

Analyse der IWSECO- Workshops

TRENDS UND TOPICS

MICHAEL GALETZKA, WALDEMAR FLAT

1

Herzlich Willkommen zur Seminar-Präsentation zu den IWSECO-Workshops. IWSECO bedeutet „International Workshop on Software Ecosystems“. Im Folgenden werden die Trends und Topics dieser Workshops vorgestellt.

Agenda

- I. Statistische Analyse der Paper
- II. Übersicht der Paper
- III. Interessante Paper
- IV. Hauptthemen und Trends
- V. Offenen Fragen und Herausforderungen
- VI. Diskussion

Hier der Ablauf der Präsentation.

Als erstes werden einige Zahlen und Statistiken zu den Workshop-Papern genannt.

Als nächstes wird eine Übersicht der Paper gegeben. Dabei wird jedes Paper kurz zusammengefasst.

Anschließend werden einige interessante Paper näher vorgestellt.

Als viertes werden die Haupt-Themenblöcke des Workshops präsentiert.

Der fünfte Punkt soll zeigen, wo der Workshop sich in Zukunft hin entwickeln muss.

Als letztes wird mit einer persönlichen Meinung eine mögliche Diskussion eröffnet.

Übersicht der bisherigen Workshops

Jahr	Ort	Veranstaltet innerhalb von
2009	Falls Church, Virginia, USA	11th International Conference on Software Reuse
2010	Kopenhagen, Dänemark	4th European Conference on Software Architecture
2011	Brüssel, Belgien	Second International Conference on Software Business
2012	Cambridge, Massachusetts, USA	Fourth International Conference on Software Business

Hier wird eine Übersicht über die während der Präsentation vorgestellten Workshops dargestellt.

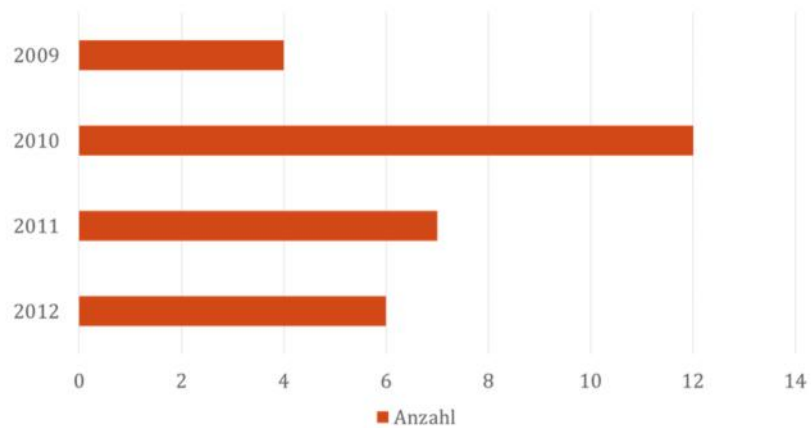
Die Workshops sind immer Teil größerer Konferenzen. Ab dem Jahr 2011 (auch im kommenden Jahr 2013) sind sie Teil der „International Conference on Software Business“.

Dass diese größeren Konferenzen die Themen der Workshops natürlich beeinflussen zeigt sich später recht deutlich.

Statistische Analyse der Paper

Im folgenden werden die Workshops als Ganzes aus einem statistischen Blickwinkel betrachtet.

Anzahl der Paper pro Jahr



MICHAEL GALETZKA, WALDEMAR FLAT

5

Hier sieht man die Anzahl der veröffentlichten Workshop-paper pro Jahr. Auffällig ist insbesondere der starke Ausschlag im Jahr 2010 in dem doppelt so viele Paper veröffentlicht wurden wie in den übrigen Jahren.

Übersicht der Paper

Auf den folgenden Folien werden die einzelnen Paper der Workshops kurz aufgelistet. Dabei wird bewusst nicht konkret auf einzelne Paper eingegangen, da dies viel zu viel Zeit beanspruchen würde. Die Folien sind nur dafür gedacht, dem Betrachter eine Übersicht über die Paper zu geben. Deshalb sind die wichtigsten Schlüsselwörter in den Titeln der Paper fett markiert.

Übersicht der Paper 2009

- **Building** Software Ecosystems from a **Reuse Perspective** (Talk, kein Paper)
- **Business Network Management** as a Survival Strategy - A Tale of Two Software Ecosystems
- Specification and Analysis of **Requirements Negotiation Strategy** in Software Ecosystems
- The Role of **Software Licenses** in **Open Architecture** Ecosystems

Da es der erste Workshop war, war auch die Zahl der Paper überschaubar. Insgesamt gab es drei Paper und einen Talk.

Es fällt auf, dass die Themen sich am Thema der „Conference on Software Reuse“ orientieren und sich zudem mit Problemen beschäftigen, die nicht notwendigerweise etwas mit SECOs zu tun haben (z.B. „Software Licenses“ oder „Requirements Negotiation“). Das hängt vermutlich damit zusammen, dass sich zum damaligen Zeitpunkt noch nicht genügend Forscher mit dem Thema SECO direkt auseinander gesetzt haben.

Übersicht der Paper 2010

- A method for **analyzing** software **product line** ecosystems
- A **three-dimensional view** of software ecosystems
- Architecting in software ecosystems - **interface translucence** as an enabler for scalable collaboration
- **Architecture challenges** for software ecosystems
- Evaluating architectural **openness in mobile software** platforms
- **Industry taxonomy** engineering - the case of the European software ecosystem

Im Vergleich zu 2009 sind 2010 nicht nur viel mehr Paper eingereicht worden, es beschäftigen sich auch mehrere Paper direkt mit SECOs (z.B. „A **three-dimensional view** of software ecosystems“).

Auch hier kann man gut erkennen, wie das Motto der veranstaltenden Konferenz (“Conderence on Software Architecture”) sich auf die eingereichten Paper auswirkt.

Übersicht der Paper 2010

- Integrating **recommender information** in social ecosystems decisions
- **Knowledge management** in software ecosystems - **software artefacts** as first-class citizens
- On the role of software **process modeling** in software ecosystem design
- Revisiting the concept of **components** in software engineering from a software ecosystem perspective
- Software ecosystems vs. **natural ecosystems** - learning from the ingenious mind of **nature**
- Software ecosystems - a software ecosystem **strategy assessment model**

Ein etwas kurios anmutendes Paper, das aus den anderen hervorsticht, ist „Software ecosystems vs. **natural ecosystems** - learning from the ingenious mind of **nature**“, das später noch etwas genauer vorgestellt wird.

Übersicht der Paper 2011

- A Proposal for Software Ecosystems **Engineering**
- A **Survey of Associate Models** used within Large Software Ecosystems
- A **Systematic Mapping Study** on Software Ecosystems
- Analysing the **evolution of social aspects** of open source software ecosystems
- **Cloppeness** of Systems - The **Interwoven** Nature of Ecosystems
- Hybrid **Revenue Models** of Software Companies and their Relationship to Hybrid **Business Models**
- Understanding Software Ecosystems - A **Strategic Modeling** Approach

Obwohl weniger Paper als noch 2010 eingereicht wurden, zeigt sich, dass diese von den Themen viel modell- und systemlastiger geworden sind (z.B. „Understanding Software Ecosystems - A **Strategic Modeling** Approach“).

Dies könnte damit zusammen hängen, dass die Forschung auf dem SECO-Gebiet sich von konkreten Problem immer mehr den theoretischen Grundlagen zuwendet.

Übersicht der Paper 2012

- App Store, Marketplace, Play! An Analysis of **Multi-Homing in Mobile** Software Ecosystems
- Defining Software Ecosystems - A Survey of **Software Platforms** and **Business Network Governance**
- EcoSysNetworks - A Method for **Visualizing** Software Ecosystems
- Leveraging **Open Source** Licenses and Open Source Communities in Hybrid **Commercial Open Source Business Models**
- **Mashups** - Software Ecosystems for the **Web** Era
- **Theoretical foundations** of software ecosystems

Bei den Papern von 2012 zeigt sich einerseits sehr stark der Charakter der veranstaltenden Konferenz („Conference on Software Business“), da beispielsweise der Business-Aspekt sehr stark vertreten ist. Zum anderen wird aber mit Papern wie „**Theoretical foundations** of software ecosystems“ auch der Theorie hinter den SECOs weiter große Bedeutung beigemessen.

Interessante Paper

Auf den folgenden Folien werden die einzelnen Paper der Workshops kurz vorgestellt.

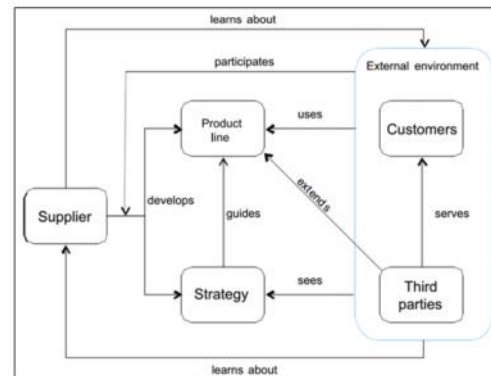
Theoretical foundations of software ecosystems

- **Eckdaten**

- Jahr: 2012
- Autoren: Geir K. Hanssen und Tore Dybå
- 7 Seiten (+5 Seiten Referenzen)

- **Worum geht es?**

- Review verschiedener Publikationen zur Theorie von SECOs
- Stellen ein eigenes theoretisches Framework für SECOs vor
- Ihr eigenes Framework soll als Basis für weitere Forschungen im SECO Bereich dienen



Quelle: Geir K. Hanssen, A longitudinal case study of an emerging software ecosystem: Implications for practice and theory

Wieso interessant?

Die Autoren des Papers versuchen, den bisherigen Stand der theoretischen Grundlagen von SECOs zu erfassen und schlagen ihr eigenes Modell als sinnvolle Ergänzung hierzu vor. Insbesondere das ausführliche Review der vorhandenen Paper vermittelt einen guten Einblick in die aktuelle Forschungslage.

Fazit

Das Paper stellt das bis dato ausführlichste Review von Papern zur Theorie von SECOs dar. Die Autoren hoffen, dass ihr vorgestelltes Modell als Ausgangspunkt für die weitere Forschung genutzt wird.

Paper

IWSECO 2012: Theoretical foundations of software ecosystems

Software ecosystems vs. natural ecosystems

- Fördert das allgemeine SECO Verständnis
- Nennt Herausforderungen auf dem Weg zum gesunden SECO

Beispiele für Gemeinsamkeiten zwischen den beiden Ökosystemen:

Aspekt	Natürliches Ökosystem	Software Ökosystem
Ressourcen	Nahrung, Nährstoffe, Wasser	Plattform, Geld, Entwickler usw.
Beziehungen	Wettbewerb, Symbiose	Wettbewerb, Zusammenarbeit
Energie-Transfer	Nahrungsketten	Weitergabe von Wissen
Prozesse	Lebenszyklen	Marktzyklen, techn. Fortschritte
Anpassung	An Kontextänderung z. B. nach Sturm	An Marktsituation

MICHAEL GALETZKA, WALDEMAR FLAT

18

Wieso interessant?

Dieses Paper hilft zu verstehen, wie der Begriff „Software Ecosystem“ entstanden ist und welche Gemeinsamkeiten zwischen diesen Systemen existieren. Wenn man versteht wie die Natur mit Problemen in natürlichen Ökosystemen umgeht, kann man auch in der Softwareentwicklung von diesen Lösungsansätzen profitieren. Z. B. dient die Evolutionstheorie als Vorbild für genetische Algorithmen, wo die natürliche Selektion auf den Daten angewendet wird.

Inhalt:

Diese Arbeit dient als Agenda für weitere Forschungen.

Es werden die existierenden regulatorischen Faktoren und Mechanismen in der Natur diskutiert und daraus die wichtigsten Herausforderungen abgeleitet, mit denen ein gesundes SECO im Laufe der Zeit konfrontiert wird.

Dabei sind 2 Herausforderungen besonders wichtig: 1) das Ressourcenmanagement und 2) die Nachhaltigkeit

Ressourcenmanagement

Wie im natürlichen Ökosystem so auch im SECO sind die Ressourcen knapp und müssen überlegt eingesetzt werden.

Im natürlichen Ökosystem wird die Energie z. B. durch Fotosynthese erzeugt. Diese Energie wird übertragen und in andere Formen transformiert, die anschließend durch die verschiedenen Nahrungsketten wandert. Die Nährstoffe werden durch Kreisläufe recycelt.

Im SECO braucht man auch eine kontinuierliche Zufuhr von Energie z. B. in Form von neuen Entwicklungen. Anbieter von Plattformen investieren große Mengen an Ressourcen (wie Geld, Zeit, Personal) um ein SECO in Gang zu bringen. Auch hier gibt es Kreisläufe, die das Plattform-Wissen über die Entwickler zu den Produkt-Anwendern transportieren.

Nachhaltigkeit

Die Nachhaltigkeit spielt eine entscheidende Rolle bei der Widerstandsfähigkeit von Ökosystemen, d. h. die Fähigkeit nach Störereignissen sich zu reorganisieren und wieder auf die Beine zu kommen. Ökosysteme scheinen sehr robust zu sein, wenn es mehrere Arten gibt, die die gleiche Funktion erfüllen und jede Art unterschiedlich auf Störungen reagiert. Dadurch können diese Arten bei Störungen einander kompensieren.

In SECO kann es z. B. durch die Unterstützung von verschiedenen Programmiersprachen, Plattformen und Geräten erreicht werden. Der Verlust eines Abnehmers kann besser kompensiert werden, wenn eine solche Vielfalt vorhanden ist.

Fazit

Wenn man die genannten Herausforderungen gemeistert hat, kommt man zum gesunden SECO.

Paper

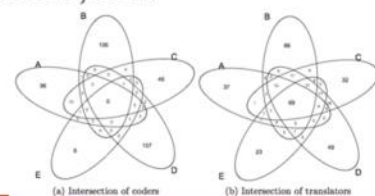
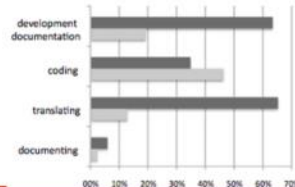
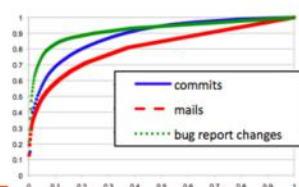
IWSECO 2010: Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature

Analysing the evolution of social aspects of open source software ecosystems

- Empirische Studie an Projekten aus dem GNOME OpenSource SECO
- Ziel: Verstehen wie die Kommunikation und Interaktion den Software-Evolutionsprozess beeinflusst

Folgende 3 Messungen werden durchgeführt:

1. Wie sind die Aktivitäten unter den Projektbeteiligten verteilt?
2. Wie ist die Aufgabenverteilung unter den Projektbeteiligten?
3. Wie sind die Aktivitäten der Beteiligten in unterschiedlichen Projekten?



MICHAEL GALETZKA, WALDEMAR FLAT

19

Wieso interessant?

Es ist das einzige Paper, wo der soziale Aspekt eines SECOs im Vordergrund steht. Die Mehrheit der Paper beschäftigen sich mit den technischen und wirtschaftlichen Eigenschaften eines SECOs.

Da aber in SECO der Mensch eine entscheidende Rolle spielt, darf der soziale Faktor nicht vernachlässigt werden.

Inhalt

In einer empirischen Studie wird aus unterschiedlichen GNOME-Repositories wie Versionsverwaltung, Mailinglisten und Bugtracker Informationen extrahiert, um die sozialen Aspekte der Softwareentwicklungsgemeinde zu studieren. D. h. man will verstehen wie die Entwickler arbeiten, kommunizieren und Informationen austauschen.

Die zugrunde liegende Hypothese ist, dass die sozialen Aspekte wesentlich die Art und Weise beeinflussen, wie ein Softwareprojekt im Laufe der Zeit sich entwickelt. Z. B. wenn eine Schlüsselperson ein Projekt verlässt oder eine Software-Bibliothek ausgetauscht wird, kann es eine erhebliche Auswirkung auf das Projekt haben. Ein besseres Verständnis könnte es ermöglichen Prognosemodelle, Richtlinien und Best Practices zu definieren, die den Gemeinden erlauben ihre derzeitigen Praktiken

zu verbessern und Tools zum Kontrollieren und Verbessern des Arbeitsprozesses zu erstellen.

Vorgehen

Die Studie wird nach der GQM-Methode durchgeführt. Zuerst werden spezifische Fragen mit den dazugehörigen Hypothesen definiert. Danach wählt man die passenden Metriken, um die Hypothesen zu überprüfen.

Studie

Im vorliegenden Paper werden 3 Messungen beschrieben.

Man muss an diesem Punkt auch sagen, dass die gesamte Studie noch nicht abgeschlossen ist. Es wird hier lediglich ein vorläufiges Ergebnis präsentiert. Trotzdem sind hier schon interessante Muster zu beobachten.

1. Messung

Hier will man verstehen wie die Aktivitäten unter den Projektbeteiligten verteilt sind. Dabei sind die Aktivitäten in drei Kategorien unterteilt Commits, E-Mail Senden und Bug-Report Ändern.

Das Ergebnis dieser Frage zeigt 1. Diagramm. Hier sind die Verteilungsfunktionen der drei Aktivitäten zu sehen und man kann sehen, dass die meiste Arbeit von 20 % der Personen erledigt wird.

2. Messung

Hier werden die Aktivitätsmuster von Personen innerhalb eines Repositories analysiert. Dabei werden bestimmte Datentypen gewissen Aktivitätstypen zugeordnet. D. h. wenn eine Person eine *.cpp oder eine *.java Datei ändert, dann führt sie eine Programmierstätigkeit aus.

Das Ergebnis ist im 2. Diagramm zu sehen. Hier sieht man, dass 35% der Personen Code schreiben, während die Gesamtanzahl der Commits für diesen Aktivitätstyp bei 46% liegt. Man kann daraus schließen, dass die Code-Schreiber die aktivsten Leute in der Gemeinde sind.

3. Messung

Bei der letzten Messung sucht man die Zusammenhänge zwischen den Projekten. Dabei sammelt die Informationen über jede Person in mehreren Projekten. Das Ergebnis im Diagramm 3 zeigt die Aktivitäten von Programmierern und Übersetzern in mehreren Projekten. Die Schnittmengen zeigen, dass die Programmierer scheinen bei einem Projekt zu bleiben. Während die Übersetzer in mehreren Projekten tätig sind. D. h. die Annahme aus der letzten Messung kann widerlegt werden.

Fazit

Wie man gesehen hat, zeigen bereits die vorläufigen Ergebnisse interessante Muster und es macht Sinn weiter zu machen.

Paper

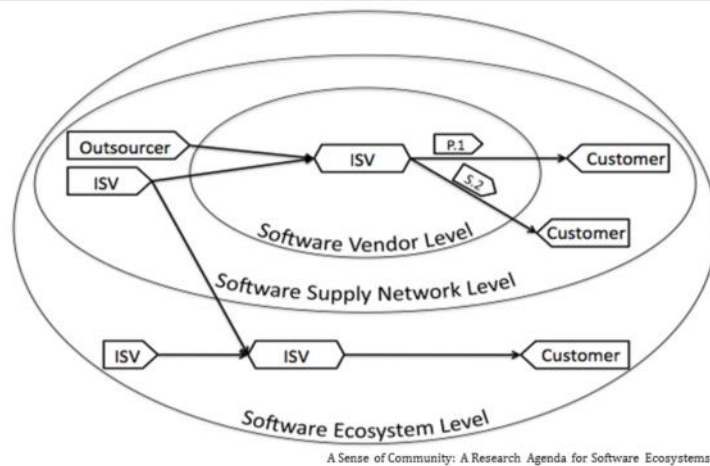
IWSECO 2011: Analysing the evolution of social aspects of open source software ecosystems

Hauptthemen und Trends

Im Folgenden werden die Hauptthemen der aus den IWSECO-Workshops vorgestellt. Zum Schluss werden die sich daraus ergebenden Trends angesprochen.

Architektur

- Software Produktlinien
- Herausforderungen
- 3 Ebenen Sichtweise
- Mashups



Wie jedes System hat auch ein SECO eine Architektur.

Im Workshop werden Themen angesprochen wie ein SECO entstehen kann und wie es aufgebaut ist.

Software Produktlinien

Software-Produktlinien-Unternehmen können große und komplexe Gebilde sein. Z. B. Nokia. Ziel dabei ist ein Softwareprodukt auf Basis einer gemeinsamen Plattform mit mehreren individuellen Ausprägungen zu erstellen. Erreicht wird es durch bekannte Wiederverwendungsansätze aus der Softwareentwicklung wie Komponenten oder Services.

Um Produkte auch für eine Domain anbieten zu können, wo das Unternehmen wenig Erfahrung hat, wird ein SPL-Unternehmen gezwungen externe Entwickler zu beschäftigen. Durch diesen Schritt entsteht ein SECO, da jetzt mehrere Beteiligte (Zulieferer, Kunden, Partner) innerhalb einer gemeinsamen Plattform zusammen entwickeln.

Eine gute Architektur begünstigt, einen schnellen und billigen Fluss von Informationen und Artefakten. Es ist auch zu berücksichtigen, dass während der Entwicklung die Anforderungen sich ändern können, wodurch man gezwungen wird den Zulieferer für eine bestimmte Komponente zu wechseln.

Architektur Herausforderungen

Ein SECO muss auch attraktiv für externe Entwickler sein. Hier gibt es folgende Herausforderungen an die Architektur:

1. SECO muss ein stabiles Interface zwischen externen Entwicklern und der Plattform anbieten.
2. SECO muss für ein angemessenes Maß an Integration zwischen der Plattform und extern entwickelter Lösung sorgen.
3. SECO muss die Wahrscheinlichkeit reduzieren, dass ein defekter oder schädlicher Code produktiv genommen wird, der ggf. das gesamte System beeinträchtigt.

Drei Ebenen

Ein SECO hat drei Ebenen:

1. *Organisationsebene*: Der Hersteller stellt die Software her und gibt die Funktionalität innerhalb des SECO frei. Das Ziel der Organisation ist es den Gewinn durch den Verkauf von Produkten bzw. Services zu maximieren. Die Leistung und die Evolution vom Unternehmen sind abhängig vom SECO und sollten genau analysiert werden, um den maximalen Ertrag zu erhalten. Da das Unternehmen offen zu seinen Partnern ist, muss es auch das Wissen, die Forschung, den Markt und die Technologien mit Ihnen teilen.
2. *Zulieferer-Ebene*: Auf dieser Ebene befinden sich alle Stakeholder (wie Zulieferer, Kunden, Verteiler und externe Entwickler) des Unternehmens und die Beziehungen untereinander. Ebenso auch die Eigenschaften, die die Gesundheit und die Stabilität vom SECO betreffen. Wie z. B. Größe, Typ, Rollen, Vernetzung usw.
3. *SECO-Ebene*: Die letzte Ebene ist das SECO selbst und ihre Beziehungen.

Mashups

Webbasierte Software und Services sind in aller Welt erreichbar. Diese können ohne Installation verwendet und aktualisiert werden. Die Services können auch im anderen Kontext wieder verwendet werden. Die Anzahl der Services und der Geräte ist explodiert. Es ist schwierig geworden an die relevanten Daten zu kommen. Mashups sind eine neue Generation von Anwendungen, die dabei helfen die Daten zu aggregieren und den Inhalt weltweit zu teilen. Dadurch werden implizit neue SECO's erzeugen, da die Mashups die Service-Anbieter miteinander verbinden.

Paper

IWSECO 2009: Building Software Ecosystems from a Reuse Perspective

IWSECO 2010: A method for analyzing software product line ecosystems

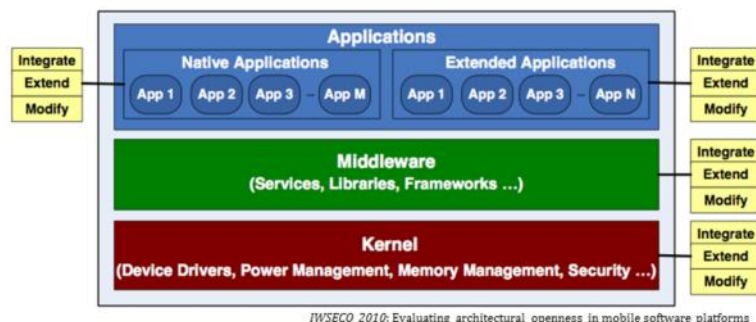
IWSECO 2010: Architecture challenges for software ecosystems

IWSECO 2011: A Proposal for Software Ecosystems Engineering

IWSECO 2012: Mashups - Software Ecosystems for the Web Era

Offenheit

- Open Source vs. proprietär
- Lichtdurchlässigkeit der Schnittstellen
- clopenness



Offenheit ist ein zwingende Eigenschaft einer Plattform auf dem Weg zum SECO. Im Workshop werden die Vorteile und Nachteile davon erläutert.

Open Source oder proprietär

Offenheit wird von Plattform-Anbieter festgelegt und dadurch auch die Frage beantwortet, wie die Nutzer mit der Plattform interagieren können und wie die Nutzer die Plattform-Komponenten anzeigen, erweitern oder ändern können. Die Offenheit hängt auch von unterschiedlichen technischen und wirtschaftlichen Aspekten wie Architektur, Marketing, Politik usw. ab. D. h. die Offenheit entscheidet ob und wie die Dritt-Entwickler die Plattform akzeptieren. Die Akzeptanz ist essenziell für das Wachstum eines SECO's und ist nachweislich auch eine erfolgreiche Strategie. Es stellt sich die Frage welche Strategie erfolgreicher ist OpenSource oder proprietär? Welche Strategie hat mehr Aufmerksamkeit von den Entwicklern erhalten? Welche führt zu mehr innovativen Anwendungen? Man kann sagen, dass OpenSource aufgrund der Kreativität und Innovationen erfolgreicher ist. Auf der anderen Seite kann man glauben, dass die proprietäre Strategie erfolgreicher ist, weil es aufgrund der strengen Kontrollen zu besseren Produkten führt. Die Realität zeigt, dass es keine binäre Entscheidung ist. In der Regel stellt sich die Frage „wie offen“ soll die Plattform sein.

Lichtdurchlässigkeit

SECO's haben zwei Grundprinzipien:

1. Transparenz, die man aus der OpenSource-Entwicklung kennt: Jede Art an Informationen in Bezug auf Entwicklung ist frei verfügbar. Wie z. B. Design-Dokumente, Quellcode, Entwicklungsaufgaben, Defects und Kommunikation zwischen den Mitgliedern.
2. Modularer Systemaufbau: Zerlegung des Systems in überschaubare Teile und Minimierung der Kopplung zwischen den Systembestandteilen.

Diese Prinzipien haben aber auch ihre Herausforderungen.

Transparenz könnte zu einer Informationsflut führen und folglich die zentrale Rolle als Koordinierungsmechanismus reduzieren. Darüber hinaus könnte die Transparenz auch nicht wünschenswert sein, wie im Falle des geistigen Eigentums oder Geschäftsgeheimnisses, Sicherheitsbedenken oder Haftung. Modulares Design beruht auf der Annahme, dass die System-Struktur definiert werden kann und danach fest bleibt. Das ist aber der Widerspruch zu dem Evolutionsgedanken.

Um diesen Herausforderungen gerecht zu werden, wird hier das Konzept der Lichtdurchlässigkeit der Schnittstellen vorgestellt. Durch dieses Konzept versucht man die technischen und organisatorischen Dimensionen der Software-Entwicklung zu überbrücken.

Dadurch wird die Transparenz in dem Maße eingeschränkt, damit eine Informationsüberfrachtung begrenzt wird und sensible Informationen geschützt werden. Darüber hinaus fördert dieses Konzept die dynamische Natur der Software-Entwicklung explizit in den Schnittstellen zwischen den Komponenten.

Clopenness

Offenheit wird oft als binäres Konzept wahrgenommen. Wenn ein Unternehmen geschlossen ist, wird es damit assoziiert, dass es diktatorisch, undemokratisch und undurchsichtig ist. Während offene Unternehmen positiv betrachtet werden, das sie transparent und gut sind. Diese binäre Vorstellung ist aber schädlich für die Software-Industrie, da man hier falsche Entscheidungen treffen kann. Aus diesem Grund wird hier das clopenness-Modell vorgestellt, das einen Einblick in die Offenheit einer Organisation gewährt, um dadurch eine bessere Entscheidungen treffen zu können. D. h. dadurch kann man feststellen ob ein Unternehmen wirklich offen oder geschlossen ist.

Paper

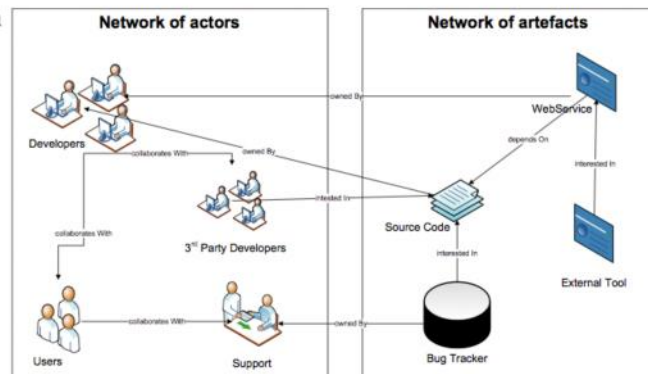
IWSECO 2010: Architecting in software ecosystems - interface translucence as an enabler for scalable collaboration

IWSECO 2010: Evaluating architectural openness in mobile software platforms

IWSECO 2011: Clopenness of Systems - The Interwoven Nature of Ecosystems

Kommunikation

- Kommunikation von Anforderungen
- Kommunikation mit Stakeholdern
- Wissensverwaltung



IWSECO 2010: Knowledge management in software ecosystems - software artefacts as first-class citizens

Bei diesem Punkt geht es um den Wissensaustausch und Beziehungen innerhalb SECO.

Kommunikation von Anforderungen

In ein SECO sind eine große Anzahl an Stakeholdern involviert und diese müssen bei der Anforderungsdefinition der Produkte und Plattform berücksichtigt werden. Zur Zeit werden unterschiedliche Spezifikationen und Dokumente zum Verhandeln und Abstimmen von Anforderungen unter den Stakeholder benutzt. Um diese Komplexität in den Griff zu bekommen, sollte ein Stakeholder-Netzwerk aufgebaut werden, damit die unabhängigen Stakeholder ihre Produkthanforderungen verhandeln können. Solche Netzwerke könnten auch zum Identifizieren und Beseitigen von Kommunikationsproblemen verwendet werden und dadurch die Prozesse verbessern. Ein weiteres Problem bei den heutigen Ansätzen ist, dass sie die Beziehungen zwischen den Stakeholdern nicht berücksichtigen und sie nehmen an, dass alle Stakeholder gleich sind.

Wissensverwaltung

Für den Umgang mit verschiedenen Softwareartefakten wird ein neuer Ansatz vorgeschlagen, der auf der Funktionsanpassung aus sozialen Netzwerken wie z. B.

Facebook oder XING beruht. In einem SECO soll das implizite Wissen von Herstellern, Zulieferern, Entwicklern und Anwendern reduziert werden und die einfache Verwaltung von Softwareartefakten gefördert werden.

In einem Software-Entwicklungsprojekt trifft das Projektmanagement eine Menge an Entscheidungen wie z. B. Releaseplan, Projektziele usw. Diese Entscheidungen beruhen auf dem Wissen, welches im SECO verteilt ist. Der Umgang mit der dezentralen und geteilten Wissensbasis wird zu einem Kriterium, das über den Erfolg oder das Versagen eines Produktes entscheidet. Wenn die Parteien unabhängig ihre Entscheidungen nur auf den Teilen der Informationen machen, steigt die Komplexität des Entscheidungsproblems.

Derzeit sind die Produkt-Entscheidungen nur innerhalb eines Teams verfügbar, das ein Artefakt entwickelt. Dieser Zustand muss aber aus folgenden zwei Gründen verbessert werden:

1. Wenn ein neuer Akteur ins SECO kommt, benötigt er möglicherweise Informationen über die komplexen Anhängigkeiten zwischen geteilten Artefakten, und dies kann schwierig werden, da die Informationen außerhalb der Organisationsgrenzen liegen.
2. Wenn Akteur das SECO verlässt nimmt er Wissen mit und dadurch geht es für das SECO verloren.

Dies kann durch das Erstellen einer Social-Network-Seite erfolgen, wo die Software-Artefakte im Vordergrund stehen. Dadurch kommen neue Akteure an die erforderlichen Informationen und das Wissen bleibt in SECO. Durch diesen Ansatz wird das implizite Wissen extrahiert und für jeden in SECO verfügbar gemacht.

Paper

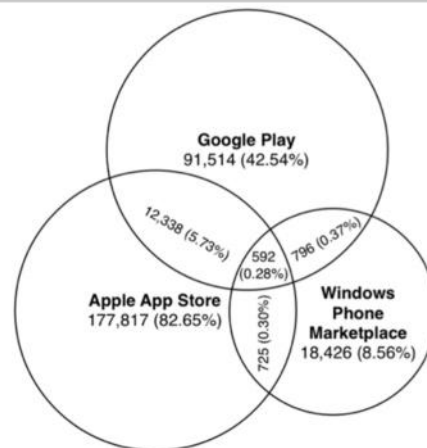
IWSECO 2009: Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems

IWSECO 2010: Knowledge management in software ecosystems - software artefacts as first-class citizens

IWSECO 2011: A Systematic Mapping Study on Software Ecosystems

Business

- Einnahmemodelle
- Geschäftsmodelle
- Multi-Homing



IWSECO 2012: App Store, Marketplace, Play! An Analysis of Multi-Homing in Mobile Software Ecosystems

MICHAEL GALETZKA, WALDEMAR FLAT

24

Hier wird der Wirtschaftsaspekt im SECO diskutiert.

Einnahmemodelle/Geschäftsmodelle

Software-Unternehmen erwirtschaften ihre Einnahmen durch den Verkauf von Software-Lizenzen und Dienstleistungen an die Kunden.

Man versucht die Einnahmequellen von Software-Firmen zu modellieren und dabei folgende Fragen zu beantworten:

- Aus was bestehen die Erlösmodelle?
- Wie unterscheiden sich die Einnahmemodelle von verschiedenen Unternehmen?

Nach der Strukturierung der Erlösmodelle und der Beschreibung der Attribute der Einnahmequellen, werden die Beziehungen zwischen Erlösmodellen und Geschäftsmodellen gezeigt. Des Weiteren bietet man Antworten auf die Fragen mit Hilfe eines semi-formalen Ansatzes zur Klassifizierung und der Modellierung von Geschäftsmodellen und Erlösmodellen.

Multi-Homing

Multi-Homing ist eine Strategie, wo ein Entwickler seine Artikel für mehrere Plattformen veröffentlicht. In dem vorliegenden Paper wurden die Daten von mehr als 850.000 Anwendungen und deren Entwickler aus Apple App Store, Google Play

und Microsofts Windows Phone Marketplace gesammelt. Die Entwickler in mehreren SECO's tätig sind wurden dann identifiziert. Die Ergebnisse zeigen, dass Multi-Homing nur von einer kleinen Gruppe von Entwicklern genutzt wird. Hinzu kommt, dass die Anwendungen, die auf verschiedenen Plattformen verfügbar sind, offenbar sich nicht durch die Art oder die Beliebtheit von Single-Homing Anwendungen unterscheiden. Wenn man sich entschließt in mehreren Plattformen seine Produkte anzubieten, dann hat man zwar einen potenziell größeren Markt, ist aber mit mehr Kosten bei der Portierung der Produkte verbunden.

Es ist sehr wenig über Multi-Homing Anbieter bekannt und ebenso auch ihre Auswirkungen auf die Anwendungen, Entwickler und SECO. Deshalb ist das wichtigste Forschungsziel herauszufinden wie Multi-Homing sich auf die SECOs auswirkt. Anwendungsangebot eines SECO's kann benutzt werden um sich von der Konkurrenz zu unterscheiden. (Es wird damit sogar Werbung gemacht wie z. B. Apple verwendete den Ausdruck „Es gibt eine App dafür“.) Allerdings wenn Multi-Homing eine verbreitete Strategie für Entwickler wird, dann ist die Unterscheidung schwer. Zum Verständnis der Popularität von diesem Ansatz wurden die drei größten App-Stores analysiert, um zu erfahren wie viele Anwendungsentwickler zur Zeit die Strategie von Multi-Homing nutzen.

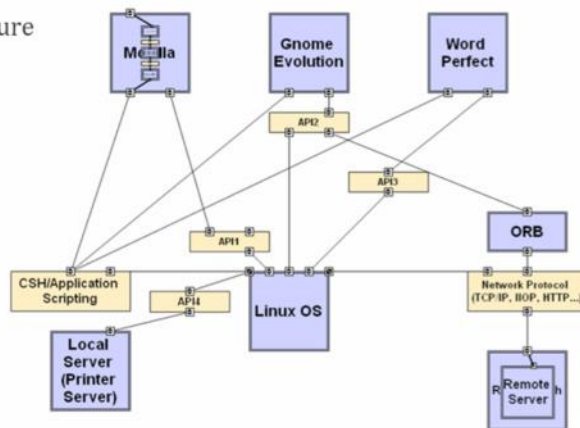
Paper

IWSECO 2011: Hybrid Revenue Models of Software Companies and their Relationship to Hybrid Business Models

IWSECO 2012: App Store, Marketplace, Play! An Analysis of Multi-Homing in Mobile Software Ecosystems

Lizenzen

- Die Rolle von Lizenzen in Open Architecture
- Kommerzielle Nutzung von Open Source



IWSECO 2009: The Role of Software Licenses in Open Architecture Ecosystems

In diesem Abschnitt wird besprochen, welche Rollen die Lizenzen in SECO spielen.

Die Rolle von Lizenzen in OA SECO

In einem SECO können die Komponenten Open Source oder proprietär mit einer offenen API sein. Solche Systeme entwickeln sich nicht nur durch die Evolution von eigenen Komponenten, sondern auch durch das Ersetzen von externen Komponenten durch Andere, möglicherweise von anderem Anbieter oder unter anderer Lizenz.

Die Unternehmen integrieren Komponenten in ihre Produkte, die zum größten Teil wo anders produziert werden. Zum Verbinden von externem Code müssen notwendigerweise Schichten geschaffen werden um das erwartete Ergebnis zu erfüllen. Ein OA-Entwicklungsprozess endet in einem Ökosystem, wo der Integrator durch die Ziele, Schnittstellen, Lizenzentscheidungen und Release Zyklen von Komponenten-Entwickler beeinflusst wird. Des Weiteren wird er auch von Kundenwünschen gesteuert. Als Ergebnis werden die Software-Komponenten wiederverwendet und die resultierenden OA-Systeme profitieren damit von Vorteilen wie fallende Kosten durch die Wiederverwendung, steigende Zuverlässigkeit und

potenziell erhöhte Agilität um auf die Veränderungen reagieren zu können. Die Herausforderung an diesen Systemen sind die heterogen-lizenzierten Komponenten d.h. es gibt mehrere Lizenzen, die alle beachtet werden müssen.

Die Entwicklungen in einem OA System werden durch unterschiedliche Mechanismen gesteuert. Einige davon findet man auch in anderen Systemen und Andere sind das Ergebnis des Einsatzes von verschiedenen Lizenzen im System.

- Änderungen in Komponenten können die globalen Eigenschaften beeinflussen.
- Ersatz-Komponenten können das gleiche Interface haben aber unterschiedliches Verhalten. Oder es muss eine Schicht entwickelt werden, damit die neue Komponente verwendet werden kann.
- Durch unterschiedliche System-Konfigurationen wird die Architektur und die Eigenschaften verändert. Durch Konfiguration kann eine Komponente angeschlossen werden, die Auswirkung auf die Rechte und Pflichten des Gesamtsystems hat.
- Die Lizenz einer Komponente kann sich ändern.
- Es kann sein die Kunden zusätzliche Rechte sich wünschen oder einige Rechte nicht mehr brauchen. Durch die neuen Anforderungen kann es bedeuten, dass man zu neuen Komponenten wechselt oder dass man die gewünschten Rechte mit akzeptablen Pflichten anbietet.

Diese Evolutionsmechanismen fordern, dass die System-Entwickler ständig die Rechte und Pflichten von ihren Systemen prüfen müssen. Damit das ganze automatisch passiert, braucht man ein geeignetes Tools dafür. Das Tool muss folgende Eigenschaften besitzen:

- Die Lizenzstruktur von Rechten und Pflichten muss rechtswirksam und prüfbar sein. Z. B. die Aussage „... nach Meinung von Experten ...“ ist nicht rechtswirksam und nicht prüfbar, da man nicht prüfen kann was jemand meint.
- Es muss die Unterschiede berücksichtigen zwischen Design-, Entwicklungs- und Auslieferungs-Phase.
- Das Tool muss unterscheiden können welche Architektur-Komponenten für die Softwarelizenz relevant sind und eine Auswirkung auf die Rechte und Pflichten

haben.

- Der Nutzer soll in der Lage sein Lizenz-Architekturen zu definieren.
- Automatische Umgebung zum Erstellen und Verwalten von Lizenz-Architekturen.
- Automatische Berechnung von Rechten und Pflichten anhand der System-Architektur.

Kommerzielle Nutzung von Open Source

Das Ziel hier ist die kommerzielle Nutzung von Open Source Software anzubieten. Des Weiteren zu beschreiben wie Open Source in hybrid-kommerziellen Geschäftsmodellen eingebettet ist.

Und zum Schluss wie man in diesen Modellen Open Source Lizenzen wirksam einsetzen kann.

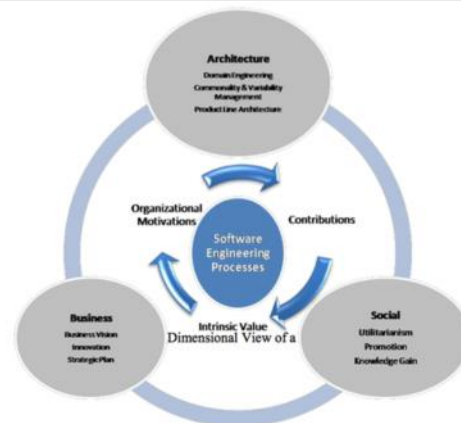
Paper

IWSECO 2009: The Role of Software Licenses in Open Architecture Ecosystems

IWSECO 2012: Leveraging Open Source Licenses and Open Source Communities in Hybrid Commercial Open Source Business Models

Modelle

- 3-dimensionale Sicht von SECO
- Strategiemodell
- Bewertungsmodell
- Beziehungsmodell
- Prozessmodellierung



IWSECO 2010: A three-dimensional view of software ecosystems

In IWSECO Workshop werden zahlreiche Modelle vorgestellt, die dazu benutzt werden SECO zu beschreiben und dadurch besseres Verständnis fördern.

3-dimensionale Sicht von SECO

Dieses Modell besteht aus den 3 Säulen:

1. Business: Die Business-Dimension des SECO wird durch drei Faktoren von Vision, Innovation und strategische Planung angetrieben.
 - Business Vision: Ist eine Vorstellung vom gewünschten zukünftigen Zustand. Es ist keine Hoffnung und kein Traum sondern eine Verpflichtung ein bestimmtes Ziel zu erreichen.
 - Innovation: Innovation und kontinuierliche Verbesserungen von Prozessen und Produkten, die die Fähigkeiten einer Organisation veranschaulichen. Kreativ und Pioniere in der Produktentwicklung zu sein, so dass man zu den wichtigsten Teil im Marktsegment zählt.
 - Strategische Pläne: Stehen im Mittelpunkt der Bemühungen eines Unternehmens, um das gewünschte Maß an Leistung in einem bestimmten Bereich zu erreichen.
2. Architektur: Hier liegt das Anliegen nicht in der Entwicklung eines einzelnen Produktes, sondern der Entwicklung von mehreren Produkten durch das Teilen

einer gemeinsamen Plattformarchitektur und durch die Beteiligung externer Entwickler.

- **Domain Entwicklung:** In der Anfangsphase sollte ein umfassendes Domain-Engineering gemacht werden, wo der Anwendungsbereich der Plattform definiert und eingegrenzt wird. Dadurch werden die SECO-Eigenschaften und die Produkte, die das SECO umgeben, identifiziert.
 - **Produktlinien Architektur:** Bietet die Möglichkeit externe Entwickler in die Entwicklung einzubeziehen, indem ihnen die Kern-Architektur zur Verfügung gestellt wird.
 - **Allgemeinheit & Variabilität Management:** Allgemeinheit Management befasst sich mit der Art, Eigenschaften und Merkmalen, die für alle Produkte gleich sind. Während Variabilität Management verwaltet die variablen Eigenschaften und Merkmale von verschiedenen Produkte im selben SECO.
1. **Soziale Aspekte:** Die soziale Dimension hat eine enge Beziehungen zu den Open Source Software Gemeinschaften, die seit vielen Jahren in den gemeinschaftlichen Entwicklungen von Anwendungen beteiligt waren.
- **Nützlichkeitsprinzip:** Die Dritten werden motiviert sich am SECO zu beteiligen und man sucht eine Form der Belohnung ihr Engagement für das SECO zu entschädigen. Dies kann entweder in Form von Geldzahlung oder einer anderen Art der Entschädigung erfolgen.
 - **Wissensgewinn:** Engagement in SECO ermöglicht es den Entwicklern ein Teil einer Entwickler-Gemeinschaft zu werden und ihre Fähigkeiten und Kenntnisse durch ein gemeinschaftliches Medium zu entwickeln.

Während die Geschäfts- und Architektur-Dimension in gewisser Weise statisch sind, ist der Erfolg des SECO von der sozialen Dimension abhängig. D. h. von der Akzeptanz der Nutzer.

i* Framework

Die vorhandenen Modelle zeigen nur die Beziehungen innerhalb eines SECO aber nicht strategischen Aspekte davon. Dieser Mangel wird durch die Modellierung mit dem i* Framework behoben. Strategische Modellierung konzentriert sich auf die Untersuchung und Analyse der strategischen Ziele, Absichten und Beziehungen der einzelnen Akteure in einem Netzwerk. Das Modell trägt zum SECO-Verständnis bei und zeigt alternative Wege zur Erreichung von strategischen Zielen. Es ist ein Mittel, um das "Warum" hinter einer Organisationsentscheidung zu beantworten.

Strategie Bewertungsmodell

Das Problem für viele Unternehmen ist, wie sie ihr SECO bestehend aus Zulieferern, Entwicklern und Partnern verwalten sollen. Diese Akteure und vor allem der Plattform-Besitzer haben oft Schwierigkeiten Einblick in ihr SECO zu bekommen und zu bestimmen welchen Einfluss ihre Handlungen auf das gesamte SECO und seine

Leistung haben.

Der Erfolg des eigenen Unternehmens hängt nicht mehr von der eigenen Entwicklungsqualität ab, sondern auch von der Art und Weise wie die Beziehungen geführt werden. Ein Bewertungsmodell dient dem SECO-Verwalter, wenn es richtig eingesetzt wird, richtige Strategien zur Erhaltung und Verbesserung der SECO-Gesundheit zu treffen.

Diverse Beziehungsmodelle

Die Beziehungsmodelle ermöglichen Einblick in die SECO Struktur.

Durch solche Modelle lassen sich leichter Cluster bzw. Subsysteme innerhalb SECO identifizieren. Diese Subsysteme können leichter von der SECO-Verwaltung gesteuert werden.

Prozessmodellierung

Das Papier behandelt die Lücke der Notwendigkeit für eine präzise Prozessmodellierung. Das Paper basiert auf Erfahrungen aus der Analyse eines SECO für mobiles Lernen und bringt verschiedene Aspekte und Einsichten für diese besondere Domain. Mobiles Lernen bedeutet Lernen mit mobilen Geräten.

Hier gibt unterschiedliche Herausforderungen zu beachten:

- Es gibt vielfältige Geräte
- Einbeziehung von Stakeholdern
- Wiederverwendung und Anpassung
- Variabilität auf vielen Ebenen:
 - Unterstützung von verschiedenen Lern-Aktivitäten, d. h. jeder Schüler hat einen eigenen Lehrpfad.
 - Verwaltung und Nutzung von Kontextänderungen.

Paper

IWSECO 2010: A three-dimensional view of software ecosystems

IWSECO 2010: Software ecosystems - a software ecosystem strategy assessment model

IWSECO 2010: On the role of software process modeling in software ecosystem design

IWSECO 2010: Industry taxonomy engineering - the case of the European software ecosystem

IWSECO 2011: A Survey of Associate Models used within Large Software Ecosystems

IWSECO 2011: Understanding Software Ecosystems - A Strategic Modeling Approach

IWSECO 2012: Defining Software Ecosystems - A Survey of Software Platforms and Business Network Governance

IWSECO 2012: EcoSysNetworks - A Method for Visualizing Software Ecosystems

Trend(s)

- Historische Daten auswerten
- Zunehmend auch in der Industrie beliebt
- Hilfe durch die Gemeinde



MICHAEL GALETZKA, WALDEMAR FLAT

27

Der Trend für den IWSECO Workshop ist Open Source und das aus unterschiedlichen Gründen.

Die Fülle und Zugänglichkeit von Software-Projekten mit historischen Daten ist frei verfügbar.

Die aktuellen Forschungsergebnisse müssen validiert werden und hier bieten die Open Source Projekte jede Menge an historischen Daten, die frei verfügbar sind. Aber auch die aktuell laufenden oder geplanten Studien habe hier gute Aussichten. Was hier auch Vielfältigkeit der Open Source Projekte ist enorm. Die Palette reicht von Komponenten, Frameworks, Entwicklungsumgebungen, Webserver bis zu Betriebssystemen, Plattformen.

Die zunehmende Beliebtheit von Open-Source-Software, auch in der Industrie.

Darüber hinaus wird es auch in der Industrie immer beliebter. Zu diesem Trend hat auch die Qualität der Produkte beigetragen, die bereits über mehrere Jahrzehnte entwickelt werden. Das bedeutet für die Forschung, dass hier schon viele Evolutionschritte stattgefunden haben und man weiß auch was die Gründe dafür waren.

Hilfe durch die Gemeinde

Zum Schluss hilft die Open Source Gemeinde die Möglichkeit, wissenschaftliche Ergebnisse über diese Systeme zu veröffentlichen und erlaubt damit anderen Forscher die erhaltenen Ergebnisse zu überprüfen und zu reproduzieren.

Dieser Trend wird durch die Themen 2013 bestätigt:

Open Source Themen

1. On Clusters in Open Source Ecosystems
2. On the Software Ecosystem Health of Open Source Content Management Systems
3. Hadoop and its evolving ecosystem
4. Towards the roles and motives of open source software developers

Rest Themen

1. Reviewing the Health of Software Ecosystems – A Conceptual Framework Proposal
2. Towards the Analysis of Software Projects Dependencies: An exploratory visual study of software ecosystems

Offene Fragen und Herausforderungen

Als Nächstes werden die Offenen Fragen aus den Workshops vorgestellt und die bestehen Herausforderungen erläutert.

Offene Fragen

Much more work remains to be done ... The models in this paper are much simplified ...
Additional research is needed to understand how ...
The overview serves as a jumping board for future research.
Part of our future work is to more fully model ...
Several open problems remain insufficiency of literature and documents, lack of time ...
The results presented herein are not completely evaluated.
This article has only scratched the surface ... Not a lot of research however is done ...
Furthermore, studying ... may help improving ...
Future work consists of expanding the proposal
with other case studies and with expert-based surveys ...
We advice future work to further and deeper investigate ...
At the current stage it is hard to claim completeness ...

MICHAEL GALETZKA, WALDEMAR FLAT

29

Aufgrund der Neuheit des Forschungsgebietes sind noch viele Fragen offen. Da das Gebiet sehr groß ist und noch weiter ansteigt, wird es noch einige Zeit dauern, bis man das SECO formal beschreiben kann.

Die Studien müssen evaluiert und erweitert werden, um die Modelle verallgemeinern zu können.

Zukünftige Themen wären:

- (1) Es soll eine qualitative Studie von Multi-Homing Entwicklern und Anwendungen gemacht werden, um die Unterschiede zu Single-Home zu erforschen. Des Weiteren könnte man auch besser erklären die Motivationen und Strategien der Unternehmer.
- (2) Welchen Einfluss hat das eigene SECO auf die Konkurrenz bzw. wie werde ich beeinflusst.
- (3) Empirische Studien notwendig, um den Ansatz i* Modellierung zu validieren. Die Wirksamkeit der Strategien, Argumentation und Design kann durch Fallstudie an realen SECO's nachgewiesen werden.

Paper

IWSECO 2009

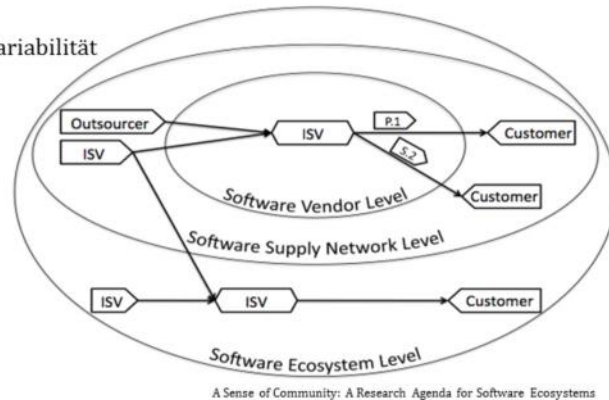
IWSECO 2010

IWSECO 2011

IWSECO 2012

Herausforderungen

1. Organisationsebene
 - Produktplanung
 - Wissensmanagement
 - Erweiterbarkeit, Portabilität und Variabilität
2. Zulieferer Ebene
 - Aufbau von Beziehungen
 - Release Planung
 - Qualitätsmanagement
3. SECO Ebene
 - Modellierung
 - Verwaltung
 - Wirtschaften



A Sense of Community: A Research Agenda for Software Ecosystems

Herausforderungen für ein SECO gibt es auf allen 3 Architekturebenen.

Organisationsebene

- Produktplanung: Die Anbieter müssen entscheiden in welcher Konfiguration die neuen Komponenten veröffentlicht wird. Koordination zwischen Entwicklungs- und Release-Planung. Ständig überdenken: Was kann ich wiederverwenden? Was muss ich selbst entwickeln? Was kann ich kaufen?
- Wissensmanagement: Wie viel Wissen soll ich teilen? Wie offen soll mein Interface sein? Soll ich alle Bugs an die Beteiligten melden?
- Erweiterbarkeit, Portabilität und Variabilität: Wie flexibel soll die Software-Architektur sein? Kann die Software einmal entwickelt werden und dann auf unterschiedliche Plattformen, Architekturen und mit verschiedenen Konfigurationen deployed werden?

Zulieferer Ebene

- Aufbau von Beziehungen: Methoden zum Identifizieren von Partnern und Beziehungen werden benötigt, um die Softwarehersteller bei der Gründung und Entwicklung ihrer Netzwerke zu unterstützen. Des Weiteren braucht man

Strategien wie man sich seinen potentiellen Kunden und Zulieferern präsentiert.

- Release Planung: Wann ist der optimale Zeitpunkt neue Versionen der Komponenten oder Services zu veröffentlichen? Für einige ist es so schnell wie möglich, damit man die neuen Features nutzen kann.
- Qualitätsmanagement: Wer kontrolliert die Qualität in einem Netzwerk? Zulieferer oder der Kunde?

SECO Ebene

- Modellierung: Die formalen Eigenschaften von SECO müssen festgelegt werden. Es gibt zur Zeit keine Modellierungsformalismen für SECO.
- Verwaltung: Wie sollen die SECO's verwaltet werden. Welche Entwicklungsstandards und Strategien soll es geben.
- Wirtschaften: Die Herausforderung besteht hier in der Auswahl des Geschäftsmodells. Open Source oder proprietär?

Paper

A Sense of Community: A Research Agenda for Software Ecosystems

Diskussion

Die nachfolgenden Folien sollen als Grundlage für eine Gruppendiskussion nach der Präsentation dienen, sollten die Zuhörer nicht von sich aus mit ihren Fragen eine Diskussion starten.

Diskussion

- Ist es wirklich etwas Neues?
 - Plattformen für Software gibt es schon sehr lange (z.B. Betriebssysteme)
 - Sind die einzelnen Software-Bestandteile von SECOs nicht einfach nur die bekannten Komponenten?

Diskussion

- Ist die SECO Forschung nur ein Hype?
 - Sind SECOs wichtig und komplex genug um ein eigenes Forschungsfeld zu haben?
 - Sind die neueren Paper wirklich neu oder werden nur alte Paper „recycelt“?
 - Bringt die Forschung einen Mehrwert oder folgt sie nur als Hype auf den Erfolg von Smartphones?

Diskussion

- Wie aussagekräftig sind die vorgestellten Forschungsergebnisse?
 - Wie gut sind die Aussagen mit Daten belegt?
 - Lassen sich die Ergebnisse auf die Praxis übertragen oder sind sie zu theoretisch?