwarefirmen entsprechende Erfahrungen gesammelt haben, z.B. SAP [13] und Microsoft [10].

Philips Healthcare ging weiter als die bisher angeführten Unternehmen und integrierte ISE mit einer traditionellen zentralen Organisationseinheit für die Wiederverwendung [14]. Ziel war es, ein einheitliches Wiederverwendungsprogramm zu schaffen. Die Lösung von Philips Healthcare entwickelte sich über mehrere Stufen, deren letzter Stand eine zentrale Organisationseinheit für das Management der Wiederverwendung ist, die über eine jährliche Pauschalgebühr finanziert wird. Diese Einheit verwaltet und betreut wiederverwendbare Komponenten, arbeitet aber bei der Bestimmung, Entwicklung und Wartung derartiger Komponenten mit den verwendenden Abteilungen zusammen. Diese Zusammenarbeit in der Entwicklung, bei der die nutzenden Abteilungen Entwicklungskapazität bereitstellen, geht über die traditionelle Organisation für Wiederverwendung hinaus. In einem weiteren Schritt sollen die nutzenden Abteilungen auch ohne Vermittlung und Beteiligung einer zentralen Einheit gemeinsam wiederverwendbare Komponenten entwickeln. Abb. 3 fasst die Entscheidungen bei Philips Healthcare zusammen.

Zwei Arbeiten versuchen, mehrere Anwendungsbeispiele zu vergleichen. Gurbani et al. diskutieren zwei Modelle der ISE (wobei sie anstelle von „Inner Source“ den Begriff „*Corporate Open Source*“ verwenden) [7]: Die *infrastrukturbasierte ISE* entspricht den in Abb. 2 dargestellten Organisationsentscheidungen. Es wird lediglich eine zentrale Software-Forge zur Verwaltung von Inner-Source-Artefakten bereitgestellt, alles andere wird dezentral und weitgehend unkoordiniert interessierten Parteien überlassen. Wir haben unsere Erfahrungen mit diesem Modell bei SAP detailliert dargestellt [13]. Das Modell der *projektbasierten ISE* entspricht der in Abb. 3 dargestellten Entscheidungswahl, wobei sich Gurbani et al. aber nicht auf einen breiten Ansatz wie bei Philips Healthcare abstützen, sondern von Einzelprojekterfahrungen ausgehend extrapolieren.

Gaughan et al. vergleichen sieben verschiedene Implementierungen von ISE, welche sie über Literaturrecherche und Interviews herausgearbeitet haben [5]. Im Fokus ihrer Arbeit stehen primär die Motivation für ISE sowie die Probleme, welche sich bei deren Implementierung ergeben. Ein Hauptproblem, so Gaughan et al., ist die notwendige Verhaltensumstellung, die für ISE erforderlich ist. Insbesondere stehen häufig sowohl Entwickler als auch Manager der erhöhten Transparenz ihrer Arbeit kritisch gegenüber.

4. GESTALTUNG DER INNER-SOURCE-ENTWICKLUNG

Das Inner-Source-Konzept erweitert die traditionelle Organisation von Wiederverwendung konstruktiv. Die Bestimmung, Entwicklung, Wartung und Betreuung von wiederverwendbaren Komponenten kann wahlweise von einer zentralen Einheit oder auch von jenen Abteilungen übernommen werden, welche die wiederverwendbaren Komponenten nutzen. Wichtig ist, dass im Prinzip alle Aufgaben Profit-Center-übergreifend ausgeführt werden können, lediglich die Verwaltung, d.h. insbes. die Bereitstellung einer Software Forge, ist aus Praktikabilitätsgründen meist bei einer zentralen Einheit angesiedelt. Abb. 4 stellt die resultierenden Optionen mit der Notation aus dem vorigen Abschnitt dar.

4.1 Aufgaben und Rollen in der ISE

Die Grundsatzentscheidung, ISE zu ermöglichen, führt zu weiteren Gestaltungsoptionen und Entscheidungen. Jedes Projekt, das sich mit einer neuen oder existierenden wiederverwendbaren Komponente beschäftigt, muss nun entscheiden, welche der beteiligten Parteien welche Rollen spielen und welche Aktivitäten sie durchführen sollen. Wir vereinfachen die Rollen und Aktivitäten durch Beschränkung auf 3 wesentliche Aufgaben:

- *Anforderungsermittlung*. Die Bestimmung der von einer Komponente umzusetzenden Anforderungen. Es gibt üblicherweise mehrere Personen, welche in Vertretung ihrer Abteilungen Anforderungen einbringen.
- *Projektleitung*. Die Projektkoordination und Leitung der Entwicklung der wiederverwendbaren Komponente. Üblicherweise gibt es genau eine Person in jedem Projekt, welche diese Rolle übernimmt.
- *Entwicklung*. Die technische Umsetzung der priorisierten Anforderungen. Es gibt üblicherweise mehrere Entwickler. Idealerweise stellt jede Abteilung, die sich mit Anforderungen einbringt, hierfür Personal bereit.

Diese Rollen können von Personen aus verschiedenen Abteilungen erfüllt werden. Ein Beispiel ist in Abb. 5 dargestellt. Folgende Abteilungen werden dabei unterschieden:

- *Zentrale Einheit (ZE)*. Die zentrale Einheit für die Organisation der Wiederverwendung kann aktiv alle Rollen und Aufgaben erfüllen. Sie kümmert sich zudem häufig um die Wartung von Komponenten, die nicht mehr aktiv weiterentwickelt werden. In der Anforderungsermittlung vertritt sie meist nicht-funktionale Anforderungen wie die Kompatibilität zu gewünschten Standards und zur Infrastruktur.
- *Anwendungsabteilung (AI-An)*. Eine Anwendungsabteilung definiert als künftiger Nutzer Anforderungen an wiederverwendbare Komponenten und arbeitet im Inner-Source-Szenario auch an deren Entwicklung mit. Ggf. kann eine Anwendungsabteilung auch den Projektleiter stellen. Üblicherweise folgt aus einer Beteiligung an der Anforderungsdefinition auch das Einbringen von Entwicklungsressourcen; dies ist aber nicht zwingend.

Im Beispiel aus Abb. 5 wurde eine Person aus der zentralen Einheit zum Projektleiter erklärt. Diese Position ist anspruchsvoll, weil sie häufig bei Konflikten um die Priorisierung von Anforderungen zwischen den Abteilungen vermitteln muss. In Abschnitt 5 schlagen wir daher vor, eine weitere Rolle, den *Inner-Source-Betreuer*, zu etablieren.

4.2 Organisatorische Probleme und Lösungen

Wiederverwendung ist schwierig. Ein ISE-Ansatz hilft, die organisatorischen Probleme zu adressieren. Wir unterteilen die Probleme in drei Bereiche: Identifikation, Entwicklung (inkl. Wartung) und Nutzung wiederverwendbarer Komponenten.

4.2.1 Probleme bei der Identifikation von wiederverwendbaren Komponenten

Bei der Identifikation wiederverwendbarer Komponenten können u.a. folgende Probleme auftauchen:

- *Unklarer Nutzen einer Komponente jenseits des eigenen Verwendungszwecks*. Entwickelt eine Abteilung einen Kandidaten für eine wiederverwendbare Komponente, so kann diese Abteilung häufig nicht allein entscheiden, ob der Kandidat für andere Abteilungen relevant ist und ob sich weitere Investitionen lohnen.

Ein ISE-Ansatz hilft hier, weil die transparente Entwicklung und Bereitstellung von Softwarekomponenten es anderen Abteilungen ermöglicht, Wiederverwendungskandidaten für eigene Zwecke zu evaluieren und ggf. deren Realisierung zu unterstützen.

- *Unerwünschte zusätzliche Arbeit durch die Beteiligung an der Entwicklung neuer Komponenten*. Erkennt eine Abteilung das Wiederverwendungspotential einer Komponente, so zögert oft-

mals der zuständige Manager, deren Verwendung in anderen Abteilungen zu vermarkten, weil er befürchtet, zusätzlichen Aufwand tragen zu müssen, ohne daraus einen mindestens äquivalenten Nutzen zu ziehen.

ISE verhindert zwar nicht zusätzlichen Aufwand, kann aber durch frühzeitiges Erkennen von Komponentenkandidaten frühzeitige Unterstützung aus anderen Abteilungen kanalisieren. Ist eine Organisation an ISE gewöhnt, reduziert dies den Aufwand für alle.

Zentrale Wiederverwendungsprogramme (auch die eingangs erwähnte extraktive Methode zur Ableitung wiederverwendbarer Komponenten aus existierenden Projekten) lösen das zweite Problem nicht. Ein Manager, der zusätzliche Arbeit befürchtet, wird nicht mit anderen Abteilungen kooperieren wollen.

4.2.2 Probleme bei der Entwicklung von Komponenten

Bei der Erstellung und Weiterentwicklung wiederverwendbarer Komponenten können u.a. folgende Probleme auftreten:

- *Mangelnde Investitionsbereitschaft.* Mancher Projektleiter wird nicht in wiederverwendbare Komponenten investieren, solange er begründete Hoffnung hat, dass diese von anderen Projekten für ihn „kostenfrei“ entwickelt werden. Entsprechend werden manche Komponenten, auch und gerade bei deutlich erkennbarem Wiederverwendungspotential, wegen Ressourcenmangels nie entwickelt werden.

Die traditionelle Antwort auf dieses Problem besteht darin, eine zentrale Einheit mit der Entwicklung der Komponenten zu beauftragen, für welche dann alle zahlen. Diese Maßnahme führt aber zu Problemen bei der Definition und Priorisierung von Anforderungen. Der Inner-Source-Ansatz ermöglicht es Abteilungen, sich mit Ressourcen in ein Projekt einzubringen und so die Priorisierung von Anforderungen zu beeinflussen.

- *Kommunikationsprobleme hinsichtlich der Anforderungen.* Wenn eine Zentralabteilung mit der Entwicklung wiederverwendbarer Komponenten beauftragt wird, ergeben sich häufiger Kommunikationsprobleme bzgl. der Komponentenanforderungen. Dies ist darauf zurückzuführen, dass die organisatorische Trennung zwischen den Entwicklern und den potenziellen Nutzern der Komponente in der Regel auch eine zusätzliche Kommunikationsbarriere darstellt.

Abteilungen, die eine zu entwickelnde Wiederverwendungskomponente nutzen wollen, können in der ISE Ressourcen bereitstellen, die nicht nur Anforderungen kommunizieren, sondern auch an deren Umsetzung direkt mitarbeiten. Somit bringen sich die Stakeholder mit spezifischem Wissen in die Entwicklung einer Komponente ein, was dazu beiträgt, die Kommunikationsprobleme zu reduzieren.

- *Streit um die Priorisierung von Anforderungen.* Gibt es eine zentrale Abteilung für wiederverwendbare Komponenten, so sieht sich diese häufig dem Problem ausgesetzt, konfligierende Anforderungen priorisieren zu müssen. Jede nutzende (und zahlende) Abteilung möchte natürlich ihre Anforderungen in den wiederverwendbaren Komponenten höchstprior realisiert sehen.

Bei der ISE beeinflussen interessierte Abteilungen durch Einbringung von Entwicklungsressourcen, welche Funktionalität als nächstes entwickelt wird. In dem Maße, wie eine Abteilung sich durch Entwicklungsressourcen einbringt, gestaltet sie die Zukunft und die von einer wiederverwendbaren Komponente realisierte Funktionalität. Wesentlich an den Problemlösungen, welche die ISE bietet, ist die Steuerbarkeit der Investition und der daran gekoppelten Investitionsergebnisse für die investierenden Abteilungen.

4.2.3 Probleme bei der Verwendung von Komponenten

Beim Einsatz existierender wiederverwendbarer Komponenten können u.a. folgende Probleme auftreten:

- *Fehlendes Wissen um existierende Komponenten.* Häufig wissen Entwickler nicht um existierende Komponenten und nutzen sie somit auch nicht.

ISE hilft dadurch, dass sie grundsätzlich Öffentlichkeit für die Komponenten schafft, insbesondere wenn alle an einem zentralen Ort auffindbar und verfügbar sind. Müssen Entwickler nach möglicherweise geeigneten Komponenten an unterschiedlichen Stellen suchen, so stellt dies eine erhebliche zusätzliche Hürde dar. Muss nur an einem Ort gesucht werden, erhöht sich dagegen die Wahrscheinlichkeit, dass wiederverwendbare Komponenten auch tatsächlich gesucht und gefunden werden.

- *Fehlende Bereitschaft, bekannte wiederverwendbare Komponenten zu nutzen.* Es gibt verschiedene Gründe, warum bekannte wiederverwendbare Komponenten nicht verwendet werden: Ein technischer Grund kann z.B. die zu große Komplexität einer Komponente sein, die u.U. als mittelbare Folge ihrer sehr allgemeinen Verwendbarkeit entsteht. Ein weiterer Grund kann z.B. in der bei Entwicklern und Managern gleichermaßen verbreiteten Angst vor Kontrollverlust liegen.

Gegen psychologische Probleme, insb. gegen die Angst vor Kontrollverlust, hilft ISE insofern, als sie die Komponenten firmenintern öffentlich zugänglich macht und allen Abteilungen die Chance gibt, sich durch Entwicklungsressourcen einzukaufen. Es ist unsere Erfahrung, z.B. aus der bei SAP durchgeführten Fallstudie [13], dass psychologische Ängste abgebaut werden können, sobald Manager einmal den Schritt in die firmeninterne Öffentlichkeit gewagt haben und mit ihr vertraut sind.

5. DIE INNER-SOURCE-ORGANISATION

Die uns bekannte Forschung zu Inner-Source-Ansätzen leitet diese aus der öffentlichen Open-Source-Entwicklung ab. Praktikern der Inner-Source-Entwicklung ist dies vermutlich gleichgültig: Sie greifen zu Profit-Center-übergreifender Organisation von Wiederverwendung, um ihre Probleme zu lösen und die relevanten Komponenten umgesetzt zu bekommen. Gleichwohl sind Open Source und die in Abschnitt 3 und 4 dargestellten Inner-Source-Modelle für unsere Betrachtungen interessant, da sie helfen, eine mögliche Zukunft vorherzusagen. In diesem Abschnitt leiten wir daher aus der Evolution des Open Source eine Prognose für die mögliche Zukunft der Inner-Source-Entwicklung ab.

5.1 Open-Source-Vereinigungen

Nach initialen Jahren der Unsicherheit wurden im Bereich der Open-Source-Entwicklung die sogenannten *Open-Source-Vereinigungen* („foundations“) geschaffen, die der rechtlichen und organisatorischen Grauzone um Open-Source-Projekte ein Ende bereiteten. Ein Beispiel ist die *Apache Software Foundation* (ASF, siehe <http://www.apache.org>). In einem anderen Artikel schildern wir, warum diese Vereinigungen ökonomisch nachhaltig sind [12]. Die ASF stellt eine organisatorische und technische Plattform für Open-Source-Projekte zur Verfügung. Viele rechtliche Fragen sind für Inner Source nicht relevant, da bei ISE das geistige Eigentum an allen entstehenden Artefakten von Anfang an dem Unternehmen gehört. Wir fokussieren unsere Betrachtung deswegen primär auf den Aspekt der Projektorganisation.

Die ASF selbst entwickelt keine Software, begleitet aber den Prozess der Gemeinschaftsbildung um Open-Source-Projekte auf Basis von jahrelang gewachsenen Erfolgsmethoden („best practices“). Damit ein Open-Source-Projekt zu einem ASF-Projekt werden kann, muss es zunächst der ASF vorgeschlagen werden. Erscheint es den ASF-Mitgliedern sinnvoll, wird es in den sogenannten *Projektinkubator* übernommen. Das Ziel des Inkubators ist es, das Projekt zu voller Größe „auszubrüten“, wodurch es dann zu einem vollwertigen Apache-Projekt wird. Zu den wesentlichen Kriterien für die Reife eines Apache-Projekts gehört, dass es eine florierende Gemeinschaft aktiv beitragender Unternehmen oder natürlicher Personen gibt und dass die Menge der an der Entwicklung beteiligten Parteien möglichst vielfältig ist.

Erfolgreiche Projekte hängen also vom Geschick der Projektinitiatoren ab. Diese müssen weitere Parteien an Bord bringen, um jenen Grad an Diversität der Gemeinschaft zu erreichen, den die ASF auf Dauer verlangt. Die ASF selbst entwickelt keine Software und mischt sich in technische Belange nicht ein. Sie stellt aber jedem Projekt einen Betreuer zur Seite. Dieser ist kein Projektleiter, sondern vielmehr ein erfahrener Vermittler und Berater, der das Projekt mit konkreten Ratschlägen und methodischer wie sozialer Kompetenz unterstützt. Anforderungsermittlung, Projektleitung und Entwicklung

werden ausschließlich von den sich einbringenden Parteien wahrgenommen. Die Betreuerrolle wird dagegen stets mit Mitgliedern der ASF besetzt.

5.2 Die Inner-Source-Organisation

Aus der geschilderten Evolution im Bereich der Open-Source-Entwicklung leiten wir nun die Etablierung einer neuen firmeninternen Organisation für die ISE ab: Die Inner-Source-Organisation. Diese muss keine eigenständige Organisationseinheit sein, sondern kann als virtuelle Organisation oder auch nur als Arbeitskreis erfahrener Betreuer organisiert werden. Aufgabe der Mitglieder einer solchen Organisation ist es, Inner-Source-Projekte zu betreuen, aber nicht selbst durchzuführen. Somit kommen auf diese Organisation die folgenden Aufgaben zu:

1. Definition eines Prozesses zur Erstellung und Qualifizierung von Vorschlägen für Inner-Source-Komponenten,
2. Auswahl erfolgversprechender Vorschläge für Inner-Source-Komponenten,
3. Definition eines Lebenszyklusmodells für Inner-Source-Komponenten,
4. Personalrekrutierung für die Entwicklung von Inner-Source-Komponenten,
5. Betreuung der Entwicklung von Inner-Source-Komponenten,
6. Rekrutierung von Personen für die Inner-Source-Organisation,
7. evolutionäre Weiterentwicklung der Inner-Source-Organisation,
8. firmeninterne Verbreitung und Bewerbung der Inner-Source-Idee,
9. Herstellung ausreichender firmeninterner Sichtbarkeit für existierende bzw. entstehende Inner-Source-Komponenten,
10. Bereitstellung einer technischen Plattform zur Unterstützung der Aufgaben 1. bis 9.

In Gesprächen mit Unternehmen, die an ISE interessiert sind oder aktiv ISE betreiben, haben wir den Eindruck gewonnen, dass es im Regelfall eine ausreichende Anzahl von Personen mit entsprechender Erfahrung und Seniorität gibt, die bereit sind, sich für die o.g. Aufgaben zu engagieren, und die den Gründungskern einer Inner-Source-Organisation bilden können. Diesem Personenkreis obliegt es dann, einen aktiven Prozess der Auswahl und Initialisierung von ISE-Projekten zu betreiben (1. bis 3.), deren Durchführung sie anschließend betreuen (5.). Insbesondere wird die Inner-Source-Organisation auch Überzeugungsarbeit bei den Managern von potenziell beitragenden Abteilungen leisten müssen, damit diese die erforderlichen Ressourcen in das Projekt einbringen (4.). Gleichzeitig sind Projektleiter und Mitarbeiter erfolgreicher ISE-Projekte natürlich auch potenzielle Kandidaten für eine künftige Mitgliedschaft in der Inner-Source-Organisation (6. und 7.).

Wesentlich für den Erfolg einer Inner-Source-Organisation sind die sozialen Vermittlungs- und Betreuungsfähigkeiten ihrer Mitglieder. Diese müssen in der Lage sein, zusammen mit dem Projektleiter eines Inner-Source-Projektes bei den Abteilungsleitern ein aufgeklärtes Eigeninteresse zu mobilisieren, das letztlich dazu führt, dass die Abteilungen sich bei der Entwicklung einer wiederverwendbaren Komponente freiwillig mit der Definition von Anforderungen und der Bereitstellung von Ressourcen einbringen.

6. SCHLUSSFOLGERUNGEN UND ABSCHLUSS

In diesem Artikel erörtern wir das Konzept der Inner-Source-Entwicklung, bei der wiederverwendbare Softwarekomponenten in firmeninterner Profit-Center-übergreifender Gemeinschaftsarbeit erstellt werden. Wir berichten über unsere sowie die in der Literatur zu findenden Erfahrungen mit Inner-Source-Ansätzen. In Analogie zur Evolution im Bereich der Open-Source-Entwicklung prognostizieren wir die Etablierung von Inner-Source-Organisationen in großen Unternehmen, die das firmenintern vorhandene Potenzial an Optimierungsmöglichkeiten durch Wiederverwendung von Softwarekomponenten besser als bisher nutzen möchten. Unsere Beratungserfahrung zeigt, dass bereits existierende In-

ner-Source-Projekte in den meisten Unternehmen unterschätzt werden und daher keine organisatorische Unterstützung finden. Geschähe dies, so könnten Softwareentwicklungsorganisationen ihre Produktivität deutlich verbessern.

7. LITERATURVERZEICHNIS

- [1] Ars Technica. „Novell Hack Week: An Experiment in Innovation“. Siehe <http://arstechnica.com/business/news/2007/07/novell-hack-week-an-experiment-in-innovation.ars>
- [2] Dinkelacker, J.; Garg, P.K.; Miller, R. „Progressive Open Source“. In Proceedings of the 24 International Conference on Software Engineering (ICSE 2002). IEEE Press, 2002.
- [3] Fox, S. Personal Communication, 2007.
- [4] Frakes, W.B.; Kyo Kang; „Software Reuse Research: Status and Future“. IEEE Transactions on Software Engineering, Volume 31, Issue 7.
- [5] Gaughan, G.; Fitzgerald, B.; Shaikh, M. „An Examination of the Use of Open Source Software Processes as a Global Software Development Solution for Commercial Software Engineering“. In Proceedings of the 35th Euromicro Conference.
- [6] Google Inc. „20% Time“. Siehe www.google.com/jobs/lifeatgoogle/englife/index.html
- [7] Gurbani, V.K.; Garvert, A.; Herbsleb, J.D. „Managing a Corporate Open Source Software Asset“. Communications of the ACM, Volume 53, Issue 2.
- [8] Krueger, C.W. „Easing the Transition to Software Mass Customization“. In Software Product-Family Engineering. Springer, 2002.
- [9] Melian C. et al. „Building Networks of Software Communities in a Large Corporation“. Technical Report. Hewlett Packard, 2002.
- [10] Microsoft Corp. „Open Source at Microsoft: Codebox, Bringing the Open Source Approach In-House“. Technical Report. Microsoft, 2007.
- [11] Mohagheghi P.; Conradi, R. „Quality, Productivity and Economic Benefits of Software Reuse: A Review of Industrial Studies.“ Empirical Software Engineering (2007) 12.
- [12] Riehle, D. „The Economic Case for Open Source Foundations“. IEEE Computer Volume 43, Issue 1.
- [13] Riehle, D.; Ellenberger, J.; Menahem, T.; Mikhailovski, B.; Natchetoi, Y.; Naveh, B.; Odenwald, T.; „Open Collaboration within Corporations Using Software Forges“. IEEE Software, Volume 26 Issue 2.
- [14] Wesselius, J. „The Bazaar Inside the Cathedral: Business Models for Internal Markets.“ IEEE Software, Volume 25, Issue 3.