

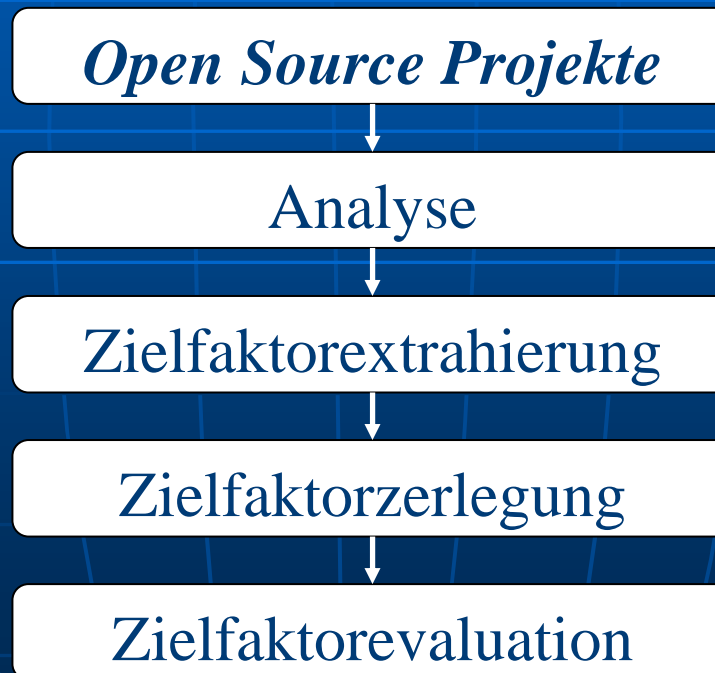
Evaluation von Open Source Projekten:

Ein GQM-basierter Ansatz

von Timo Fleischfresser

Ziele der Arbeit

- Entwicklung einer schematischen Vorgehensweise zur Evaluation von Open Source Projekten
- Durchführung einer Evaluation für einen bestimmten Zielfaktor



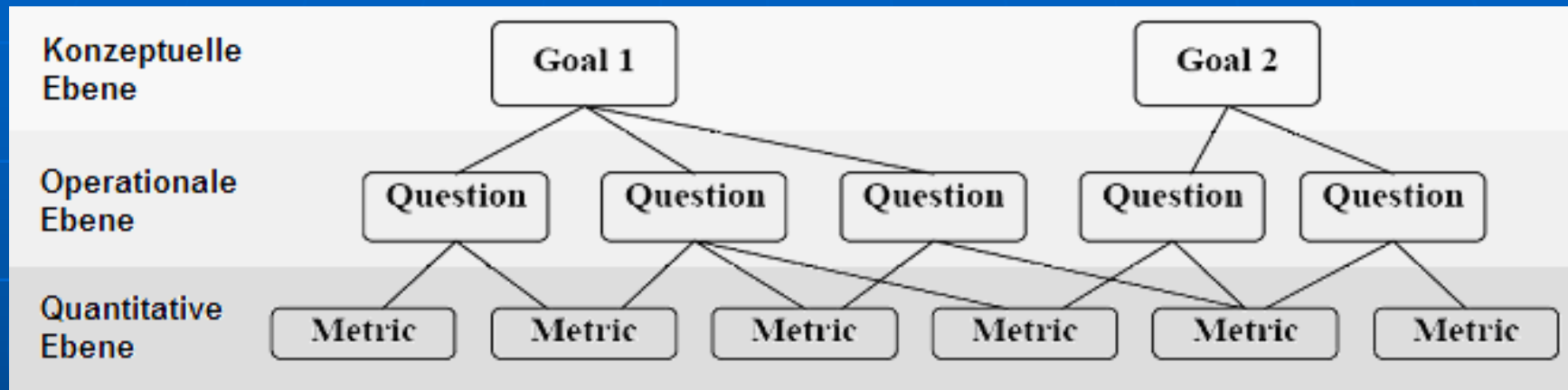
Unternehmensbefragung

- 605 angeschriebene Unternehmen
 - 315 Softwareunternehmen
 - 290 Anwendungsunternehmen
- Befragung zu den Risiken und Vorteilen eines Einsatzes von Open Source Software
- Zusammenfassung der Risiken und Vorteile zu Anforderungskategorien

Unternehmensbefragung

- Anforderungskategorien von Anwendungs- und Softwareunternehmen
 - Softwarequalität
 - Kostenkontrolle
 - Planbarkeit
 - Rechtssicherheit
 - Zukunftssicherheit
- Zusätzliche Anforderungskategorie von Softwareunternehmen
 - Nutzbarkeit für eigene Produkte

Goal-Question-Metric-Verfahren



- Iteratives Top-Down-Verfahren
- 3 Analyseebenen
 - Konzeptuelle Ebene (Ziele)
 - Operationale Ebene (Fragestellungen)
 - Quantitative Ebene (Metriken)

Goal-Question-Metric-Verfahren

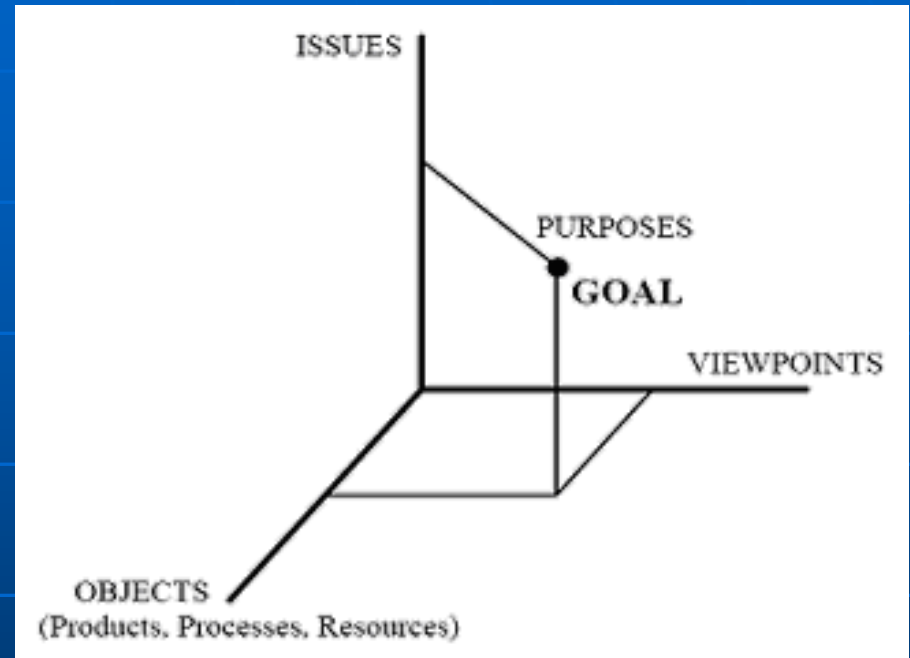
■ *Definition:*

Der Begriff „Zielfaktor“ bezeichnet die Ziele einer Untersuchung innerhalb der konzeptuellen Ebene des GQM-Verfahrens.

■ *4 Dimensionen*

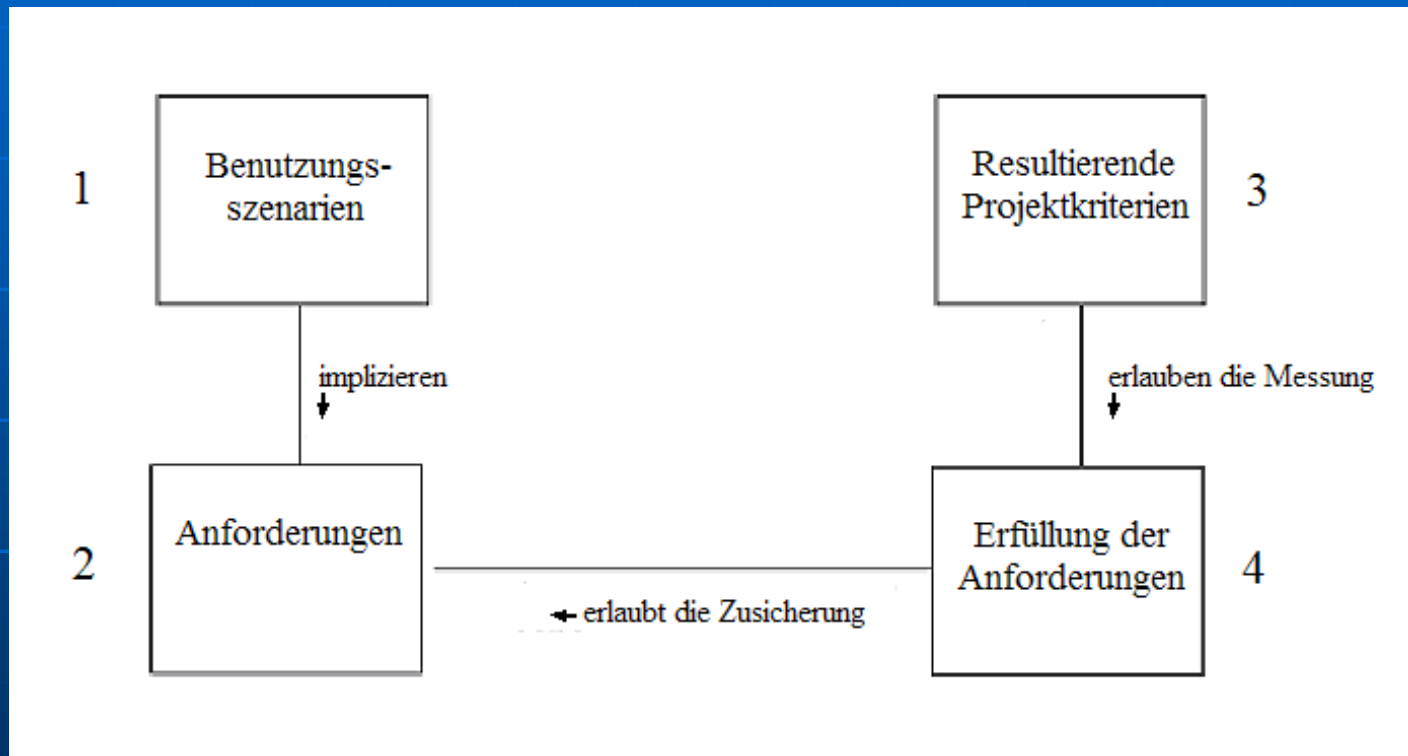
- *Object*
- *Issue*
- *Viewpoint*
- *Purpose*

Reduzierung der Softwarekosten durch den Einsatz von OSS
aus der Sicht von Unternehmensmanagern



Allgemeine Zerlegung nach Cruz et al.

Vorgehensweise



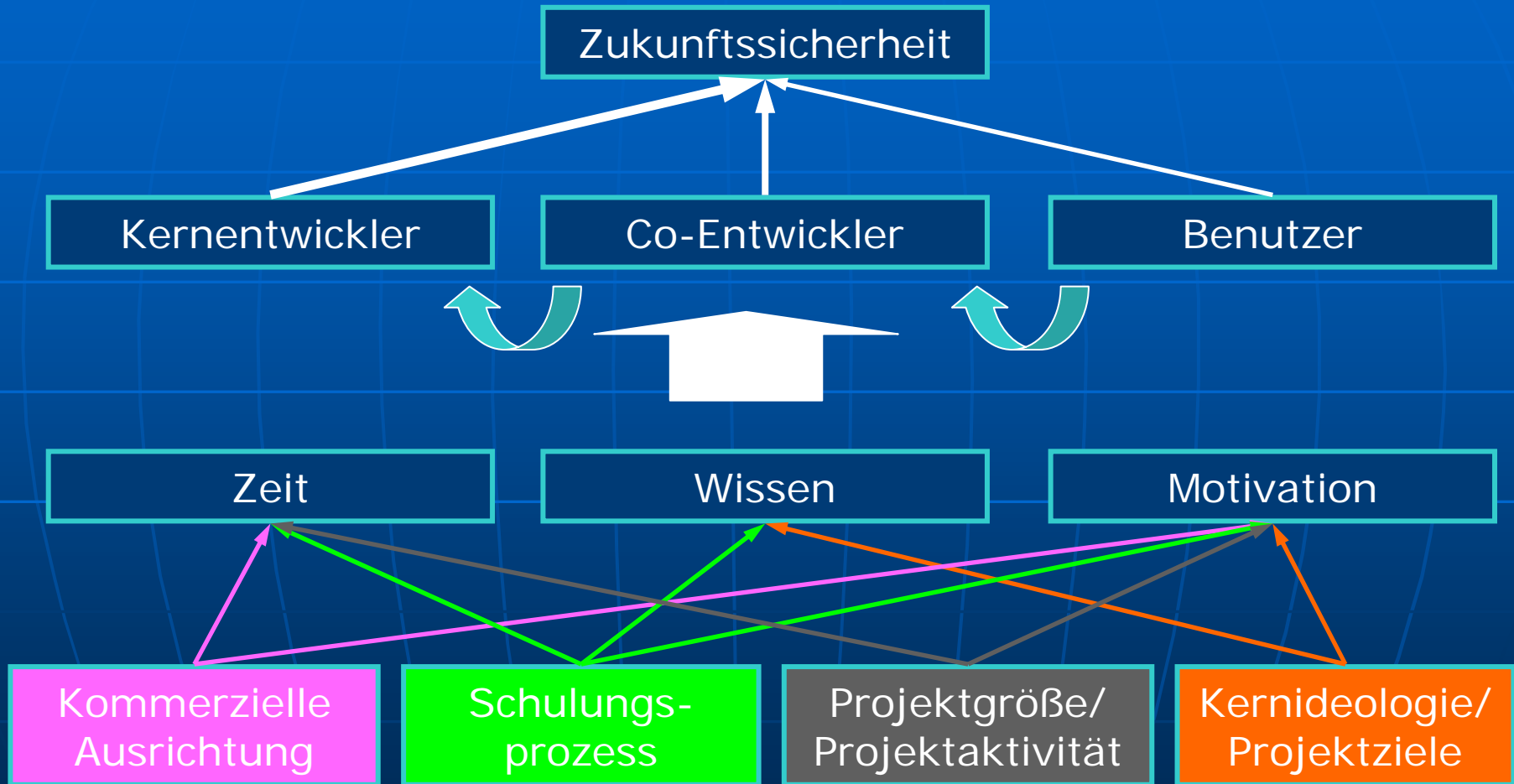
Aber:

- Sehr allgemeine Zerlegung
- Keine empirische Validierung/Durchführung

Spezieller Kontext

- Konkretisierung des Evaluationsgegenstandes „Open Source Projekt“ durch zwei Aspekte:
 - Einen Anwendungsbereich
 - Web Content Management Systeme
 - Einen speziellen Zielfaktor
 - Zukunftssicherheit (*die Wahrscheinlichkeit, mit der ein Open Source Projekt langfristig besteht und seine Software weiterentwickelt wird*)
- Der untersuchte Zielfaktor besteht in der **Evaluation** der **Zukunftssicherheit** von Open Source Projekten aus dem Bereich der Web Content Management Systeme aus der Sicht von **Unternehmen**

Zerlegung des Zielfaktors



Zerlegung des Zielfaktors

Goal	
Absicht	Evaluation
Betrachteter Aspekt	der Zukunftssicherheit
Objekt	von Open Source Projekten
Perspektive	aus der Sicht von Unternehmensmanagern
Question	Metrics
Q1 Welche Einnahmequellen besitzt das Open Source Projekt?	M1.1 Befragung der Komentwickler M1.2 Angaben auf der Projekthomepage
Q2 Wie viele bezahlte Entwickler besitzt das Open Source Projekt?	M2.1 Befragung der Komentwickler
Q3 Wie viele Unternehmen bieten Support für die Software des Open Source Projektes an?	M3.1 Befragung der Komentwickler M3.2 Angaben auf der Projekthomepage
Q4 Wie lange benötigt ein neuer Entwickler, um das Wissen eines Kernentwicklers zu erlangen?	M4.1 Einschätzung der Kernentwickler
Q5 Welche Mechanismen sind in dem Open Source Projekt etabliert, um neue Entwickler dabei zu unterstützen, dieses Wissen zu erlangen?	M5.1 Befragung der Komentwickler M5.2 Überprüfung der Projekthomepage auf vorhandene Dokumente und Strukturen zur Unterstützung von Benutzern und Entwicklern
Q6 Wie viel Zeit verbringen die Komentwickler des Projektes damit, andere Entwickler zu unterstützen?	M6.1 Befragung der Komentwickler
Q7 Welche Ideologie besitzt das Projekt und wie ist diese Ideologie manifestiert?	M7.1 Befragung der Komentwickler M7.2 Überprüfung der Projekthomepage auf formulierte Kernwerte oder eine Daseinsberechtigung M7.3 Analyse der Mailinglisten-Archive auf gemeinschaftlich getragene Prinzipien
Q8 Welche Ziele besitzt das Projekt und wie sind diese Ziele manifestiert?	M8.1 Befragung der Komentwickler M8.2 Überprüfung der Projekthomepage auf formulierte Kernwerte M8.3 Analyse der Mailinglisten-Archive auf gemeinschaftlich getragene Prinzipien
Q9 Wie groß ist die Projektcommunity?	M9.1 Angaben zur Entwicklerzahl auf der Projekthomepage M9.2 Anzahl der Entwickler, die in das Versionsverwaltungssystem eingechekkt haben?
Q10 Wie groß ist die Projektsoftware?	M10.1 Anzahl der „Lines of Code“ der Software im Versionsverwaltungssystem
Q11 Wie aktiv ist die Projektcommunity?	M11.1 Anzahl der in den letzten 3, 6 und 12 Monaten aktiven Entwickler in den Versionsverwaltungssystemen M11.2 Anzahl der in den letzten 3, 6 und 12 Monaten hinzugefügten „Lines of Code“ in den Versionsverwaltungssystemen M11.3 Aktivität auf den Mailinglisten des Projektes

Validierung der Zerlegung

- Fallstudie mit 6 Open Source Web Content Management Systemen
 - Drupal
 - Magnolia (\$)
 - Mambo
 - OpenCms (\$)
 - TGS CMS
 - Typo3
- Datenerhebung durch 3 Daten- und Informationsquellen
 - Projektwebseiten
 - CVS/SVN-Daten
 - Befragung der Projektadministratoren

Ergebnisse – Kommerzielle Ausrichtung

Die kommerzielle Ausrichtung von Open Source Projekten beeinflusst die zukünftigen Herausforderungen

- Kommerzielle OSP
 - Finanzierung des Entwicklungsaufwandes
 - Gewinnung neuer Entwickler
- Nicht-kommerzielle OSP
 - (Finanzierung des Entwicklungsaufwandes)
 - Gewinnung neuer Entwickler
 - Wenige Kernentwickler
 - Wenige Benutzer
 - Schulung neuer Entwickler
 - Fähigkeiten neuer Entwickler
 - Koordination der Entwickler
 - Unterschiedliche Ziele der Entwickler
 - Motivation der Entwickler

Ergebnisse - Schulungsprozess

- Direkt quellcodeorientierte Hilfestellungen sind ein wichtiger Bestandteil des Schulungsprozesses in nicht-kommerziellen Open Source Projekten
 - Drupal ist das Einzige der vier nicht-kommerziellen OSP, das in der Ausbildung und den Fähigkeiten neuer Entwickler ein zukünftiges Problem sieht
 - Drupal ist das Einzige der vier nicht-kommerziellen OSP, in dem weder Codierrichtlinien implementiert sind noch ein Feedback zu eingehenden Kontributionen erfolgt

	<i>Drupal</i>	<i>Mambo</i>	<i>TGS CMS</i>	<i>Typo3</i>
<i>Inv.Zt</i>	3-5	3-5	>10	2
<i>API</i>	ja	ja	nein	ja
<i>B.T.</i>	ja	ja	ja	ja
<i>T.F.</i>	ja	ja	nein	ja
<i>E.R.</i>	ja	nein	nein	ja
<i>N.K.</i>	nein	nein	nein	ja
<i>FAQ</i>	nein	ja	ja	nein
<i>Cod.R.</i>	nein	ja	ja	ja
<i>F.K.</i>	nein	ja	ja	ja
<i>Tut.</i>	nein	ja	nein	nein
<i>Com.G.</i>	nein	nein	nein	ja

<i>Inv.Zt.</i>	= Wöchentlich investierte Zeit der Projektadministratoren
<i>API</i>	= API-Spezifikation
<i>B.T.</i>	= Bug-Tracker
<i>T.F.</i>	= Beantwortung technischer Fragen
<i>E.R.</i>	= Beantwortung von Fragen zur Entwicklungsrichtung
<i>N.K.</i>	= Beantwortung von Fragen zu neuen Beiträgen
<i>FAQ</i>	= Frequently Asked Questions
<i>Cod.R.</i>	= Codierrichtlinien
<i>F.K.</i>	= Feedback zu eingegangenen Codierbeiträgen
<i>Tut.</i>	= Tutorials
<i>Com.G.</i>	= Committer Guide

Ergebnisse - Projektgröße

<i>Open Source Projekt</i>	<i>Lines of Code</i>	<i># Entwickler</i>
<i>TGS CMS</i>	99.165	3
<i>Mambo</i>	261.516	15
<i>Typo3</i>	389.242	25
<i>OpenCms</i>	678.326	25
<i>Magnolia</i>	218.706	16
<i>Magnolia-Core</i>	36.582	8
<i>Drupal</i>	59.131	8

- Bestehende Korrelation zwischen der Anzahl der Quellcodezeile und der Anzahl der insgesamt beteiligten Entwickler
- Mehr Entwickler produzieren mehr Quellcode oder mehr Quellcode lockt mehr Entwickler an?

Ergebnisse - Projektaktivität

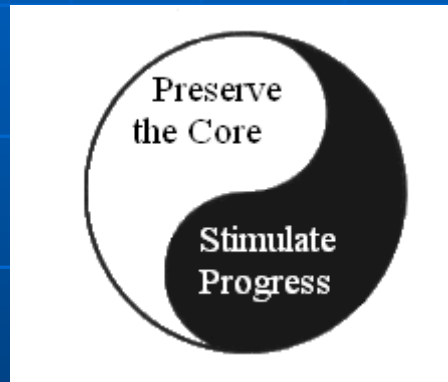
- Der Quotient aus hinzugefügten und geänderten Quellcodezeilen in den letzten drei Monaten korreliert mit der zukünftigen Reife der Software
- 3 der 6 beteiligten Open Source Projekte signalisieren, dass ihre Software in fünf Jahren „mature“ sein wird. Diese drei Projekte besitzen gleichzeitig den höchsten Quotienten aus hinzugefügten und geänderten Quellcodezeilen in den letzten drei Monaten

	<i>LOC changed</i>	<i>LOC added</i>	<i>changed/added</i>	<i>Zuk.SW.Qualität</i>
<i>Drupal</i>	7.841	1.611	4,87	mature
<i>Magnolia</i>	4.695	1.226	3,83	higher/mature
<i>Mambo</i>	84.050	18.652	4,51	mature
<i>OpenCms</i>	9.331	7.341	1,27	higher
<i>TGS CMS</i>	58.674	54.307	1,08	higher
<i>Typo3</i>	11.131	6.767	1,64	higher

Ergebnisse - Kernideologie

Projektideologie/Projektziele

- Rahmenwerk von Collins und Porras für visionäre Unternehmen



- Konservative, orientierende Maßnahmen
 - Kernideologie = Kernwerte + Daseinsberechtigung
- Fortschrittsorientierte Maßnahmen
 - Ambitionierte Ziele

Wichtig: Existenz + Maßnahmen zur Umsetzung

Ergebnisse - Kernideologie

Beispiel: Magnolia

Aber:

- Datenerhebung ergab nur bei 2 der 6 Open Source Projekte eine zu Collins und Porras vergleichbare Formulierung von Kernwerten auf der Webseite
- Zudem teilweise auseinander gehende Angaben zwischen den Webseiten und den Antworten der Projektadministratoren
- => Für eine genauere Analyse der Existenz und Umsetzung einer Kernideologie sind die Datenquellen unzureichend
- Ambitionierte Ziele werden eher formuliert, aber auch hier ist der Grad der Umsetzung mit den drei Datenquellen nicht ermittelbar

Simple is beautiful

At Magnolia, we focus on providing an outstanding user experience that encompasses all of the product's lifecycle, not just a nice GUI. Your content can be published to a large audience, your business tools can be integrated easily and support is just a phone call (or email) away. The result: no worries. Magnolia. Simple Enterprise Content Management.

Easy-to-use

Everybody claims their software is easy-to-use, so with all other benefits Magnolia offers you might wonder why we mention it at all. If it were only for the nice words, easy-to-use would be a non-feature. But at Magnolia we know that nice icons are no substitute for rigorous user interaction design. We understand that ease-of-use offers serious business value, and so do our customers.

Extensible

Magnolia is easily extended through a number of innovative features - a packaging module allows the easy deployment of any modifications you have made, a module concept lets you add functionality without touching the source code of Magnolia, and the easy templating mechanisms allow you to simply extend Magnolia even without writing java code.

Battle-tested

Magnolia started development in early 2003, and has seen more than 160'000 downloads. Thousands of websites around the world are running Magnolia today and enjoy our early investment in the technologies relevant today.

Scalable

Magnolia has been designed for enterprise requirements - with its low memory footprint, a high-performance cache and built-in clustering capabilities, Magnolia shines where others crumble, and makes high-performance websites simple.

Standards-Based

Magnolia is based on standards so that its easy for you to exchange data, to implement custom additions and to hire personell that already knows how to work with the technologies we use.

Commercial Support Availability

Magnolia Enterprise Edition is available with the support option of your choice. Companies around the world enjoy the peace of mind that a support contract with us provides.

Open Source

Developers love open source products because it makes modifications easy. Enterprises love open source because it provides security. There is no expensive "vendor-only" consulting needed to make Magnolia do what you want (but we happily provide support if you need it). Magnolia has been Open Source from day one. Source availability has many advantages amply discussed elsewhere, so we will not repeat them here. Source-code availability eases integration tasks a great deal, and integration is a major part of building value-added websites. So what better way to add value than to provide you with the sources?

Erkenntnisse und Ausblick

Erkenntnisse der 1. Iterationsstufe

- Die kommerzielle Ausrichtung und ein Schulungsprozess für Entwickler und Benutzer beeinflussen die Zukunftssicherheit von OSP
- Ein hohes Verhältnis von geänderten zu hinzugefügten Quellcodezeilen signalisiert die zukünftige Reife einer Projektsoftware

Erweiterungen für die 2. Iterationsstufe

- Untersuchung der Mailinglisten auf die Existenz und den Grad der Verwurzelung einer Kernideologie nach Collins und Porras
- Untersuchung von „Marketingmaßnahmen“

Danke