
Studie Open-Commons-Region Linz

Fakten, Perspektiven, Maßnahmen

AP 5

Kriterienkatalog zur Identifikation
von Open-Source-Einsatzgebieten

Inhaltsverzeichnis

1	Einleitung und Überblick.....	3
2	Kriterienkataloge in der Literatur.....	4
2.1	Open Business Readiness Rating Model (OpenBRR)	4
2.1.1	Vorgehensweise	4
2.1.2	OpenBRR Kriterienkatalog	5
2.1.3	Diskussion des Ansatzes	6
2.2	Qualification and Selection Open Source Model (QSOS)	7
2.2.1	Vorgehensweise	8
2.2.2	QSOS Kriterienkatalog	9
2.2.3	Diskussion des Ansatzes	10
2.3	Qualipso Trustworthiness Model	11
2.3.1	Vorgehensweise	11
2.3.2	Qualipso Kriterienkatalog	12
2.3.3	Diskussion des Ansatzes	14
3	Kriterienkatalog für Potenzialanalyse	15
3.1	Themenbereich Kosten	15
3.2	Themenbereich Normen.....	16
3.3	Themenbereich Support.....	17
3.4	Themenbereich Community	17
3.5	Themenbereich Politische Aspekte	18
3.6	Themenbereich Verfügbarkeit	18
4	Kriterienkatalog für OS-Software Auswahl	19
4.1	Themenbereich Funktionalität	19
4.2	Themenbereich Kosten	20
4.3	Themenbereich Nutzen.....	21
4.4	Themenbereich Rechtliche Aspekte	22
4.5	Themenbereich Dokumentation.....	23
4.6	Themenbereich Service	26
4.7	Themenbereich Produktreife	29
4.8	Themenbereich Interne Qualität.....	31
4.9	Themenbereich Benutzbarkeit.....	33
4.10	Themenbereich Integrierbarkeit	35
5	Entscheidungsprozesse	36
5.1	Stage-Gate Modell nach Cooper	36
5.2	Entscheidungsmodell für Potentialanalyse und Softwareauswahl	37
5.2.1	Durchführung der Potentialanalyse (Gate 0)	37
5.2.2	Vorbereitungsarbeiten für die Bewertung in den Gates 1 und 2	38
5.2.3	Bewertung der Kriterien von Gate 1	39
5.2.4	Bewertung der Kriterien von Gate 2	39
5.2.5	Interpretation der Ergebnisse	39

1 Einleitung und Überblick

Dieses Kapitel beschreibt einen Kriterienkatalog, der für folgende typische Aktivitäten im Rahmen einer OS-Region eingesetzt werden kann:

- Feststellen, ob OS für einen bestimmten fachlichen Bereich einer OS-Region nutzvoll eingesetzt werden kann, d.h. über ausreichend Potential verfügt. OS ist dabei ganz bewusst nicht auf Software eingeschränkt, sondern gilt gleichermaßen auch für OS-Content, etc.
- Bewertung von OS-Alternativen. Dabei liegt der Schwerpunkt auf Auswahlentscheidungen für alternative OS-Software Lösungen.

Ausgangspunkt der Arbeit ist eine systematische Analyse von Kriterienkatalogen, wie sie in der Literatur zu finden sind. Allgemeine Kriterienkataloge für die Auswahl von Software sind in unterschiedlichem Detaillierungsgrad und in unterschiedlicher Qualität verfügbar. Bei der Analyse der Literatur haben wir uns ausschließlich auf Kriterienkataloge konzentriert, die explizit für die Auswahl von OS-Software konzipiert wurden. Diese Kriterienkataloge werden in Kapitel 2 beschrieben.

Kapitel 3 beschreibt den Kriterienkatalog für die Ermittlung des Potentials von OS-Lösungen, gegliedert nach unterschiedlichen Themenbereichen.

Kapitel 4 beschreibt den Kriterienkatalog für den Vergleich von OS Software Alternativen, wiederum gegliedert nach unterschiedlichen Themenbereichen.

Kriterienkataloge sind ein wichtiger Bestandteil für die Ermittlung des Potentials bzw. für die Auswahl von OS-Alternativen. Es ist definitiv kein Ziel, eine umfassende Methode für die Anwendung des Kriterienkatalogs zu entwickeln. Da es jedoch Wechselwirkungen zwischen Methode und Kriterienkatalog gibt, gehen wir bei der Konzeption des Kriterienkatalogs von einem Stufen-Entscheidungsprozess im Stile eines Stage-Gate Prozesses [Cooper 2001] aus. Dieser Stage-Gate Prozess ist allgemein genug, um individuelle Entscheidungsprozesse abbilden zu können, legt aber doch einen Rahmen für das Treffen von Entscheidungen (Gates) fest, der für die Modellierung des Kriterienkatalogs von Bedeutung ist.

In Kapitel 5 beschreiben wir die Grundlagen von Stage-Gate Prozessen allgemein sowie deren Adaption für die Ermittlung des OS-Potenzials bzw. für die Auswahl von OS-Alternativen – also ganz im Sinne der Zielsetzung dieser Arbeit.

2 Kriterienkataloge in der Literatur

Wie schon in der Einleitung erwähnt, gibt es eine Reihe von Kriterienkatalog für die Auswahl von alternativen Softwarepaketen ganz allgemein oder für die Auswahl von Software in einer bestimmten Domäne. Eine Beschränkung auf eine bestimmte Domäne ist nicht möglich, da die möglichen Einsatzgebiete für eine OS-Region zu vielfältig sind. Wir haben uns bei der Analyse entschieden, den Schwerpunkt nur auf jene publizierten Kriterienkataloge zu legen, deren Fokus die Beurteilung von OS-Software ist. Es wurden drei Ansätze identifiziert (*OpenBRR*, *QSOS* und das *Qualipso Trustworthiness Modell*) die in den nachfolgenden Kapiteln beschrieben werden. Die Beschreibung erfolgt dabei immer nach dem Schema, dass zuerst die Vorgehensweise bei der Bewertung der Software dargestellt wird und dann der vorgeschlagene Kriterienkatalog. Danach erfolgt eine Darstellung der Vor- und Nachteile.

2.1 Open Business Readiness Rating Model (OpenBRR)

In diesem Kapitel wird das Open Business Readiness Rating Modell, im Folgenden als *OpenBRR-Modell* bezeichnet, vorgestellt. OpenBRR soll sich als neuer Standard für die Beurteilung von OS-Software etablieren. Ziel von OpenBRR [OpenBRR 2009]:

“The ultimate goal of BRR is to give companies a trusted, unbiased source for determining whether the open source software they are considering is mature enough to adopt. It will help adopters assess which open source software is best suited to their needs and enable them to share findings with the community. It promotes use and adoption of open source software and may assist developers in creating and delivering software geared to enterprise use.”

2.1.1 Vorgehensweise

Die Vorgehensweise des *OpenBRR-Modells* gliedert sich in die Stufen Vorselektion möglicher OS-Produkte, Kriterien anpassen, Kriterienkategorien gewichten und Evaluierung der Produkte. Diese vier Stufen werden im Folgenden näher beschrieben [Deprez und Alexandre 2008]:

- *Vorselektierung von OS-Produkten*: In diesem Schritt wird von einer Liste mit OS-Produkten ausgegangen, die prinzipielle Anforderungen und Kriterien erfüllen müssen. Das *OpenBRR-Modell* gibt acht Kriterien für die Vorselektion vor. Darüber hinaus können weitere Kriterien definiert werden. Es wird auch angemerkt, dass die Anforderungen und Kriterien durch die Produkte oder Systeme bestimmt werden, die ein OS-Produkt integrieren wollen. Welches Bewertungsmodell (k.o. Kriterien, Skalenwerte, etc.) bei der Vorselektion zur Anwendung kommen soll, bleibt offen. Am Ende der Vorselektion sollte eine kurze Liste mit Produkten überbleiben, die in die engere Auswahl kommen und weiter analysiert werden.
- *Kriterien anpassen*: In diesem Schritt werden die vom *OpenBRR-Modell* vorgeschlagenen Kriterien geprüft und - abgestimmt auf das jeweilige Auswahlproblem - ausgewählt. Die

Methode lässt es zu, neue Kriterien aufzunehmen und diese dann zu bewerten. Ergebnis dieser Aktivität ist ein Evaluierungsbogen, der auf den Kontext des Unternehmens und des Produktes, das ausgewählt werden soll, angepasst ist.

- *Kriterienkategorien gewichten:* Für jedes zu bewertende OS-Produkt werden die Rohdaten erfasst, um die definierten Kriterien später bewerten zu können. Des Weiteren wird ein Gewichtungsfaktor für jedes Kriterium festgelegt.
- *Evaluierung der Produkte:* In diesem Schritt wird der Erfüllungsgrad aller Kriterien festgestellt. Der Erfüllungsgrad wird anhand einer ordinalen Skala (1 bis 5) bestimmt, wobei 1 sehr schlecht und 5 sehr gut ist. Siehe dazu beispielhaft Abbildung 1 wo anhand von drei Kriterien gezeigt wird, welche Bedingungen jeweils erfüllt sein müssen, um einen bestimmten Erfüllungsgrad des Kriteriums zu erreichen. Der Erfüllungsgrad bestimmt gleichzeitig die Punktezahl, die ein bestimmtes OS-Projekt für ein Kriterium erreicht. Die Summe der erreichten Punkte (unter Berücksichtigung der Gewichtungsfaktoren) ergibt dann eine Gesamtpunktezahl für jedes zu evaluierende Softwareprodukt. Jenes Produkt mit den meisten Punkten erhält den Zuschlag.

Criteria	1	2	3	4	5
<i>Time for vanilla installation</i>	> 4 hours	1-4 hours	30min to 1 hours	10-30 minutes	≤ 10 minutes
<i>User Contribution Framework</i>	Users cannot contribute		Users are allowed to contribute		Users are allowed to contribute and contribution are edited / filtered by experts
<i>Reference Deployment</i>	No		Yes		Yes, with publication of user's size

Abbildung 1: Mögliches OpenBRR Bewertungsschema für drei mögliche Kriterien [Deprez und Alexandre 2008, S. 6]

2.1.2 OpenBRR Kriterienkatalog

OpenBRR stellt eine Liste von Kriterien für die Vorselektion und für die eigentliche Evaluierung zur Verfügung. Bei der Vorselektion schlägt das *OpenBRR-Modell* vor, sich auf acht Kriterien zu konzentrieren und abhängig vom Kontext anzupassen [Deprez und Alexandre 2008].

- *Licensing / Legal:* Welche Lizenzbestimmungen kommen zur Anwendung.
- *Standards compliance:* Für bestimmte Softwarekategorien müssen Standards erfüllt werden; so das für die betrachtete Domäne gilt, muss die Erfüllung dieser Standards geprüft werden.
- *Referenceable adopters:* Beurteilung der Erfahrungen, die andere Organisationen oder Firmen mit der jeweiligen OS-Software gemacht haben.
- *Availability of support:* Prüfung der Verfügbarkeit von Support-Infrastrukturen und von typischen Reaktionszeiten.
- *Implementation language(s):* Prüfung der Konformität der verwendeten Implementierungssprachen und Technologien im Hinblick auf die für die in der Organisation geltenden Richtlinien für Softwareentwicklung.
- *Third party reviews:* Sind Reviews vorhanden, die von Drittpersonen durchgeführt wurden?

- *Books*: Verfügbarkeit von öffentlichem zugänglichem Informationsmaterial, wie Bücher, Tutorials, white-papers
- Review by industry analysts: Berücksichtigung von Ergebnissen systematischer Reviews, wobei die Qualität und Seriosität der untersuchenden Organisation bei der Bewertung der Review-Ergebnisse zu berücksichtigen ist (z.B. Review von Analysten der Gartner Group sind als vertrauenswürdiger einzustufen, als anonyme Reviews, die im Internet veröffentlicht werden).

Für die Vorselektion ist seitens OpenBRR nicht definiert, wie diese zu erfolgen hat.

Der von *OpenBRR* nach der Vorselektion zu verwendete Kriterienkatalog ist überblicksartig in Abbildung 2 dargestellt.

2.1.3 Diskussion des Ansatzes

Folgende Vorteile wurden in der Literatur dazu gefunden [Deprez und Alexandre 2008, S. 15]:

- Es sind eine Reihe von Kriterien – sowohl für die Vorselektion, als auch dann für die tatsächliche Auswahl – vorgegeben; das Modell sieht jedoch auch vor, dass individuelle Anpassungen vorgenommen werden müssen.
- Das prinzipielle Bewertungsverfahren für die Auswahl von Alternativen ist vorgegeben und gut an individuelle Bedürfnisse anpassbar; die exakte Auswahlmethodik bei der Vorselektion ist nicht definiert.
- 5-stufige Bewertungsskala.
- 2-stufiges Bewertungsverfahren.

Folgende Nachteile wurden in der Literatur zu diesem Modell gefunden [Deprez und Alexandre 2008, S. 15]:

- Es sind keine Daten von durchgeführten Evaluierungen verfügbar.
- Der Kriterienkataloge steht nicht in unterschiedlichen Varianten oder Versionen zur Verfügung.
- 5-stufige Bewertungsskala nicht bei allen Kriterien angewandt – siehe beispielsweise Abbildung 1 beim Kriterium „*User contribution framework*“.
- Keine eindeutigen Bezeichnungen der Kriterien-Kategorien.

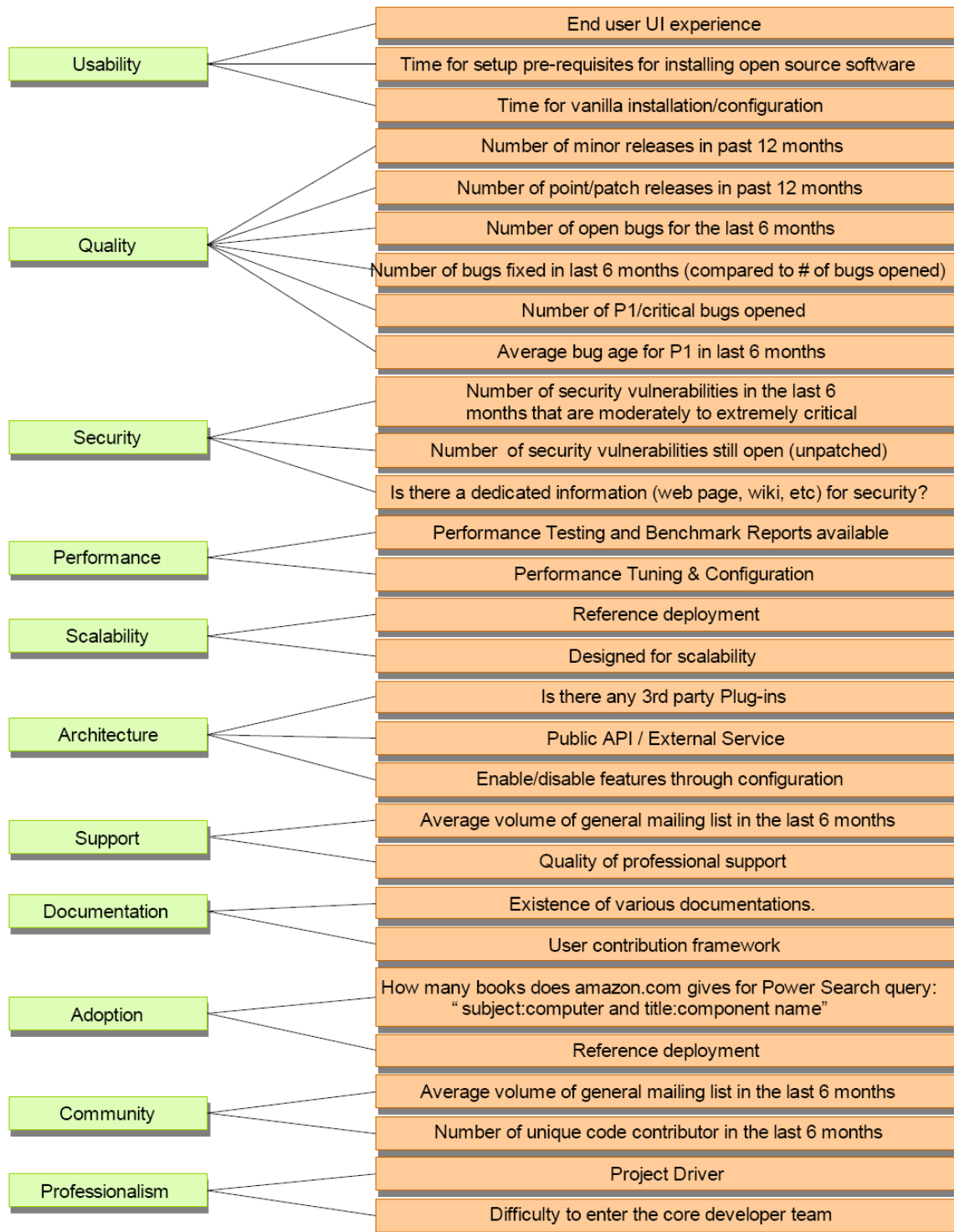


Abbildung 2: Hierarchie von Evaluierungskriterien [Deprez und Alexandre 2008, S. 5]

2.2 Qualification and Selection Open Source Model (QSOS)

In diesem Kapitel wird das Qualification and Selection Open Source-Modell, kurz *QSOS-Modell* in der Version 1.6 zur Evaluierung von OS-Software vorgestellt werden. Kurzcharakterisierung der Methode:

“QSOS is a method, designed to qualify, select and compare free and open source software in an objective, traceable and argued way. It is publicly available under the terms of the GNU Free Documentation License.”

2.2.1 Vorgehensweise

Die Vorgehensweise gliedert sich im Wesentlichen in folgende 4 Schritte:

- *Definition:* Vorbereiten der Evaluierung – Ermittlung grundlegender Daten wie Domäne der Software, Lizenzmodelle, Community, etc.
- *Evaluierung:* Evaluierung der Software (Ermittlung der Daten) entlang der Kriterienachsen Funktionalität, Risiken für Benutzer und Risiken für den Bereitsteller der IT-Dienstleistung
- *Qualifikation:* Gewichtung von Kriterien
- *Selektion:* Bewertung und Selektion von Produkten

Etwas detaillierter für die Bereiche Evaluierung, Qualifikation und Selektion ergibt sich folgende Vorgehensweise:

1. *Erstellen einer Liste von OS-Lösungen*, welche oberflächlich betrachtet die wesentlichen Anforderungen erfüllen. Hierbei werden keine Kriterien oder Themenschwerpunkte seitens der Autoren erwähnt, welche als Ausgangspunkt für die Auswahl möglicher Lösungen sein könnten. Die Anforderungen, welche das Produkt mindestens erfüllen sollte, werden durch den Nutzer oder Produktintegrator vorgegeben.
2. *Evaluierung* der im Schritt 1 ausgewählten OS-Lösungen anhand der Kriterien, welche in der Spezifikation des QSOS-Modells vorgegeben sind. In diesem Schritt erfolgt die Ermittlung der Daten für jede der im ersten Schritt ausgewählten OS-Lösung anhand des vom *QSOS-Modell* definierten Kriterienkatalogs, indem der Erfüllungsgrad für jedes Kriterium (und natürlich jedes zu untersuchenden Softwareproduktes) festgestellt wird. Der Erfüllungsgrad wird anhand einer ordinalen Skala (0 bis 2) bestimmt, wobei 0 sehr schlecht und 2 sehr gut ist. Siehe dazu beispielhaft Abbildung 3 , wo anhand von drei Kriterien gezeigt ist, welche Bedingungen jeweils erfüllt sein müssen, um einen bestimmten Erfüllungsgrad des Kriteriums zu erreichen.

Criteria	Score = 0	Score = 1	Score = 2
<i>Age</i>	Less than 3 month old	Between 3 month old and 3 year old	More than 3 year old
<i>Training</i>	No offer of training identified	Offer exists but is restricted geographically and to one language or is provided by a single contractor	Rich offer provided by several contractors, in several languages and split into modules of gradual levels
<i>Source Code Quality</i>	Not very readable code or of poor quality, incoherence in coding styles	Readable but not really commented in details	Readable and commented code implementing classic design patterns with a coherent and applied coding policy

Abbildung 3: Mögliche Klasseneinteilung des QSOS-Modells [Deprez und Alexandre 2008, S. 4]

3. *Gewichtung der Kriterien:* In diesem Schritt besteht die Möglichkeit, für jedes Kriterium die Wichtigkeit durch einen Gewichtungsfaktor zum Ausdruck zu bringen. Vorschläge, wie eine solche Gewichtung aussehen könnte, werden seitens der Autoren nicht gemacht, d.h., es obliegt zur Gänze dem Bewerter, wie die Gewichtung vorzunehmen ist.
4. *Berechnung der Gesamtpunkteanzahl* für jede OS-Lösung anhand der Bewertung aus dem Schritt 2 und der Gewichtung aus dem Schritt 3. In diesem Schritt erfolgt die Berechnung der Gesamtpunktezahl für jedes OS-Produkt, indem für jedes Kriterium der Bewertungsfaktor und der Gewichtungsfaktor mit einander multipliziert und eine Gesamtpunktesumme errechnet wird. Als Ergebnis ergibt sie eine Reihenfolge, welche eine Aussage darüber zulässt, welche Produkte am meisten den Anforderungen entsprechen.
5. *Testen der Top-Favoriten:* In diesem letzten Schritt wird empfohlen, dass die Top-Favoriten getestet werden sollen. Die Anzahl der Favoriten kann dabei, anhängig vom Grad der kritischen Betrachtung, zwischen zwei oder mehr Kandidaten variieren.

2.2.2 QSOS Kriterienkatalog

Der Kriterienkatalog des QSOS-Modells gliedert den Kriterienkatalog in drei Hierarchieebenen und lässt sich in einen generellen Teil und einen speziellen Teil unterscheiden. Der generelle Teil (siehe dazu Abbildung 4) beinhaltet Themenbereiche und zugehörige Kriterien, welche auf jedes OS-Produkt zutreffen, unabhängig davon, ob es sich beispielsweise um ein CMS oder um eine Datenbank handelt. Der spezifische Teil beinhaltet nur den Themenbereich „Funktionalität“ und wird von den Autoren lediglich unterteilt in „required functionality“, „optimal functionality“ und „not required functionality“. Da die Funktionalität abhängig ist von der Art der zu beurteilenden Software, z.B. CMS, Datenbank, etc. erfolgt keine konkrete Unterteilung in weitere Kriterien. In wieweit eine Funktion im weiteren Sinne einer der drei Unterteilungen zugeordnet wird, erfolgt mittels eines Gewichtungsfaktors [Deprez und Alexandre 2008, S. 1], [QSOSORG, S. 32].

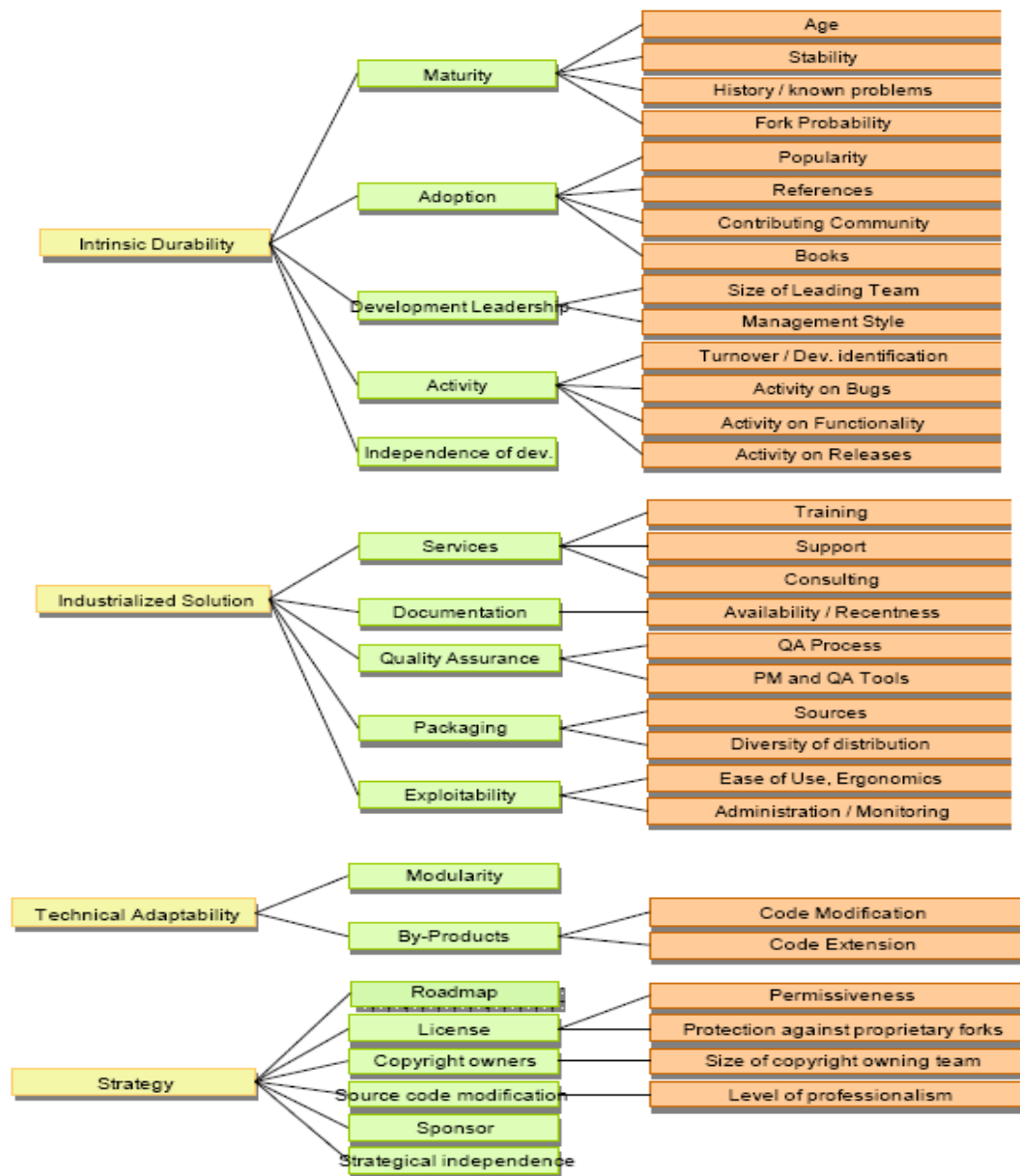


Abbildung 4: generelle Kriterien des QSOS-Modells [DEPREZ UND ALEXANDRE 2008, S. 3]

2.2.3 Diskussion des Ansatzes

In der Literatur werden folgende Vorteile des QSOS-Modells angeführt [Deprez und Alexandre 2008, S. 15]:

- Es existieren zahlreiche, durchgeführte Evaluierungen.
- QSOS ist in unterschiedlichen Versionen vorhanden (Version 1.6 ist die aktuelle Version).
- Der Kriterienkatalog ist umfangreich.
- Die Kriterienbenennung ist aussagekräftig.

Als Schwachstellen werden folgende Punkte genannt [Deprez und Alexandre 2008, S. 15]:

- Die 3-stufige Bewertungsskala erschwert die Entscheidung.
- Die Bewertungsausprägung ist schwach formalisiert.
- Die Bewertungsskala ist nicht universell auf alle Kriterien anwendbar.

2.3 Qualipso Trustworthiness Model

Qualipso (Quality Standard for Open Source Software) ist eine von der Europäischen Kommission unterstützte Initiative, die die Qualität von OS-Software sicherstellen und in weiterer Folge das Vertrauen in OS-Software steigern soll. Die Initiative wurde gegründet, um jene Faktoren zu beseitigen, welche die Akzeptanz von OS-Software senken bzw. deren Einsatz verhindern. [Fraunhofer 2009]

2.3.1 Vorgehensweise

Laut [Fraunhofer 2009] hat Qualipso folgende Ziele:

- Definition von Geschäftsmodellen, die den Einsatz von OS-Software erleichtern.
- Vereinfachung der juristischen Rahmenbedingungen bei OS-Software.
- Interoperabilität zwischen unterschiedlichen OS-Software Produkten bzw. zwischen OS-Software und kommerzieller Software fördern.
- Beispiele für das Informationsmanagement bei der Entwicklung von OS-Software implementieren.
- Definition von Qualitätsprozessen.
- Aufbau einer standardisierten Entwicklungsplattform.
- Gründung von Kompetenzzentren zur Unterstützung von OS-Software.

Qualipso verwendet zur Erhöhung der Vertrauenswürdigkeit den GQM (goal/question/metrics) Ansatz [Basili 1988]. Dabei werden zuerst ein oder mehrere Ziele festgelegt, welche erreicht werden sollen. Im Rahmen des Qualipso Projektes ist dabei das primäre Ziel die Erhöhung der Vertrauenswürdigkeit von OS-Produkten.

Jedes Ziel wird in weiterer Folge mit einem so genannten „Abstraction Sheet“ verbunden. Ein Abstraction Sheet setzt sich aus folgenden Bestandteilen zusammen:

- *Quality focus (QF)*: Stellt detaillierte Informationen über die Eigenschaften des zu betrachtenden Objekts bzw. der zu betrachtenden Software zur Verfügung.
- *Variation factors (VF)*: Produkt- und prozessspezifische Eigenschaften, die den QF beeinflussen könnten.
- *Baseline hypotheses (BH)*: Beschreiben den aktuellen Status des zu betrachtenden Objekts in Bezug auf den QF.
- *Impact on baseline hypotheses (IoBH)*: Beschreibt wie die VF den Status des Objekts beeinflussen könnten. [Bianco et al. 2008]

Ausgehend vom Abstraction Sheet werden eine Reihe von Fragen extrahiert, welche beantwortet werden müssen um feststellen zu können, ob die in Schritt 1 gestellten Ziele erreicht bzw. in welchen Ausmaß sie erfüllt wurden. Die Fragen aus diesem Schritt sind detaillierter als das Abstraction Sheet. [Bianco et al. 2008]

Zuletzt wird aus den Fragen eine Reihe von Metriken abgeleitet. Diese werden verwendet um die nötigen Daten bezüglich der Beantwortung der zuvor gestellten Fragen erheben zu können. Abbildung 5 zeigt nochmals überblicksweise den Aufbau eines GQM- Prozesses.

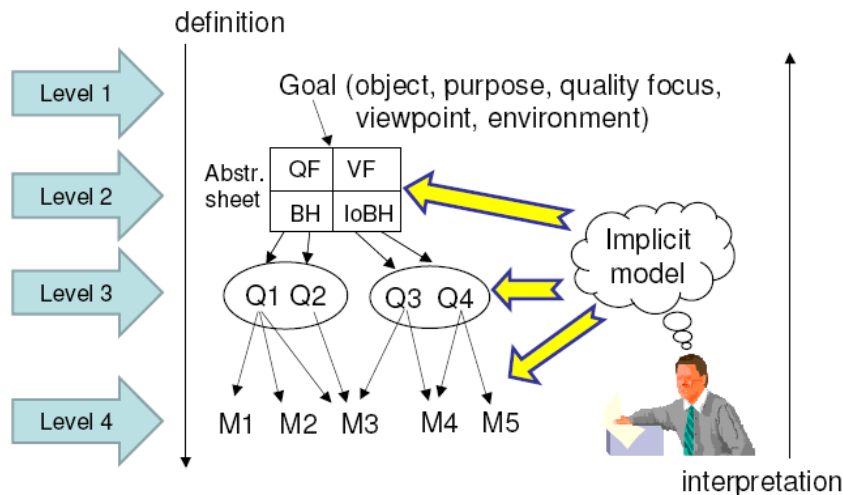


Abbildung 5: Qualipso GQM Prozess [Bianco et al. 2008]

2.3.2 Qualipso Kriterienkatalog

Um die Vertrauenswürdigkeit von OS-Software bestimmen zu können, legten [Bianco et al. 2008] folgende Bereiche fest:

- As-is utility
- Exploitability in development
- Functionality
- Interoperability
- Reliability
- Security
- Performance
- Cost effectiveness
- Customer satisfaction
- Developer quality

Ausgehend von diesen Bereichen haben [Bianco et al. 2008] diese weiter verfeinert. Dabei haben sie die Bereiche, die die Vertrauenswürdigkeit beeinflussen in Eigenschaften des Produkts und in Eigenschaften des Prozesses unterteilt. Die dabei festgelegten produktspezifischen Eigenschaften bzw. Kriterien, welche die Vertrauenswürdigkeit von OS-Software beeinflussen, werden in Abbildung 6 dargestellt:

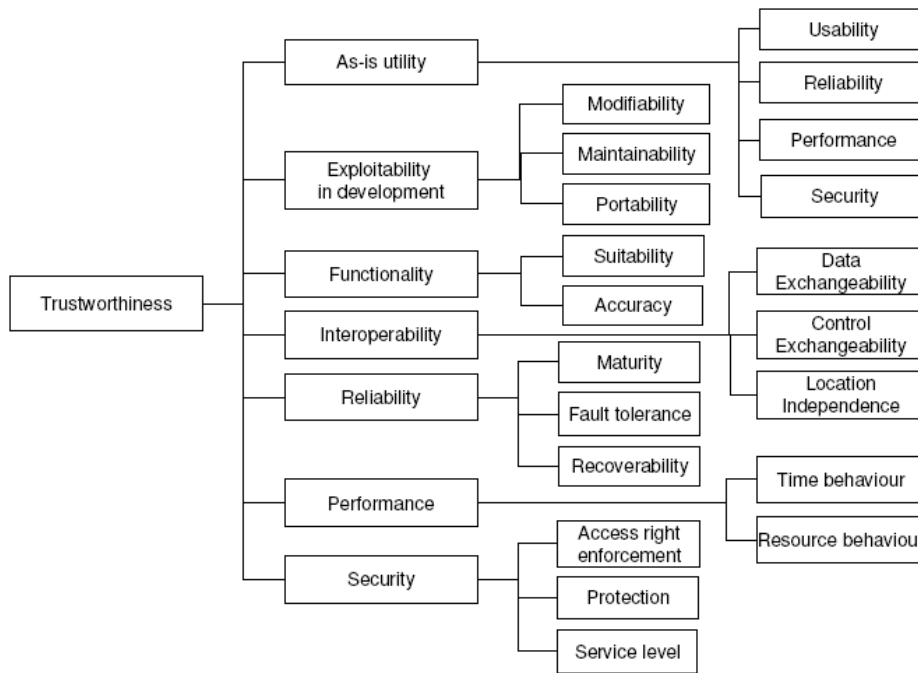


Abbildung 6: Produktspezifische Dimensionen von Vertrauenswürdigkeit [Bianco et al. 2008]

Abbildung 7 gibt einen Überblick über die von Qualipso festgelegten prozessspezifischen Kriterien von OS-Software:

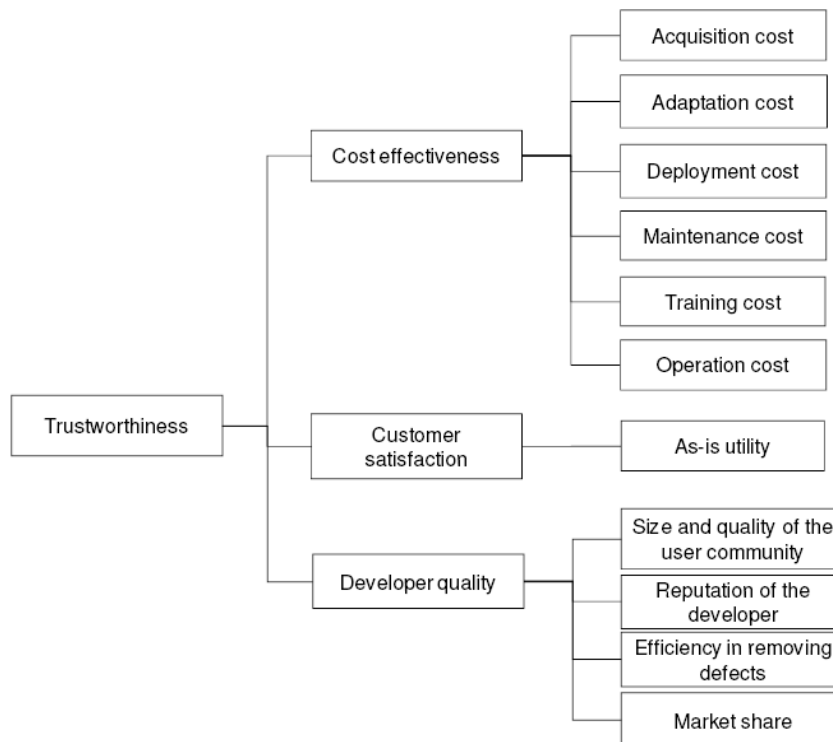


Abbildung 7: Prozessspezifische Dimensionen von Vertrauenswürdigkeit [Bianco et al. 2008]

Der GQM-Ansatz von Qualipso ist in mehrere Ebenen untergliedert. Dabei sind zur Erreichung des Ziels (hier zur Erhöhung der Vertrauenswürdigkeit), mehrere Kriterien betroffen. Ein Kriterium ist unter anderem *As-Is Utility*, welches sich weiter untergliedert in *Usability*, *Reliability*, *Security* und *Performance*. Diese können wiederum untergliedert werden; beispielsweise hat die Benutzbarkeit

(*Usability*) einen Unterpunkt *Learnability*. Die Kriterien befinden sich auf der Ebene des quality focus. Zu den Kriterien werden in weiterer Folge, wie bereits zuvor beschrieben, Fragen gestellt. Eine solche wäre beispielsweise für den Punkt Erlernbarkeit, ob eine entsprechende Dokumentation für den Benutzer vorhanden ist. Um die Fragen beantworten zu können, werden auf der untersten Ebene die Metriken festgelegt, z.B.: ob die Benutzerdokumentation vollständig, lesbar und aktuell ist. Abbildung 8 zeigt nochmals zusammenfassend den zuvor beschriebenen Teilbaum des GQM Modells:

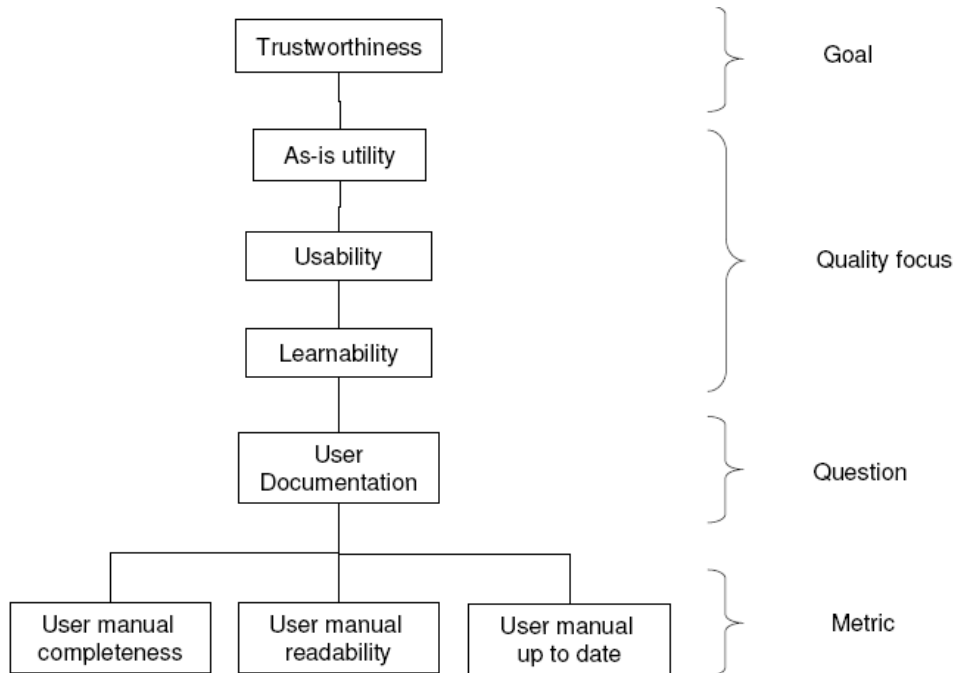


Abbildung 8: Qualipso GQM Prozess [Bianco et al. 2008]

2.3.3 Diskussion des Ansatzes

Die Qualipso Initiative konzentriert sich in erster Linie auf die systematische Entwicklung von zielgerichteten Kriterien für die Bewertung von OS-Software. Eine definierte Vorgehensweise für die Anwendung des Modells in der Praxis ist bis dato nicht öffentlich verfügbar. Der entwickelte Kriterienkatalog ist systematisch aufgebaut und kann auch teilweise für die Evaluierung von OS-Produkten zum Einsatz kommen.

3 Kriterienkatalog für Potenzialanalyse

Bei den Kriterien für die Potenzialanalyse wird die Bewertung auf einem hohen Abstraktionsgrad durchgeführt und kann meist nicht durch bestimmte Werte abgebildet werden. Ob ein Einsatzgebiet identifiziert werden kann, hängt von der subjektiven Betrachtung bzw. dem jeweiligen Anwendungsgebiet ab; z.B. kann der politische Wille stärker gewichtet sein als alle anderen (negativ) bewerteten Kriterien. Die Gliederung und Darstellung des Kriterienkatalogs erfolgt anhand der Themenbereiche *Kosten*, *Normen*, *Support*, *Politische Aspekte* und *Verfügbarkeit*.

Jedes Kriterium wird wie folgt beschrieben:

- Themenbereich: Jedes Kriterium ist einem Themenbereich zugeordnet.
- Kriterium: Bezeichnung des Kriteriums.
- Beschreibung: Beschreibung des Kriteriums.
- Metrik: Wie wird das Kriterium gemessen inkl. Bewertung.
- Anmerkung: Optionale Anmerkungen für eine detailliertere Beschreibung.

Es gibt für die Bewertung der Metriken verschiedene Skalen:

- 2-stufige Bewertungsskala: Bei Bewertungsergebnisse mit genau 2 Möglichkeiten.
- 4-stufige Bewertungsskala: Bei Bewertungsergebnisse mit 4 Abstufungen.

3.1 Themenbereich Kosten

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Investitionskosten
<i>Beschreibung:</i>	Als Investitionskosten für eine OS-Lösung sind in erster Linie die Migrationskosten (Schulungskosten, Integrationskosten), zu berücksichtigen.
<i>Metrik:</i>	<i>Sind die Investitionskosten, die durch den Einsatz der OS-Software bzw. OS-Content entstehen niedriger als von kommerzieller Software/Content?</i> JA oder NEIN
<i>Anmerkung:</i>	Eine detaillierte Analyse der anfallenden Kosten ist in dieser frühen Entscheidungsphase weder sinnvoll noch möglich. Gerade die Integrationskosten sind jedoch grob abzuschätzen, vor allem dann, wenn in der frühen Phase schon klar ist, dass Anpassungen und Erweiterungen vorzunehmen sind.

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Laufende Kosten
<i>Beschreibung:</i>	Die laufenden Kosten für eine OS-Lösung setzen sich aus Betriebskosten und Wartungskosten zusammen;
<i>Metrik:</i>	<i>Sind die laufenden Kosten, die durch den Einsatz der OS-Software bzw. OS-Content entstehen niedriger als von kommerzieller Software/Content?</i> JA oder NEIN
<i>Anmerkung:</i>	Eine detaillierte Analyse der anfallenden Kosten ist in dieser frühen Entscheidungsphase weder sinnvoll noch möglich.

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Kostensenkung
<i>Beschreibung:</i>	Reduktion der Kosten durch den Einsatz von Open Source.
<i>Metrik:</i>	<i>Kann durch Einsatz von OS-Software/Content eine Kosteneinsparung erzielt werden?</i> JA oder NEIN
<i>Anmerkung:</i>	Eine detaillierte Analyse des Kosteneinsparungspotentials ist in dieser frühen Entscheidungsphase weder sinnvoll noch möglich. Man muss sich hier auf grobe Einschätzungen verlassen.

3.2 Themenbereich Normen

<i>Themenbereich:</i>	Normen
<i>Kriterium:</i>	Rechtliche Aspekte
<i>Beschreibung:</i>	Rechtliche Aspekte können lokale, landesweite, bundesweite bzw. europäische Gesetzgebung betreffen.
<i>Metrik:</i>	<i>Darf bei den bestehenden rechtlichen Rahmenbedingungen in dem durch die gegebenen Anforderungen identifizierten Einsatzgebiet OS-Software bzw. OS-Content eingesetzt werden?</i> JA oder NEIN
<i>Anmerkung:</i>	In dieser Phase ist zu prüfen, ob beispielsweise eine Zertifizierung der Software, ein bestimmtes Gütesiegel, ein Gutachten einer unabhängigen Organisation etc. vorhanden sein muss. Daneben ist abzuschätzen, inwieweit es die zu erfüllenden Normen schwierig erscheinen lassen, dass diese durch eine OS-Lösung abgedeckt werden können (je detaillierter die gesetzlichen Regelungen sind, desto schwieriger wird dies prinzipiell sein).

<i>Themenbereich:</i>	Normen
<i>Kriterium:</i>	Lizenzen
<i>Beschreibung:</i>	Rechtliche/vertragliche Bedingungen für die Verwendung.
<i>Metrik:</i>	<i>Wie entspricht das Lizenzmodell den gewünschten Anforderungen?</i> Nicht passend → Unzureichend Ausreichend → Genügend Akzeptabel → Überdurchschnittlich Optimal → Hervorragend
<i>Anmerkung:</i>	Die Bewertung kann individuell und den jeweiligen Anforderungen entsprechend angepasst durchgeführt werden. Deswegen kann keine generelle Metrik festgelegt werden. Es mag auch als Indikator dienen, ob die jeweilige Lizenz der OS-Definition der OS-Initiative entspricht.

<i>Themenbereich:</i>	Normen
<i>Kriterium:</i>	Revisionsfähigkeit
<i>Beschreibung:</i>	Durch Revisionsfähigkeit wird garantiert, dass das Unternehmen den rechtlichen Vorgaben von externen Prüfstellen entspricht.
<i>Metrik:</i>	<i>Wird bei dem identifizierten Einsatzgebiet von offiziellen Prüfstellen eine Revisionsfähigkeit für OS-Software/Content bescheinigt?</i> JA oder NEIN

3.3 Themenbereich Support

<i>Themenbereich:</i>	Support
<i>Kriterium:</i>	Service Level Agreements
<i>Beschreibung:</i>	Ein Service Level Agreement bezeichnet die messbare Beschreibung einer zu erbringenden Dienstleistung, einschließlich der zu erreichenden Qualität und der anzuwendenden Messgrößen.
<i>Metrik:</i>	<i>Sind Service Level Agreements für die OS-Software bzw. OS-Content vorhanden?</i> JA oder NEIN

3.4 Themenbereich Community

<i>Themenbereich:</i>	Community
<i>Kriterium:</i>	Ausprägung der Community
<i>Beschreibung:</i>	Eine Community ist eine Gemeinschaft von Personen, die freiwillig, unentgeltlich Beiträge für die Nutzung von Open Source liefert.
<i>Metrik:</i>	<p><i>Falls noch keine Community existiert:</i> <i>Kann eine neue Community aufgebaut werden?</i> Nein, es kann keine Community aufgebaut werden → Unzureichend Ja, Community kann aufgebaut werden. Jedoch ist dies aufwändig. - > Genügend Ja, sie kann aufgebaut werden → Überdurchschnittlich Community kann leicht aufgebaut werden → Hervorragend</p> <p><i>Falls bereits eine Community existiert:</i> <i>Wie lange existiert sie bereits?</i> Bis 6 Monate → Unzureichend > 6 Monate → Genügend >1 Jahr → Überdurchschnittlich >3 Jahre → Hervorragend</p> <p><i>Größe der Community (Kann durch die Anzahl der Beiträge, Anzahl der aktiven Entwickler etc. gemessen werden)</i> Zu klein → Unzureichend Ausreichend → Genügend Groß → Überdurchschnittlich Sehr groß → Hervorragend</p> <p><i>Fachliche Kompetenz der Community (Kann durch Qualität der Beiträge in Foren oder dergleichen gemessen werden).</i> Unzureichend Genügend Überdurchschnittlich Hervorragend</p>
<i>Anmerkung:</i>	Es wird im Einzelfall zu entscheiden sein, welchen regionalen Bezug die Community hat bzw. haben muss. Diese Frage ist aber in diesem Kontext zu klären, da nur so eine Bewertung zuverlässig möglich ist. Es macht nämlich in der Bewertung einen Unterschied ob eine Community in der OS-Region Linz oder beispielsweise europaweit verfügbar sein muss.

3.5 Themenbereich Politische Aspekte

<i>Themenbereich:</i>	Politische Aspekte
<i>Kriterium:</i>	Politische Aspekte
<i>Beschreibung:</i>	Einfluss der Politik auf die Entscheidung.
<i>Metrik:</i>	<i>Ausmaß des politischen Interesses:</i> Kein Interesse → Unzureichend Niedrig → Genügend Hoch → Überdurchschnittlich Überdurchschnittlich hoch → Hervorragend
<i>Anmerkung:</i>	Durch den Einfluss von politischen Interessen treten sachliche Aspekte meist in den Hintergrund. Annahme: Je höher das politische Interesse, desto wichtiger wird der Einsatz von OS-Software/Content sein.

<i>Themenbereich:</i>	Politische Aspekte
<i>Kriterium:</i>	Imagegewinn
<i>Beschreibung:</i>	Veränderung des Erscheinungsbildes der OSRGL im Bezug auf die Umwelt. Veränderungspotenzial durch den Einsatz von OS-Software.
<i>Metrik:</i>	<i>Möglicher Einfluss auf das Image:</i> Imageverlust → Unzureichend Neutral → Genügend Leichter Imagegewinn → Überdurchschnittlich Starker Imagegewinn → Hervorragend
<i>Anmerkung:</i>	Hinweise für die Beurteilung des Imagegewinns lassen sich u.U. aus Vergleichen mit anderen OS-Regionen ziehen, aus Diskussion mit Betroffenen, etc.

3.6 Themenbereich Verfügbarkeit

<i>Themenbereich:</i>	Verfügbarkeit
<i>Kriterium:</i>	Verfügbarkeit
<i>Beschreibung:</i>	Beschreibt das Vorhandensein eines Marktes im jeweiligen OS-Bereich.
<i>Metrik:</i>	<i>Ist für das bestehende Problem eine passende OS-Software bzw. ein passender OS-Content vorhanden?</i> JA oder NEIN
<i>Anmerkung:</i>	Indikator dafür ist in erster Linie die Anzahl der OS-Projekte, die es für einen bestimmten OS-Bereich (Domäne) gibt sowie die Lebendigkeit dieser Community.

4 Kriterienkatalog für OS-Software Auswahl

Bei den Kriterien für die Auswahl von OS-Software erfolgt die Gliederung und Darstellung des Kriterienkatalogs anhand der Themenbereiche *Funktionalität, Kosten, Nutzen, Rechtliche Aspekte, Dokumentation, Service, Produktreife, Interne Qualität, Benutzbarkeit* und *Integrierbarkeit*.

Jedes Kriterium wird wie folgt beschrieben:

- Themenbereich: Jedes Kriterium ist einem Themenbereich zugeordnet.
- Kriterium: Bezeichnung des Kriteriums.
- Beschreibung: Beschreibung des Kriteriums.
- Metrik: Wie wird das Kriterium gemessen inkl. Bewertung.
- Anmerkung: Optionale Anmerkungen für eine detailliertere Beschreibung.

Es gibt für die Bewertung der Metriken verschiedene Skalen:

- 2-stufige Bewertungsskala: Bei Bewertungsergebnisse mit genau 2 Möglichkeiten.
- 4-stufige Bewertungsskala: Bei Bewertungsergebnisse mit 4 Abstufungen.

4.1 Themenbereich Funktionalität

<i>Themenbereich:</i>	Funktionalität
<i>Kriterium:</i>	Must Have
<i>Beschreibung:</i>	Anforderungen, die zu 100% erfüllt werden müssen
<i>Metrik:</i>	<i>Ist die Anforderung erfüllt?</i> JA oder NEIN
<i>Anmerkung:</i>	Im allgemeinen (hier vorgestellten) Kriterienkatalog kann auf die Funktionalität nicht genauer eingegangen werden, da diese natürlich abhängig von der jeweiligen Domäne ist. Es empfiehlt sich, für die Beurteilung dieses Kriteriums eine Liste von „Must Have“ Funktionalitäten zu definieren und gegen diese zu prüfen.

<i>Themenbereich:</i>	Funktionalität
<i>Kriterium:</i>	Nice to Have
<i>Beschreibung:</i>	Anforderungen, die gewünscht, jedoch nicht zwingend erforderlich sind.
<i>Metrik:</i>	<i>Erfüllungsgrad der Anforderungen:</i> 0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend
<i>Anmerkung:</i>	Im allgemeinen (hier vorgestellten) Kriterienkatalog kann auf die Funktionalität nicht genauer eingegangen werden, da diese natürlich abhängig von der jeweiligen Domäne ist. Es empfiehlt sich, für die Beurteilung dieses Kriteriums eine Liste von „Nice to Have“ Funktionalitäten zu definieren und gegen diese zu prüfen.

4.2 Themenbereich Kosten

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Migrationskosten
<i>Beschreibung:</i>	Kosten, die bis zur Produktivsetzung anfallen (Investitions-, Projekt-, Schulungs-, Integrationskosten, etc.).
<i>Metrik:</i>	<p><i>Festlegung von Untergrenze (billigstes Angebot) und Obergrenze (teuerstes Angebot).</i></p> <p>Prozent-Wert = $(\text{aktuelles Angebot} - \text{Untergrenze}) / (\text{Obergrenze} - \text{Untergrenze}) * 100$</p> <p>0-25% → Hervorragend 26-50% → Überdurchschnittlich 51-75% → Genügend ab 76% → Unzureichend</p>

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Betriebskosten
<i>Beschreibung:</i>	Kosten, die während des Betriebs anfallen.
<i>Metrik:</i>	<p><i>Festlegung von Untergrenze (billigstes Angebot) und Obergrenze (teuerstes Angebot).</i></p> <p>Prozent-Wert = $(\text{aktuelles Angebot} - \text{Untergrenze}) / (\text{Obergrenze} - \text{Untergrenze}) * 100$</p> <p>0-25% → Hervorragend 26-50% → Überdurchschnittlich 51-75% → Genügend ab 76% → Unzureichend</p>

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Wartungskosten
<i>Beschreibung:</i>	Kosten, die für Fehlerbehebungen, Aktualisierungen und Upgrades anfallen.
<i>Metrik:</i>	<p><i>Festlegung von Untergrenze (billigstes Angebot) und Obergrenze (teuerstes Angebot).</i></p> <p>Prozent-Wert = $(\text{aktuelles Angebot} - \text{Untergrenze}) / (\text{Obergrenze} - \text{Untergrenze}) * 100$</p> <p>0-25% → Hervorragend 26-50% → Überdurchschnittlich 51-75% → Genügend ab 76% → Unzureichend</p>
<i>Anmerkung:</i>	Der Betrachtungszeitraum muss individuell festgelegt werden.

<i>Themenbereich:</i>	Kosten
<i>Kriterium:</i>	Know-how Aufbau
<i>Beschreibung:</i>	Kosten, die beim Erwerb von Know-how über das einzuführende System anfallen.
<i>Metrik:</i>	<p><i>Festlegung von Untergrenze (billigstes Angebot) und Obergrenze (teuerstes Angebot).</i></p> <p>Prozent-Wert = (aktuelles Angebot – Untergrenze) / (Obergrenze – Untergrenze) * 100</p> <p>0-25% → Hervorragend 26-50% → Überdurchschnittlich 51-75% → Genügend ab 76% → Unzureichend</p>
<i>Anmerkung:</i>	Der Betrachtungszeitraum muss individuell festgelegt werden.

4.3 Themenbereich Nutzen

<i>Themenbereich:</i>	Nutzen
<i>Kriterium:</i>	Imagegewinn
<i>Beschreibung:</i>	Veränderung des Erscheinungsbildes des Unternehmens in Bezug auf die Umwelt. Veränderungspotenzial durch den Einsatz von OS-Software.
<i>Metrik:</i>	<p><i>Mögliche Veränderung des Erscheinungsbildes:</i></p> <p>Imageverlust → Unzureichend Neutral → Genügend Leichter Imagegewinn → Überdurchschnittlich Starker Imagegewinn → Hervorragend</p>

<i>Themenbereich:</i>	Nutzen
<i>Kriterium:</i>	Community
<i>Beschreibung:</i>	Eine Community ist eine Gemeinschaft von Personen, die freiwillig, unentgeltlich Beiträge für die Nutzung von Open Source liefert. In diesem Kriterium ist das Aussehen der Community zu jener Software und Größe, Alter und fachliche Kompetenz der Community in Bezug zur Software relevant.
<i>Metrik:</i>	<p><i>Wie lange existiert die Community?</i></p> <p>Bis 6 Monate → Unzureichend > 6 Monate → Genügend >1 Jahr → Überdurchschnittlich >3 Jahre → Hervorragend</p> <p><i>Größe der Community (Kann durch die Anzahl der Beiträge, Anzahl der aktiven Entwickler etc. gemessen werden):</i></p> <p>Zu klein → Unzureichend Ausreichend → Genügend Groß → Überdurchschnittlich Sehr groß → Hervorragend</p> <p><i>Fachliche Kompetenz der Community (Kann durch Qualität der Beiträge in Foren oder dergleichen gemessen werden):</i></p> <p>Unzureichend Genügend Überdurchschnittlich Hervorragend</p>

<i>Themenbereich:</i>	Nutzen
<i>Kriterium:</i>	Know-how Gewinn
<i>Beschreibung:</i>	Prognostizierte Steigerung des Wettbewerbsvorteils, Wissenszuwachs und des strategischen Unternehmenswertes.
<i>Metrik:</i>	<p><i>Gibt es eine Steigerung des Wettbewerbsvorteils, Wissenszuwachs und des strategischen Unternehmenswertes?</i></p> <p>Hervorragend Überdurchschnittlich Genügend Unzureichend</p>

<i>Themenbereich:</i>	Nutzen
<i>Kriterium:</i>	Produktivitätssteigerung
<i>Beschreibung:</i>	Dieses Kriterium vergleicht die Durchlaufzeit der Geschäftsprozesse vor Einsatz der OS-Software mit der prognostizierten Durchlaufzeit unter Einsatz der OSS.
<i>Metrik:</i>	<p><i>Verhältnis neue Durchlaufzeit zu alte Durchlaufzeit:</i></p> <p>Prozent-Wert = neue Durchlaufzeit / alte Durchlaufzeit * 100</p> <p>< 50% → Hervorragend 50-75% → Überdurchschnittlich 76-99% → Genügend Ab 100% → Unzureichend</p>

<i>Themenbereich:</i>	Nutzen
<i>Kriterium:</i>	Kostensenkung
<i>Beschreibung:</i>	Relation der Betriebs- und Wartungskosten von altem (derzeitigen) System und neuem System (mit neuer OS-Software).
<i>Metrik:</i>	<p><i>Kostensenkungspotential der Betriebs- und Wartungskosten:</i></p> <p>Prozent-Wert = neue Betriebs- und Wartungskosten / derzeitige Betriebs- und Wartungskosten * 100</p> <p><= 70% → Hervorragend >70% → Überdurchschnittlich >85% → Genügend >95% → Unzureichend</p>

4.4 Themenbereich Rechtliche Aspekte

<i>Themenbereich:</i>	Rechtliche Aspekte
<i>Kriterium:</i>	Lizenz
<i>Beschreibung:</i>	Rechtliche/vertragliche Bedingungen für die Verwendung. Die Bewertung kann individuell und den jeweiligen Anforderungen entsprechend durchgeführt werden. Deswegen kann keine generelle Metrik festgelegt werden.
<i>Metrik:</i>	<p><i>Lizenzmodell:</i></p> <p>Nicht passend → Unzureichend Ausreichend → Genügend Akzeptabel → Überdurchschnittlich Optimal → Hervorragend</p>

<i>Themenbereich:</i>	Rechtliche Aspekte
<i>Kriterium:</i>	Copyright
<i>Beschreibung:</i>	Abhängig vom Lizenzmodell muss darauf geachtet werden, dass in der wirtschaftlichen Weiterverwendung der Software die Copyright-Bedingungen des Herstellers berücksichtigt werden.
<i>Metrik:</i>	<p><i>Copyright Aspekte:</i></p> <p>Nicht passend → Unzureichend Ausreichend → Genügend Akzeptabel → Überdurchschnittlich Optimal → Hervorragend</p>
<i>Anmerkung:</i>	Die Bewertung kann individuell und den jeweiligen Anforderungen entsprechend durchgeführt werden. Deswegen kann keine generelle Metrik festgelegt werden.

<i>Themenbereich:</i>	Rechtliche Aspekte
<i>Kriterium:</i>	Copyleft
<i>Beschreibung:</i>	Copyleft ist das Gegenstück zum Copyright. Copyleft überlässt den Anwender das Recht zu Kopieren.
<i>Metrik:</i>	<p><i>Copyleft Aspekte:</i></p> <p>Nicht passend → Unzureichend Ausreichend → Genügend Akzeptabel → Überdurchschnittlich Optimal → Hervorragend</p>
<i>Anmerkung:</i>	Die Bewertung kann wiederum individuell und je nach Anforderungen durchgeführt werden. Deshalb kann auch hier keine generelle Metrik festgelegt werden.

4.5 Themenbereich Dokumentation

<i>Themenbereich:</i>	Dokumentation
<i>Kriterium:</i>	Verfügbarkeit
<i>Beschreibung:</i>	Hier wird geprüft ob die verschiedenen Arten von Dokumentationen (Benutzer-, Wartungs-, Betriebsdokumentation, etc.) verfügbar sind.
<i>Metrik:</i>	<p><i>Verfügbarkeit:</i></p> <p>Vollständig vorhanden → Hervorragend Großteils vorhanden → Überdurchschnittlich Wesentlich vorhanden → Genügend Kaum vorhanden → Unzureichend</p>
<i>Anmerkung:</i>	Die Beurteilung wird hier sehr stark von individuellen Dokumentationsanforderungen abhängen. Diese sind im Zuge der Untersuchung noch zu verfeinern.

<i>Themenbereich:</i>	Dokumentation
<i>Kriterium:</i>	Vollständigkeit
<i>Beschreibung:</i>	Darunter versteht man den Grad der Abdeckung der Themenbereiche der beschriebenen Software.
<i>Metrik:</i>	<p><i>Wie lückenlos ist die Dokumentation geschrieben?</i></p> <p>Keine Dokumentation vorhanden → Unzureichend Wesentliche Teile fehlen → Genügend Wesentliche Teile vorhanden → Überdurchschnittlich Alle Themenbereiche abgedeckt → Hervorragend</p> <p><i>Wenn Kritiken verfügbar sind: Ergebnis der Kritik:</i></p> <p>Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Bei den Kritiken muss berücksichtigt werden, dass eine optimale Aussagekraft erst bei einer repräsentativen Anzahl von Kunden gewährleistet ist.

<i>Themenbereich:</i>	Dokumentation
<i>Kriterium:</i>	Aktualität
<i>Beschreibung:</i>	Eine Dokumentation kann optimal verwendet werden, wenn eine Anpassung an den aktuellen Softwarestand erfolgt.
<i>Metrik:</i>	<p><i>Regelmäßigkeit:</i></p> <p>Prozent-Wert = Anzahl der Dokumentationsversionen / Anzahl der Software Releases * 100</p> <p><=50% → Unzureichend >50% → Genügend >70% → Überdurchschnittlich >90% → Hervorragend</p> <p><i>Aktualität (Wie lange dauert die Ausfolgung der aktualisierten Dokumentation nach einem neuen Software Release?):</i></p> <p>>12 Monate → Unzureichend >6 Monate → Genügend >1 Monat → Überdurchschnittlich <= 1 Monat → Hervorragend</p>

<i>Themenbereich:</i>	Dokumentation
<i>Kriterium:</i>	Inhaltliche Qualität
<i>Beschreibung:</i>	Beim geschriebenen Wort (z.B. Dokumentation) ist es sehr wichtig, die notwendigen Informationen in einer geeigneten Weise darzustellen. Es muss dem Leser so leicht möglich gemacht werden, die gesuchten Informationen schnell zu erfassen.
<i>Metrik:</i>	<p><i>Strukturierter Aufbau:</i></p> <p>Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p> <p><i>Verständlichkeit:</i></p> <p>Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>

4.6 Themenbereich Service

<i>Themenbereich:</i>	Service
<i>Kriterium:</i>	Support
<i>Beschreibung:</i>	Support ist die problembezogene Beratungstätigkeit in Form von beispielsweise einem Help-Desk-Systems oder einem Callcenter.
<i>Metrik:</i>	<p><i>Gibt es einen Support?</i> JA oder NEIN</p> <p><i>Qualität des Supports:</i> Unzureichend Genügend Überdurchschnittlich Hervorragend</p> <p><i>Wie rasch ist der Support verfügbar?</i> >3 Tage → Unzureichend 12h – 3 Tage → Genügend 2h – 12h → Überdurchschnittlich <2h → Hervorragend</p> <p><i>Ergebnis der Kritik:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Falls es keinen Support gibt, wird dieses Kriterium automatisch mit 0 bewertet. Falls es einen Support gibt, wird die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen drei Metriken bestimmt. Sollte es keine Kritiken geben, erfolgt die Bewertung dieses Kriteriums durch das arithmetische Mittel der ersten beiden Metriken.

<i>Themenbereich:</i>	Service
<i>Kriterium:</i>	Service Level Agreements
<i>Beschreibung:</i>	Service Level Agreements sind ein Vertrag zwischen einem Auftraggeber und einem Dienstleister. Dem Auftraggeber wird daher eine Leistung, in einer bestimmten Zeit und einem bestimmten Umfang vertraglich zugeteilt.
<i>Metrik:</i>	<p><i>Gibt es Service Level Agreements (SLAs) für diese Software?</i> JA oder NEIN</p> <p><i>Umfang der SLAs:</i> Unzureichend Genügend Überdurchschnittlich Hervorragend</p> <p><i>Qualität der SLAs:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p> <p><i>Reaktionszeit (Wie rasch wird die Leistung erbracht?):</i> >3 Tage → Unzureichend >1 Tag – 3 Tage → Genügend 6h - 1 Tag → Überdurchschnittlich <6h → Hervorragend</p> <p><i>Ergebnis der Kritik:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Falls es keine SLAs für die Software gibt, wird das Kriterium automatisch mit 0 bewertet. Falls es SLAs gibt, wird die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen vier Metriken bestimmt. Sollte es keine Kritiken geben, erfolgt die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen drei Metriken.

<i>Themenbereich:</i>	Service
<i>Kriterium:</i>	Training
<i>Beschreibung:</i>	Training bezeichnet das Anlernen von gewissen Tätigkeiten beziehungsweise Prozessen. Training kann sowohl auf körperlicher als auch auf emotionaler oder kognitiver Ebene stattfinden.
<i>Metrik:</i>	<p><i>Gibt es ein Training bzw. eine Schulung für die Software?</i> JA oder NEIN</p> <p><i>Qualität des Trainings:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p> <p><i>Umfang: Ist das Training im gewünschten Umfang? Deckt es alle Bereiche ab? Ist es zu detailliert/umfangreich?</i> Wesentliche Teile nicht abgedeckt bzw. viel zu detailliert → Unzureichend nicht ganz alle gewünschten Bereiche abgedeckt / zu detailliert → Genügend etwas zu detailliert → Überdurchschnittlich Optimal → Hervorragend</p> <p><i>Ergebnis der Kritik:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Falls es kein Training bzw. keine Schulung für die Software gibt, wird das Kriterium automatisch mit 0 bewertet. Falls es ein Training bzw. eine Schulung gibt, wird die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen drei Metriken bestimmt. Sollte es keine Kritiken geben, erfolgt die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen zwei Metriken.

<i>Themenbereich:</i>	Service
<i>Kriterium:</i>	Consulting
<i>Beschreibung:</i>	Unter Consulting ist die außerbetriebliche Beratungstätigkeit gemeint. Diese Beratungstätigkeit umfasst Aufgaben, welche im Haus des Auftraggebers nicht verfügbar sind.
<i>Metrik:</i>	<p><i>Ist ein Consulting verfügbar?</i> JA oder NEIN</p> <p><i>Qualität des Consultings:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p> <p><i>Ergebnis der Kritik:</i> Schlecht → Unzureichend In Ordnung → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Falls es kein Consulting gibt, wird das Kriterium automatisch mit 0 bewertet. Falls es ein Consulting gibt, wird die Bewertung dieses Kriteriums durch das arithmetische Mittel der anderen zwei Metriken bestimmt. Sollte es keine Kritiken geben, erfolgt die Bewertung dieses Kriteriums durch die Metrik „Qualität des Consulting“.

4.7 Themenbereich Produktreife

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Alter
<i>Beschreibung:</i>	Durch das Alter einer Software kann man einen Rückschluss auf den Reifegrade der Software ziehen. Erfahrungsgemäß beinhalten neu auf den Markt gekommene Softwareprodukte sehr viele Fehler.
<i>Metrik:</i>	<p><i>Anzahl der Jahre:</i> ≤ 1 → Unzureichend >1 → Genügend >2 → Überdurchschnittlich >3 → Hervorragend</p>

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Stabilität
<i>Beschreibung:</i>	Stabilität steht für die Beständigkeit eines Systems.
<i>Metrik:</i>	<p><i>Verfügbarkeit (An wie vielen Tagen im Jahr konnte prozentuell ein reibungsloser Produktivbetrieb gewährleistet werden?):</i> ≤96% → Unzureichend >96% → Genügend >98,5% → Überdurchschnittlich >99,5% → Hervorragend</p>
<i>Anmerkung:</i>	Gemessen wird dies durch die Jahresverfügbarkeit, sofern es mindestens einen vergleichbaren Referenzkunden gibt. Ist dies nicht der Fall, kann das Kriterium nicht evaluiert werden.

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Aktivität
<i>Beschreibung:</i>	Stillstand ist Rückschritt. Mit diesem Kriterium wird hinterfragt, in wie fern sich das Produkt in einem aktiven Entwicklungsprozess befindet.
<i>Metrik:</i>	<p><i>Dauer der Fehlerbehebung (Wie viel Zeit wird benötigt um die Software wieder funktionsfähig zu machen?):</i></p> <p>>=3 Tage → Unzureichend <3 Tage → Genügend <24 h → Überdurchschnittlich < 5 h → Hervorragend</p> <p><i>Aktivität (Wird die Software noch weiter entwickelt? Gibt es regelmäßige Updates oder ist man schon zum Stillstand gekommen?):</i></p> <p>Unzureichend Genügend Überdurchschnittlich Hervorragend</p>

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Referenzen
<i>Beschreibung:</i>	Damit sind dritte Firmen gemeint, die das Produkt in einer vergleichbaren Installation einsetzen.
<i>Metrik:</i>	<p><i>Anzahl der Referenzen:</i></p> <p>< 3 → Unzureichend <5 → Genügend <7 → Überdurchschnittlich >= 7 → Hervorragend</p>

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Popularität
<i>Beschreibung:</i>	Beim Kriterium Popularität wird die Produktreife mit dem Grad der breiten Verwendung beschrieben.
<i>Metrik:</i>	<p><i>Anzahl der Kunden:</i></p> <p><= 5 → Unzureichend >5 → Genügend >15 → Überdurchschnittlich >30 → Hervorragend</p> <p><i>Bewertung durch Kunden:</i></p> <p>Schlecht → Unzureichend Mittelmäßig → Genügend Gut → Überdurchschnittlich Sehr gut → Hervorragend</p>
<i>Anmerkung:</i>	Bei der Bewertung von Kunden muss berücksichtigt werden, dass eine optimale Aussagekraft erst bei einer repräsentativen Anzahl von Kunden gewährleistet ist.

<i>Themenbereich:</i>	Produktreife
<i>Kriterium:</i>	Entwickler
<i>Beschreibung:</i>	Die Produktreife wird durch die Qualität der Entwicklungsarbeit beschrieben.
<i>Metrik:</i>	<p><i>Anzahl der Entwickler (Wie viele Community-Entwickler arbeiten an der Software?):</i></p> <p><= 1 → Unzureichend <5 → Genügend <15 → Überdurchschnittlich >=15 → Hervorragend</p>
<i>Anmerkung:</i>	Je weniger Entwickler beteiligt sind, desto höher ist die Abhängigkeit von einzelnen Personen.

4.8 Themenbereich Interne Qualität

<i>Themenbereich:</i>	Interne Qualität
<i>Kriterium:</i>	Quellcode
<i>Beschreibung:</i>	Einhaltung von Standards im Sinne von Programmierrichtlinien.
<i>Metrik:</i>	<p><i>Ergebnisse Codeanalyse:</i></p> <p>0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend</p>
<i>Anmerkung:</i>	Die Beurteilung der internen Qualität auf Basis einer Codeanalyse erfordert Experten Know-how. Es sollten die Ergebnisse einer Code-Analyse mit den Ergebnissen ähnlicher Software verglichen werden bzw. sollte ein Augenmerk darauf gelegt werden, inwieweit schwerwiegende Fehler bei der Code-Analyse zu Tage treten.

<i>Themenbereich:</i>	Interne Qualität
<i>Kriterium:</i>	Sicherheit
<i>Beschreibung:</i>	Liste der bekannten Sicherheitslücken sowie deren Behebungsstatus.
<i>Metrik:</i>	<p><i>Prozentsatz der behobenen Sicherheitslücken:</i></p> <p>Prozentwert = Anzahl behobener Sicherheitslücken / Anzahl bekannter Sicherheitslücken * 100</p> <p>0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend</p> <p><i>Anzahl der bekannten Sicherheitslücken:</i></p> <p>> 15 → Unzureichend > 10 → Genügend > 5 → Überdurchschnittlich <=5 → Hervorragend</p>

<i>Themenbereich:</i>	Interne Qualität
<i>Kriterium:</i>	Architektur
<i>Beschreibung:</i>	<p>Der Begriff Architektur beschreibt im Allgemeinen das Zusammenspiel der einzelnen Komponenten. Jedoch wird hier auch der „Bauplan“ der Software betrachtet.</p> <p>In diesem Kriterium werden zum Beispiel der Modularisierungsgrad, die Offenheit, die Verwendung von Standard-APIs und die Skalierbarkeit betrachtet.</p> <p>Aufgrund der komplexen Einflussfaktoren einer Architektur kann keine generelle Metrik festgelegt werden.</p>
<i>Metrik:</i>	<p><i>Architekturmodell:</i></p> <p>Nicht passend → Unzureichend Ausreichend → Genügend Akzeptabel → Überdurchschnittlich Optimal → Hervorragend</p>
<i>Anmerkung:</i>	<p>Die Beurteilung der Architektur erfordert Experten Know-how. Im Sinne einer Standardisierung können Architekturprüfwerkzeuge (structure 101, Lattix LDM, JDepend) oder ähnliches eingesetzt werden, um zu einer objektiveren Beurteilung von Architektur- und Designfragen zu kommen.</p>

<i>Themenbereich:</i>	Interne Qualität
<i>Kriterium:</i>	Qualitätsmanagement
<i>Beschreibung:</i>	Unter QM werden all jene grundsätzlichen Maßnahmen verstanden, welche der Verbesserung von Software und deren Entwicklungsprozess dienen.
<i>Metrik:</i>	<p><i>Prozessqualität:</i> Unzureichend Genügend Überdurchschnittlich Hervorragend</p> <p><i>Produktqualität:</i> Prozentwert = Anzahl behobener Bugs / Anzahl bekannter Bugs * 100 0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend</p> <p><i>Einhaltung von Standards:</i> Prozentwert = Anzahl eingehaltener Standards / Anzahl notwendiger bzw. zutreffenden Standards * 100 0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend</p> <p><i>Anzahl Reviews/Zertifikate:</i> <5 → Unzureichend 5-7 → Genügend 8-10 → Überdurchschnittlich ab 11 → Hervorragend</p>
<i>Anmerkung:</i>	<p>Prozessqualität prüft ob etablierte Modelle der Softwareentwicklung verwendet werden oder nicht.</p> <p>Die Produktqualität betrachtet die Liste der bekannten Programmfehler (Bugs) sowie deren Behebungsstatus.</p> <p>Bei der Einhaltung von Standards wird die Liste der notwendigen bzw. zutreffenden Standards in Relation zu den tatsächlich eingehaltenen Standards gesetzt.</p>

4.9 Themenbereich Benutzbarkeit

<i>Themenbereich:</i>	Benutzbarkeit
<i>Kriterium:</i>	Bedienbarkeit
<i>Beschreibung:</i>	Jene Eigenschaft einer Software, welche den Anwender bei der direkten Aufgabenerfüllung effektiv und effizient unterstützt.
<i>Metrik:</i>	<p><i>Wie einfach ist die Software zu bedienen (Schwierigkeitsgrad)?</i> Schwer → Unzureichend Mittel → Genügend Leicht → Überdurchschnittlich Intuitiv → Hervorragend</p>

<i>Themenbereich:</i>	Benutzbarkeit
<i>Kriterium:</i>	Erlernbarkeit
<i>Beschreibung:</i>	Jene Eigenschaft einer Software, welche es einem durchschnittlichen Benutzer ermöglicht ohne großen Lernaufwand seine Aufgaben um zu setzen.
<i>Metrik:</i>	<p><i>Wie einfach erlernbar ist die Benutzung der Software?</i></p> <p>>3 Schulungstage → Unzureichend <= 3 Schulungstage Genügend Selbststudium mit Doku → Überdurchschnittlich Intuitiv → Hervorragend</p>

<i>Themenbereich:</i>	Benutzbarkeit
<i>Kriterium:</i>	Barrierefreiheit
<i>Beschreibung:</i>	Nutzbarkeit für Personen mit unterschiedlichen Fähigkeiten und Bedürfnissen, z.B. sehbehinderte Menschen.
<i>Metrik:</i>	<p><i>Wie viele Kriterien des WCAG2.0 Standards für Barrierefreiheit sind erfüllt?</i></p> <p><50% → Unzureichend <70% → Genügend <85% → Überdurchschnittlich >=85% → Hervorragend</p>

<i>Themenbereich:</i>	Benutzbarkeit
<i>Kriterium:</i>	Performance
<i>Beschreibung:</i>	Darunter ist das Antwortzeitverhalten eines Systems zu verstehen. Dieses ist abhängig von der Anwendung und dem Funktionsumfang.
<i>Metrik:</i>	<p><i>Wie gut ist die Performance?</i></p> <p>Unzureichend Genügend Überdurchschnittlich Hervorragend</p>

4.10 Themenbereich Integrierbarkeit

<i>Themenbereich:</i>	Integrierbarkeit
<i>Kriterium:</i>	Integrierbarkeit
<i>Beschreibung:</i>	Darunter wird der Grad, in welchem sich die Software in die bestehende Infrastruktur des Einsatzgebietes integrieren lässt, verstanden.
<i>Metrik:</i>	<p><i>In welchem Ausmaß lässt sich die Software aufgrund der standardmäßig gebotenen Schnittstellen in die Infrastruktur integrieren?</i></p> <p>Prozent-Wert= Anzahl der standardmäßig unterstützten Schnittstellen / Anzahl der notwendigen Schnittstellen * 100</p> <p>Erfüllungsgrad:</p> <ul style="list-style-type: none"> 0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend <p><i>In welchem Ausmaß lässt sich die Software aufgrund der standardmäßig unterstützten Formate in die Infrastruktur integrieren?</i></p> <p>Prozent-Wert= Anzahl der standardmäßig unterstützten Formate / Anzahl der notwendigen Formate * 100</p> <p>Erfüllungsgrad:</p> <ul style="list-style-type: none"> 0-25% → Unzureichend 26-50% → Genügend 51-75% → Überdurchschnittlich ab 76% → Hervorragend

5 Entscheidungsprozesse

Bei der Durchführung von OS-Evaluierung ist darauf zu achten, dass bei optimiertem Aufwand ein qualitativ hochwertiges Ergebnis erzielt wird. Bestmögliche Zeitersparnis kann man durch die Früherkennung jener Produkte erzielen, die den Anforderungen offensichtlich nicht entsprechen.

Damit die rasche Ausfilterung von ungeeigneten Produkten möglich ist, müssen die Evaluierungskriterien untergliedert werden. Es muss darauf geachtet werden, dass zuerst die Kriterien geprüft werden, die erstens vollständig erfüllt werden müssen und zweitens deren Prüfung ohne viel Aufwand erfolgen kann. Erst in späteren Evaluierungsschritten werden Kriterien geprüft, deren Erfüllungsgrad interpretiert und in Relation zu anderen Kriterien gesetzt werden muss. Da in dieser Evaluierungsphase der Aufwand der Prüfung ansteigt, sollen nur mehr Produkte geprüft werden, die bereits in der engeren Wahl sind.

Wie schon in der Einleitung erwähnt, ist es definitiv nicht Ziel dieser Arbeit, eine vollständige Evaluierungsmethode für die Ermittlung des Potenzials bzw. für die Bewertung von Alternativen vorzuschlagen – dies hängt sehr stark von der Kultur in der Organisation bzw. von der Kultur, wie Entscheidungen getroffen werden, ab. Als Rahmen schlagen wir dennoch ein Entscheidungsmodell vor (siehe Kapitel 5.2), das auf dem allgemeinen Stage-Gate Modell von Cooper [Cooper 2001] beruht (siehe Kapitel 5.1).

5.1 Stage-Gate Modell nach Cooper

Die Stage-Gate Methode von Cooper [Cooper 2001] repräsentiert einen Ansatz, um den Produktentwicklungsprozess wirkungsvoller zu gestalten. Von der Idee bis zur Einführung eines Produktes ist eine Anzahl von definierten Schritten oder Stadien zu durchlaufen (siehe Abbildung 9).

Stage-Gate® Product Innovation Process



Abbildung 9: Allgemeiner Stage-Gate Prozess für Produktinnovationen

Entscheidend ist, dass der Zugang zu jedem Stadium durch ein Gate (Tor) „abgesichert“ ist. Diese Gates sind normalerweise Treffen, die den Entwicklungsprozess kontrollieren und dienen als

- Qualitätskontrolle
- Go/Kill Kontrollpunkte
- Wegweiser für Aktionsplan für die folgende Phase.

Der wesentliche Vorteil dieses Ansatzes ist, dass komplexe Entscheidungsprozesse auf eine Anzahl von kleineren Schritten heruntergebrochen wird und beschleunigt (in unserem Fall) die Auswahlentscheidung, da „schlechte“ Alternativen früher ausgesondert werden.

5.2 Entscheidungsmodell für Potentialanalyse und Softwareauswahl

Für die Durchführung der OS-Evaluierung wurde ein Stufenmodell für die Untergliederung der Evaluierungskriterien erstellt. Dieses Modell besteht aus 3 Stufen (Gate 0 - 2):

- Bei der Stufe "Gate 0" wird geprüft, ob für die gegebenen Anforderungen ein OS-Einsatzgebiet vorhanden ist bzw. ein dementsprechendes Potential erkennbar ist.
- Kann grundsätzlich ein Einsatzgebiet festgestellt werden, wird in der Stufe "Gate 1" begonnen, einzelne OS-Produkte zu prüfen. Bei dieser Stufe werden k.o.-Kriterien geprüft. Diese Kriterien müssen vollständig erfüllt werden und somit können Produkte, die eines der für dieses Gate definierten Kriterien nicht erfüllt, rasch ausgefiltert werden.
- Bei der letzten Stufe "Gate 2" werden die restlichen Evaluierungskriterien geprüft. Dabei werden die einzelnen Themenbereiche priorisiert und für jeden Themenbereich ein Erfüllungsgrad festgelegt; d.h. alle zugeteilten Evaluierungskriterien müssen in Summe diesen Erfüllungsgrad erreichen.

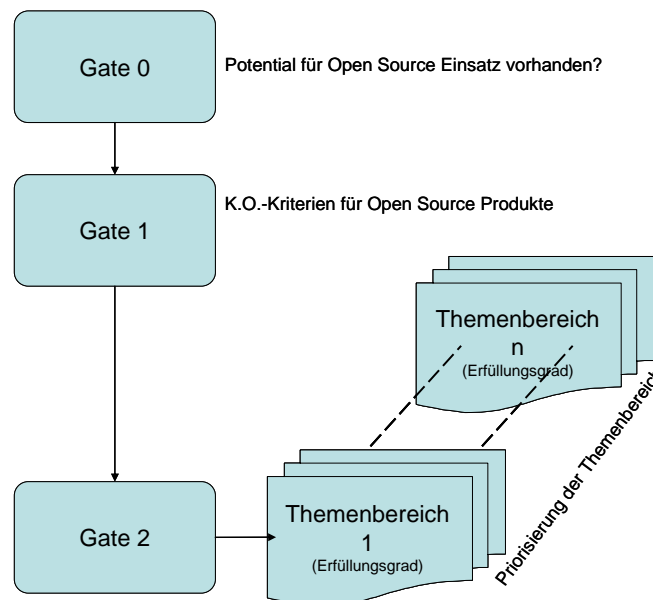


Abbildung 10: Stufenverwaltungsmodell

5.2.1 Durchführung der Potentialanalyse (Gate 0)

Im ersten Schritt wird im Gate 0 evaluiert, ob Open Source eingesetzt werden kann oder nicht. Dies erfolgt mit Hilfe der in Kapitel 3 (Kriterienkatalog für Potenzialanalyse) definierten Kriterien.

- Bevor die einzelnen Kriterien evaluiert werden können, muss ein Erfüllungsgrad für Kriterien mit 4-stufiger Bewertungsskala definiert werden. Bei den Kriterien mit 2-stufiger Bewertungsskala wird davon ausgegangen, dass diese Kriterien erfüllt werden müssen.
- Danach können die einzelnen Kriterien bewertet werden. Jedes Kriterium wird anhand seiner Metrik(en) bewertet. Hat ein Kriterium mehrere Metriken wird das arithmetische Mittel der Werte als Bewertung des Kriteriums herangezogen.
- Zuletzt wird die Bewertung der Metrik mit dem Erfüllungsgrad verglichen. Wird ein Erfüllungsgrad eines Kriteriums nicht erreicht, so wird die Evaluierung der OS-Software

bzw. des OS-Contents abgebrochen. Andernfalls wird mit der Bewertung des nächsten Kriteriums fortgefahren. Erfüllen alle Kriterien ihre Anforderungen wird mit Gate 1 fortgesetzt.

5.2.2 Vorbereitungsarbeiten für die Bewertung in den Gates 1 und 2

Falls die Evaluierung im Gate 0 ergeben hat, dass Open Source verwendet werden könnte (d.h., ausreichend Potential vorhanden ist), erfolgt in den Gates 1 und 2 die Bewertung der einzelnen Alternativen. Für diese Bewertung sind Vorbereitungsarbeiten erforderlich:

- *Zuordnung von Themenbereichen zu Gates:* Um eine Software anhand dieses Kriterienkatalogs zu evaluieren, müssen die Themenbereiche (siehe Kapitel 4) den einzelnen Gates zugeordnet werden. Dabei werden Themenbereiche, die aus Sicht der Entscheider k.o.-Kriterien enthalten, also all jene die zwingend erfüllt sein müssen um eine Software einzusetzen, Gate 1 zugeordnet, alle anderen Gate 2. Alternativ dazu besteht die Möglichkeit, die Zuordnung auf die einzelnen Gates in feinerer Granularität vorzunehmen, d.h. nicht auf der Ebene von Themenbereichen, sondern auf der Ebene von Kriterien. Dabei ist jedoch sicherzustellen, dass die Zuordnung zu den jeweiligen Gates exklusiv erfolgt; es sollte nicht vorkommen, dass ein und das selbe Kriterium parallel beiden Gates zugeordnet wird.
- *Priorisierung der Themenbereiche für Gate 2:* Weiters werden alle Themenbereiche von Gate 2 priorisiert. Welche Skala dabei verwendet wird, ist unerheblich; es geht im Wesentlichen darum, eine Reihenfolge für die Evaluierung der einzelnen Themenbereiche festzulegen.
- *Festlegen der Erfüllungsgrade:* Für alle Kriterien muss ein Erfüllungsgrad festgelegt werden. Der Erfüllungsgrad ist jener Wert den die aufsummierte Bewertung der einzelnen Kriterien der Gruppe erreichen muss, um mit der Evaluierung der Software, also mit der Bewertung der übrigen Themenbereiche, fortzufahren. Für Themenbereiche, die dem Gate 1 zugeordnet sind, sollten hier tendentiell hohe Erfüllungsgrade festgelegt werden, um gegebenenfalls früh die Anzahl der zu vergleichenden Alternativen reduzieren zu können.
- *Festlegen der Wertesysteme:* Im Anschluss an die Definition der Erfüllungsgrade müssen die Wertesysteme festgelegt werden. Diese werden benötigt um den Kriterien Prioritäten zuzuweisen. Die Wertesysteme können frei gewählt werden, jedoch muss, bei der Evaluierung mehrerer Softwarealternativen, das Wertesystem eines Kriteriums bei allen Alternativen gleich sein. Es besteht natürlich auch die Möglichkeit, die Wertesysteme für alle Kriterien eines Themenbereiches bzw. für alle Themenbereiche gleich festzulegen; im letzteren Fall gehen dadurch alle Kriterien mit dem gleichen Gewicht in die Beurteilung ein. Zur Priorisierung stehen drei verschiedene Wertesysteme zur Verfügung. Wertesystem 1 kann für wichtige, Wertesystem 2 für normale und Wertesystem 3 kann für unwichtige Kriterien verwendet werden.

Die folgende Tabelle listet die drei Wertesysteme auf:

	Hervorragend	Überdurchschnittlich	Genügend	Unzureichend
WS 1	10	6	4	0
WS 2	6	4	2	0
WS 3	3	2	1	0

Tabelle 1: Verfügbare Wertesysteme

5.2.3 Bewertung der Kriterien von Gate 1

Als erstes werden alle Kriterien von Gate 1 überprüft. Dabei werden die einzelnen Kriterien Schritt für Schritt bewertet. Ist ein Kriterium erfüllt, wird mit dem nächsten fortgefahren, ist eines nicht erfüllt (d.h., der spezifizierte Erfüllungsgrad wird nicht erreicht), wird die Bewertung der jeweiligen Software abgebrochen.

5.2.4 Bewertung der Kriterien von Gate 2

In diesem Schritt werden die einzelnen Kriterien der priorisierten Themenbereiche der Reihe nach evaluiert. Dabei wird mit dem wichtigsten Themenbereich begonnen, die am wenigsten wichtigen Themenbereiche kommen zuletzt an die Reihe. Innerhalb eines Themenbereiches werden die einzelnen Kriterien anhand ihrer Metriken bewertet. Hat ein Kriterium mehrere Metriken wird wiederum das arithmetische Mittel der Werte als Bewertung des Kriteriums herangezogen.

Wurden alle Kriterien bewertet, werden die einzelnen Bewertungen eines Themenbereichs kumuliert und der daraus resultierende Wert mit den erforderlichen Erfüllungsgrad verglichen. Dafür müssen die Erfüllungsgrade der Kriterien kumuliert werden. Wird dieser nicht erreicht, wird mit der Evaluierung der Software aufgehört. Beim Übertreffen des Wertes wird mit dem nächsten Themenbereich fortgefahren.

5.2.5 Interpretation der Ergebnisse

Zuletzt werden, falls der Erfüllungsgrad in allen Themenbereichen von Gate 2 überschritten wurde, die einzelnen Bewertungen der Themenbereiche noch aufsummiert um eine Gesamtbeurteilung der Software zu erhalten. Diese wird daraufhin mit jenen alternativer Softwareprodukte verglichen. Es sollte eine Entscheidung für jene Software mit dem höchsten Wert getroffen werden.

Literatur

- [Basili 1988] Basili, V.R. und Rombach, H.D., "The TAME Project: Towards Improvement-Oriented Software Environments", IEEE Transactions on Software Engineering, 1988, Maryland
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel1%2F32%2F300%2F00006156.pdf%3Farnumber%3D6156&authDecision=-203> (Abgerufen am 31.5.2009)
- [Bianco et al. 2008] Bianco, V.d., Lavaza, L., Morasca, S., Taibi, D., "The observed characteristics and relevant factors used for assessing the trustworthiness of OSS products and artefacts", Version 1.0, 2008, Qualipso, University of Insubria; <http://qualipso.org/node/128>
- [Cooper 2001] Cooper R.G.: *Winning at New Products: Accelerating the Process from Idea to Launch*, Perseus Books, 3. Auflage, 2001
- [Fraunhofer 2009] "Qualipso – Innovation und Wachstum stärken durch den Einsatz von Open Source Software",
http://www.fokus.fraunhofer.de/de/elan/projekte/international/laufende_projekte/QualiPSo/index.html (Abgerufen am 31.5.09)
- [Deprez und Alexandre 2008] Deprez, J.-C./ Alexandre, S.: "*Comparing Assessment Methodologies for Free/Open Source Software: Open BRR & QSOS*", 2008, Charleroi.
- [QSOSORG] "*Method for Qualification and Selection of Open Source software (QSOS)*", Version 1.6, 2006, <http://www.qsos.org/download/qsos-1.6-en.pdf>, (Abgerufen am 23.3.09)
- [WCAG2.0] *Web Content Accessibility Guidelines (WCAG) 2.0*; Stand: Dezember 2008, <http://www.w3.org/TR/WCAG20/>, (Abgerufen am 10.4.2009)

Mitwirkende:

Hanspeter Klappf und Reinhold Plösch

Studentische Mitwirkung:

Thomas Greifeneder

Michaela Kubovec

Brigitte Puffer

Daniel Sierninger