

Herausforderungen bei der Anpassung von Open Source Software an neue Einsatzbereiche

Dietrich Boles, Markus Schmees

Abteilung Informationssysteme, Universität Oldenburg
Serviceeinheit Portal, ELAN AG Oldenburg
Escherweg 2
26121 Oldenburg
dietrich.boles@informatik.uni-oldenburg.de
markus.schmees@elan-niedersachsen.de

Abstract: Spezielle Anforderungen an Software führen häufig zu aufwändigen, lang andauernden und damit teuren Neuentwicklungen. Funktionsfähige Prototypen entstehen i.d.R. erst im Laufe dieses Entwicklungsprozesses, wodurch der Gesamtfortschritt durch die Auftraggeber nur schwer einzuschätzen ist. Gleichzeitig existiert eine Vielzahl verschiedener, domänenfremder Open Source Softwaresysteme (OSS), die bereits Teile der gewünschten Funktionalität beinhalten. Nachfolgend wird eine Methode vorgestellt, die dabei hilft, Funktionalität vorhandener Software auf eine neue Domäne zu übertragen, indem sie die traditionellen Aufgaben des Softwareentwicklungsprozesses begleitet. Bei der Umwandlung eines Lernmanagementsystems in eine Bücherplattform und einen E-Learning Katalog werden diese Methode exemplarisch angewendet, Ergebnisse der Untersuchung dargelegt sowie Schwierigkeiten einer solchen Anpassung erörtert.

1 Einleitung und Methode

Um die speziellen Anforderungen an ein zu erstellendes Softwaresystem abzudecken, liegt die Idee nahe, dieses nicht komplett neu zu entwickeln, sondern den im Rahmen freier Software bereits erstellten Code wiederzuverwenden und auf einen neuen Aufgabenbereich zu übertragen. Daher wurde eine Methode erarbeitet, die diesen Übertragungsprozess strukturiert, indem sie die Phasen der Softwareentwicklung (Planung, Analyse, Entwurf, Programmierung und Validierung) sowie die unterstützenden Prozesse (Anforderungs-, Prozess-, Qualitäts-, Konfigurationsmanagement und Dokumentation) begleitet. Die zugehörigen Schritte sind nachfolgend kurz beschrieben:

1. **Leitbild formulieren:** Zunächst werden Ziele spezifiziert, die mit dem zu entwickelnden System angestrebt werden. Aus diesen groben Anforderungen lässt sich dann im Folgenden die Funktionalität ableiten, die das Zielsystem beinhalten soll.
2. **Geeignetes Grundsystem auswählen:** Auf Grundlage dieser Spezifikation erfolgt die Auswahl eines bestehenden OSS, das einen hinreichend stabilen Zustand erreicht hat und möglichst viele Teile der benötigten Funktionalität schon abdecken sollte. Bereits gesammelte Erfahrungen bei Einsatz oder Erweiterung eines solchen Systems können helfen, sich leichter in den fremden Programmcode einzufinden.

3. **Metapher übertragen:** Konzeptionell besteht der fundamentale Schritt bei der Übertragung eines existierenden Systems auf eine neue Domäne in der Abbildung der zugrunde liegenden Metapher auf eine passende Metapher des künftigen Einsatzbereichs. So können aus der *Unterstützung einer Lehrveranstaltung* z.B. *Mehrwerte eines Buches* oder *erweiterte Katalogmetadaten* werden.
4. **An neue Domäne anpassen:** Anschließend wird die bestehende Software an die neue Metapher angepasst. Insbesondere ist eine Anpassung der Begriffswelt wichtig, um Missverständnisse zu vermeiden. Alte Begriffe werden durch Begriffe des neuen Einsatzbereiches ausgewechselt, alte Bilder durch passendere ausgetauscht. Ausführliche und bisher durchaus zu Recht mehrstufige Abläufe können dem neuen Zweck entsprechend vereinfacht und unter Umständen verkürzt werden.
5. **Unnötige Funktionalität ausblenden:** Im so entstandenen Grundsystem sollten dann bestehende, aber im neuen Einsatzbereich nicht mehr benötigte Funktionen identifiziert und ausgeblendet werden, damit sie die späteren Anwender bei ihrer Arbeit oder im Verständnis der Metapher nicht irritieren können.
6. **Fehlende Komponenten ergänzen:** Da ein domänenfremdes System i.d.R. nicht sämtliche Anforderungen an den neuen Einsatzbereich abdeckt, gilt es, die noch fehlende Funktionalität auf traditionelle Weise zu erstellen bzw. aus anderen Quellen wiederzuverwenden und in das vorbereitete System zu integrieren.

Die Anwendung dieser Methode eignet sich insbesondere im Rahmen evolutionärer Softwareentwicklung, da auf diese Weise kontinuierlich ein lauffähiges System zur Verfügung steht, das bis zur Produktreife weiterentwickelt wird und laufend mit den Anforderungen der Auftraggeber abgestimmt werden kann. Bei der Auswahl eines anzupassenden OSS spielen technologische Kompetenzen sowie die bei Einsatz und Erweiterung dieses Systems bereits gesammelten Erfahrungen eine große Rolle. Nachfolgend wird die Anwendung dieser Methode zur Anpassung eines bekannten OSS auf zwei unterschiedliche Domänen exemplarisch beschrieben.

2 Anpassung des Lernmanagementsystems Stud.IP

Das Lernmanagementsystem (LMS) Stud.IP¹ ist eine internetbasierte Arbeitsumgebung, die auf der Programmiersprache PHP basiert und als freie Software unter GNU General Public License² (GPL) vorliegt. Es dient der Online-Unterstützung von Lehrveranstaltungen, insbesondere an Hochschulen. *Dozenten* können in Stud.IP für ihre *Lehrveranstaltungen* elektronische Pendanten anlegen und verwalten. Für jede dieser Veranstaltungen werden automatisch Diskussionsforen, Ablaufpläne, Dateiodner, Umfragen, Evaluationen, Wiki-Webs, Teilnehmerlisten, Ankündigungen, Termine, Chaträume etc. bereitgestellt. *Studierende* können sich bei den Veranstaltungen anmelden und die entsprechenden Online-Mehrwerte in Anspruch nehmen sowie selbst aktiv daran mitarbeiten.

¹ <http://www.studip.de/>

² <http://www.gnu.org/copyleft/gpl.html>

Veranstaltungen werden in Stud.IP durch übergeordnete *Einrichtungen* (Fachbereiche, Institute, etc.) zusammengefasst, welche die Hochschulen bei der Organisation der Lehre durch Stundenpläne, Terminkalender, Veranstaltungsverzeichnisse, Anmeldeverfahren, Raumplanung sowie Möglichkeiten zur Selbstdarstellung unterstützen.

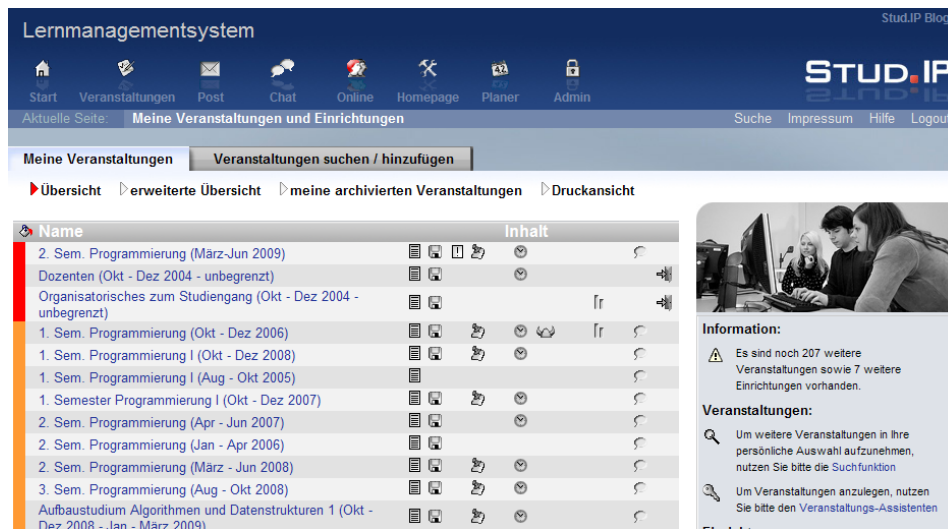


Abbildung 1: Veranstaltungen in Stud.IP

2.1 Vom LMS zur Bücherplattform

BookIP³ ist eine Internet-Plattform, die Web-2.0-konforme Mehrwerte rund um Bücher (insbesondere Lehrbücher) offeriert. Ziel von BookIP ist es, Verlage, Autoren und Leser stärker miteinander zu vernetzen und die Kommunikation untereinander zu verbessern.

Über BookIP können Verlage und Autoren Zusatzmaterialien (Leseproben, Errata, zusätzliche Kapitel und Aufgaben, Musterlösungen, Software, Folienmaterialien für Vorlesungen, Podcasts, etc.) zu ihren Büchern bereitstellen und jederzeit aktualisieren. Autoren können sich auf persönlichen Homepages vorstellen und über Chats virtuelle Lese- und Fragestunden anbieten. Leser haben die Möglichkeit, virtuelle Bücherregale zu erstellen, jederzeit Informationen zu einzelnen Büchern und Autoren abzurufen sowie Zusatzmaterialien herunter- und (wenn vom Autor gestattet) auch hochzuladen. Sie können gelesene Bücher online rezensieren, in Foren inhaltlich über einzelne Bücher und Themen diskutieren oder sogar über Wikis eigene Ergänzungen zu Büchern veröffentlichen.

Trotz der beträchtlichen funktionalen Komplexität des Systems konnte BookIP mit der in Abschnitt 1 vorgestellten Methode auf der Grundlage des Stud.IP-LMS innerhalb weniger Wochen entwickelt und produktiv in Betrieb genommen werden. Die genauen Entwicklungsschritte sahen dabei wie folgt aus:

³ <http://www.bookip.de/>

Zunächst wurden die oben skizzierten Ziele detailliert beschrieben (Schritt 1). In einer ersten Entwicklungsphase wurde das LMS Moodle⁴ als zugrunde liegendes System ausgewählt. Es stellte sich jedoch heraus, dass das Stud.IP-LMS bei der Übertragung der Metapher die geeignetere Basis darstellen sollte. Da wir weiterhin im niedersächsischen E-Learning Academic Network⁵ (ELAN) bereits massive Erfahrungen mit Einsatz, Betrieb und Erweiterung von Stud.IP gemacht hatten, fiel die Wahl auf dieses als Grundlage von BookIP (Schritt 2). Die grundlegende Transformation bei der Übertragung der Metapher (Schritt 3) bestand in der Umwandlung von *Veranstaltungen* in Stud.IP zu *Büchern* in BookIP. *Einrichtungen* wurden zu *Verlagen*, denen die Bücher zugeordnet sind. Bei der Übertragung des Rollen- und Rechtekonzeptes wurden *Dozenten* des Stud.IP zu *Autoren* in BookIP. Aus *Studierenden* wurden *Leser*. Ein Großteil der durchzuführenden Softwareanpassungen (Schritt 4) konnte durch einen einfachen Suchen-und-Ersetzen-Prozess durchgeführt werden, bei der die alte in die neue Begriffswelt überführt wurde. Das galt selbst für die Hilfe-Funktion und das Benutzungshandbuch.

Ausgeblendet (Schritt 5) wurden bspw. die Ablauf-, Stundenpläne und die Raumplanung von Stud.IP, weil es dazu in BookIP kein sinnvolles Pendant gibt. Die hauptsächlich neu zu entwickelnde Funktionalität von BookIP (Schritt 6) bestand in der Anbindung an den Amazon-Buchkatalog über die Amazon-Webservices⁶. Hierdurch können Verlage automatisch die bibliographischen Daten ihrer Bücher in das System einspielen.

Das entsprechende System wurde daraufhin zur Benutzung freigegeben und wird seither von zahlreichen Autoren und interessierten Lesenden zur Diskussion genutzt.

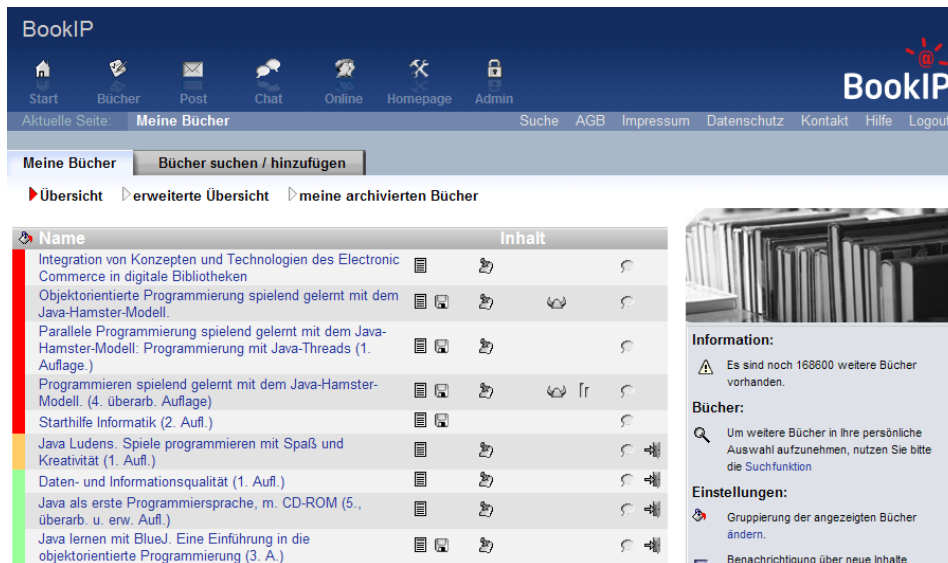


Abbildung 2: Bücher in BookIP

⁴ <http://moodle.org/>

⁵ <http://www.elan-niedersachsen.de/>

⁶ <http://aws.amazon.com/>

2.2 Vom LMS zum Metadatenkatalog

Beim E-Learning Katalog⁷ des ELAN handelt sich um eine Internet-Plattform, die Web-2.0-konforme Mehrwerte rund um E-Learning-Materialien und -Kurse offeriert. Der E-Learning-Katalog kann als Kommunikationsnetzwerk zwischen Erstellern bzw. Anbietern sowie Nutzern von E-Learning-Materialien angesehen werden. Dazu verwaltet er deren Metadaten. Content-Anbieter oder auch andere Nutzer können ihre Metadaten in den Katalog eintragen und diese pflegen. An E-Learning-Materialien Interessierte können im Katalog nicht nur recherchieren, sondern bspw. die Materialien bewerten, darüber in speziellen Foren bzw. Chats diskutieren oder Ergänzungen vornehmen.

Entwickelt wurde der Katalog von der ELAN AG⁸, die ebenfalls aus dem ELAN-Projekt hervorgegangen ist und die Arbeit der zugehörigen Teilprojekte durch unterstützende Dienstleistungen begleitet. Wegen der großen Erfahrung ihrer Mitarbeiter mit Einsatz und Erweiterung des Stud.IP-Systems im Rahmen der ersten beiden ELAN-Förderphasen, weil mit der Serviceeinheit Stud.IP ein eigenes Kompetenzteam zur Verfügung steht und um Marketing für den flexiblen Einsatz von Stud.IP zu betreiben, hat sich die ELAN AG dafür entschieden, nach Festlegung der initialen Anforderungen (Schritt 1) den E-Learning Katalog auf der Grundlage des Stud.IP-Systems aufzubauen (Schritt 2).

Bei der Metaphern-Übertragung von Stud.IP (Schritt 3) wurden die *Veranstaltungen* in Stud.IP zu *Metadaten* im E-Learning Katalog. Metadaten werden von *Content-Anbietern* verwaltet, die aus den Stud.IP-*Dozenten* hervorgegangen sind. Die Stud.IP-Rolle der *Studierenden* wurde zu *E-Learning-Interessierten* umgewandelt. Sämtliche Metadaten sind in einem fachlich strukturierten Hauptkatalog thematisch einsortiert, dessen *Verzeichnisse* den Stud.IP-*Einrichtungen* entsprechen. Materialien können per Navigation oder freie Suche über sämtliche Metadatenfelder aufgefunden werden. Während die fachliche Aufteilung an die Klassifikation von Akademisches Lehrmaterial Online⁹ (AKLEON) angelehnt ist, basieren die Metadaten auf dem im Rahmen von ELAN entwickelten ELAN Application Profile (EAP) [DHK05].

Jeder registrierte Nutzer kann eigene und interessante E-Learning-Kurse oder -Contents eintragen. Zu dem Zweck wurde im Rahmen der Domänenanpassung (Schritt 4) die Maske zum Anlegen von Veranstaltungen aus Stud.IP an eine an das EAP angelehnte Eingabemaske angepasst. Liegen Metadaten bereits im EAP-Format vor, können sie in diese Eingabemaske importiert und vor Eintragung in den Katalog auf Richtigkeit überprüft werden. Eine Kopierfunktion erlaubt die Übernahme bereits eingetragener Metadaten in eine neue Eingabemaske, falls ähnliche Eintragungen gemacht werden sollen. Das Design der Startseite wurde so überarbeitet, dass auf den ersten Blick eine navigierbare Katalogstruktur erkennbar ist. Ebenso wurde eine Seitenleiste erstellt, die Suche nach Inhalten sowie Anmeldung bzw. Registrierung am Katalog von jeder Seite aus ermöglicht.

Wie auch in BookIP wurden insbesondere organisatorische Funktionalitäten, wie Stundenpläne und Raumplanung, ausgeblendet (Schritt 5). Hinzugefügt (Schritt 6) wurde ein

⁷ <http://katalog.elan-niedersachsen.de/>

⁸ <http://www.elanag.com/>

⁹ <http://www.akleon.de/>

Modul, über das alle Nutzer selbst private und öffentliche Kataloge erstellen und diesen Inhalte nach eigener Kategorisierung zuordnen können. Die Detaildarstellung einzelner Metadaten wurde um einige fehlende Attribute sowie der Möglichkeit der Zuordnung individueller Screenshots erweitert und für jeden (auch nicht angemeldeten) Benutzer einsehbar/zugreifbar gestaltet. Um eine möglichst große Verbreitung von Informationen und Metadaten zu erreichen, wurde der XML-Sitemapgenerator¹⁰ von Google angepasst und in den Katalog integriert. Dieser erfasst einmal täglich die komplette Katalogstruktur und sendet sie zur Indizierung an die Suchmaschine. Zudem wurden ein EAP-Export von Detailinformationen sowie das OAI Protocol for Metadata Harvesting¹¹ integriert, um ein Auslesen und Abfragen der Metadaten z.B. durch externe Suchmaschinen oder Harvester zu vereinfachen. Die bereits existierende Plugin-Schnittstelle des Systems wurde verwendet, um die Wiki-Lernmodulumgebung zu integrieren, mit der Verantwortliche für die Metadaten direkt auf die beschriebenen Inhalte verlinken können.

Im Rahmen des ELAN-Projekts wurden zahlreiche E-Learning-Kurse und -Inhalte erstellt [AS09]. Nach Freischaltung des Katalogs haben zunächst die Teilprojekte ihre Ergebnisse aus dem niedersächsischen Forschungsverbund eingetragen und so nach außen dargestellt. Darüber hinaus steht der Katalog nun sämtlichen Interessierten zur Recherche und weiteren Eintragung interessanter Inhalte zur Verfügung.

The screenshot shows the 'E-Learning-Katalog des ELAN' website. The header includes the title and the ELAN logo. Below the header, there are navigation tabs for 'Katalogdetails' and 'Rankings', with 'Übersicht' selected. The main content area is divided into several sections:

- Kataloghierarchie:** A tree view showing 'Gesamtverzeichnis (2148)' and 'Informatik (27 Einträge)'.
- Zugeordnete Contents (27):** A table listing various subjects and their current ratings.
- Ihre Möglichkeiten:** A search section with a text input field and a 'Suche starten' button.
- Anmeldung:** A notice for users to register or log in to interact with the catalog.

Zugeordnete Contents (27)	Bewertung
Angewandte Informatik	Bislang ohne
Betriebssysteme und Netze	Bislang ohne
Bildkommunikation I und II	Bislang ohne
Datenkommunikationsnetze	Bislang ohne
Ein Referenzmodell zur Entwicklung multimedialer Lernprogramme	Bislang ohne
Einführung in die EDV	Bislang ohne
Electronic Design Automation	Bislang ohne
Enterprise Applications	Bislang ohne
Grundlagen relationaler Datenbanken	Bislang ohne
In 4 Sekunden um die Welt	Bislang ohne
Informationssysteme und -verarbeitung	Bislang ohne
Integrierte Netze (IgN)	Bislang ohne
Kommunikationssysteme	Bislang ohne

Abbildung 3: E-Learning-Inhalte im E-Learning Katalog

¹⁰ <https://www.google.com/webmasters/tools/docs/de/sitemap-generator.html>

¹¹ <http://www.openarchives.org/pmh/>

3 Lessons Learned, Kritische Betrachtung

Bekannte Ansätze für eine schnelle und kostengünstige Softwareentwicklung sind die Verwendung von Frameworks [Pre97] und die komponentenbasierte Softwareentwicklung [Szy02]. Die in diesem Artikel vorgestellte Methode der Umwandlung eines existierenden Systems durch dessen (Programmcode-)Anpassung an eine neue Domäne stellt eine interessante Alternative dar. Gelingt es, ein funktional ähnliches Basissystem zu finden, kann durch Übertragung seiner Metapher auf die Metapher des neuen Einsatzbereichs viel Zeit und Geld gespart werden. Grundvoraussetzung dieser Softwareentwicklungsmethode ist natürlich der Zugriff auf den Programmcode, wie es bei OSS der Fall ist. Die Architektur des Basissystems kann weitgehend beibehalten werden. An Änderungen sind insbesondere terminologische und optische Anpassungen der Benutzungsschnittstelle notwendig sowie das Auskommentieren nicht benötigter und das Integrieren zusätzlicher Funktionalitäten. Weitere Vorteile dieser Entwicklungsmethode bestehen darin, dass sehr schnell ein funktionsfähiger Prototyp als Grundlage der Kommunikation mit dem Auftraggeber zur Verfügung steht und dass das System direkt stabil läuft (insofern dies bei dem Basissystem der Fall ist). Nutzer, die das Basissystem kennen, kommen zudem ohne großen Einarbeitungsaufwand mit dem neuen System zurecht, weil ihnen die grundlegende Struktur und Interaktionsweise bekannt sind.

Auf der anderen Seite kann bestehende Software nicht alle Anforderungen eines alternativen Einsatzbereiches abdecken. Die Offenheit von Open Source Projekten führt dazu, dass sich (auch fachlich) unterschiedliche Personengruppen beteiligen. Sie begleiten das Projekt aus Sicht verschiedener Fachgebiete, was zwar ganz neue Ansätze und Impulse bietet. Dies kann aber – wie im beschriebenen Fall – auch zu monolithischem Code bzw. mangelnder Modularisierung führen und damit Wartbarkeit und Anpassbarkeit erschweren. Sind einzelne Codebereiche nur schwach voneinander getrennt, sind die Identifikation gesuchter Funktionalitäten und ihre Anpassung mühsam. Zudem ist das Herauslösen einzelner Funktionsbereiche schwierig. Im Vergleich zu proprietären Systemen sind Wartung und Weiterentwicklung angepasster OSS möglich, fallen aber wegen des Fremdcodeanteils schwerer als bei Eigenentwicklungen.

Technisch gesehen entspricht die Anpassung eines OSS an eine fremde Domäne einem neuen Zweig der Versionierungshistorie. Ausgehend von der aktuellen Version eines bestehenden Softwaresystems wird ein Schnitt gezogen und dieses komplett abgeändert, so dass künftige Änderungen im Hauptzweig nicht mehr ohne Weiteres automatisch übernommen werden können. Updates des ursprünglichen OSS oder die Behebung von Sicherheitslücken erfolgen daher nicht mehr durch einfachen Austausch bei Neuinstallation, sondern erfordern eine manuelle Anpassung geänderter Bereiche an die aktuelle Codesituation des veränderten Systems. Je nach Lizenzierungsmodell des OSS ist es erforderlich, dass der angepasste Programmcode auf Anforderung herausgegeben und der Community wieder zur Verfügung gestellt werden muss (z.B. bei der GPL). Schließlich hängt die Skalierbarkeit des angepassten Systems wesentlich von der Skalierbarkeit des Grundsystems ab, das man als Entwickler häufig nicht unter Volllast gesehen hat und das bei hohen Nutzerzahlen im Produktivbetrieb Probleme bereiten kann.

Immer wieder ist festzustellen, dass sich die Anforderungen an ein Softwaresystem schon während seiner Entwicklung ändern bzw. dass spezifizierte Anforderungen anders gemeint/gedacht waren als initial spezifiziert. Ein wesentlicher Vorteil der beschriebenen Entwicklungsmethode ist daher das vorführbare Entwicklungssystem, das von Entwicklern und Auftraggebern getestet, als Grundlage zur Diskussion genutzt und an den aktuellen Anforderungsstand angepasst werden kann. So trägt die Methode zu weniger Missverständnissen unter den Beteiligten bei und hilft, Entwicklungszeit und Geld zu sparen.

Literaturverzeichnis

- [AS09] Appelrath, H.-J.; Schulze, L.: Auf dem Weg zu exzellentem E-Learning. Waxmann, 2009.
- [DHK05] Derr, E.; Häseker, H.; Klie, T.; Mimkes, J.; Neuroth, H.; Wätjen, H.-J.: ELAN Application Profile: Metadaten für elektronische Lehr- und Lernmaterialien. In: DINI - Deutsche Initiative für Netzwerkinformationen e.V., Schriften 6-de, Version 1.0, 2005.
- [Pre97] Pree, W.: Komponentenbasierte Softwareentwicklung mit Frameworks. dpunkt, 1997.
- [Szy02] Szyperski, C.: Component Software: Beyond Object-Oriented Programming. Addison-Wesley Longman, Amsterdam, 2002.