

# **Beschaffung agiler Softwareentwicklungen im Rahmen einer offenen WTO-Ausschreibung**

als

Abschlussarbeit des CAS ICT-Beschaffungen

an der

Wirtschafts- und Sozialwissenschaftlichen Fakultät  
der Universität Bern

eingereicht bei

**Daniel Wild und Dr. Matthias Stürmer**

Forschungsstelle Digitale Nachhaltigkeit

Institut für Wirtschaftsinformatik

von

Frauchiger Florian

von Bern

Studienadresse

Humboldtstr. 9

3013 Bern

(Tel. +41 79 577 72 73)

(E-Mail: [florian.frauchiger@gmail.com](mailto:florian.frauchiger@gmail.com))

Bern, 31.10.2016

## **Zusammenfassung**

Die agile Softwareentwicklung ist eine nützliche und beliebte Methodik, welche seit einigen Jahren vermehrt eingesetzt wird. Für öffentliche Institutionen die dem Beschaffungsrecht unterstellt sind, ist die Beschaffung einer solchen – vor allem im offenen Verfahren – jedoch nicht ganz einfach.

In der vorliegenden Arbeit wird einleitend auf die agile Softwareentwicklung eingegangen: Dazu wird nicht nur erklärt um was es sich dabei handelt, sondern es werden auch die Vorteile dieser Methodik erläutert, zudem wird die Entwicklung in den letzten Jahren aufgezeigt. Anschliessend wird das öffentliche Beschaffungsrecht erläutert: Wer ist diesem unterstellt, was bietet das Beschaffungsrecht für Möglichkeiten und wie hat sich das Beschaffungsvolumen in der letzten Zeit entwickelt. In einem zweiten Teil wird anschliessend erläutert was die Schwierigkeiten sind, wenn eine agile Softwareentwicklung über ein öffentliches, offenes Ausschreibungsverfahren beschafft werden soll. In einem dritten Teil werden die gleichen Themen, wie sie im zweiten Teil genannt wurden, wieder aufgegriffen und für jeden Teilbereich mögliche Lösungsansätze aufgezeigt.

# Inhaltsverzeichnis

<b>ZUSAMMENFASSUNG</b>	<b>I</b>
<b>INHALTSVERZEICHNIS</b>	<b>2</b>
<b>ABKÜRZUNGSVERZEICHNIS</b>	<b>4</b>
<b>1 EINLEITUNG</b>	<b>5</b>
1.1 Ausgangslage / Problemstellung	5
1.2 Zielsetzung	5
1.3 Abgrenzung	5
1.4 Vorgehen	6
<b>2 AGILE SOFTWAREENTWICKLUNG UND ÖFFENTLICHE BESCHAFFUNG</b>	<b>7</b>
2.1 Agile Softwareentwicklung	7
2.1.1 Was ist agile Softwareentwicklung?	7
2.1.2 Erfolg und Verbreitung agiler Entwicklungsprojekte	9
2.2 Öffentliches Beschaffungsrecht	10
2.2.1 Was ist das öffentliche Beschaffungsrecht?	11
2.2.2 Dialogverfahren und offenes Verfahren	12
<b>3 POTENTIELLE PROBLEMZONEN</b>	<b>14</b>
3.1 Erstellung der Ausschreibungsunterlagen	14
3.2 Unsicherheiten bei der Aufwandschätzung	14
3.3 Prototyping / Proof of Concept	15
3.4 Werkvertrag	15
3.5 Bezahlung nach Time & Materials	15
3.6 Softwarequalität, technische Schuld und Marginal Cost of Ownership	16
3.7 Bewertung der Angebote	18
3.8 Vorgabe der Entwicklungsmethode	18
3.9 Formulierung des Vertrags	19
<b>4 GESTALTUNG DER AUSSCHREIBUNGSUNTERLAGEN</b>	<b>20</b>
4.1 Granularität, Einteilung und Inhalt der Ausschreibungsunterlagen	20
4.2 Aufwandschätzung und Preisbewertung	21
4.3 Sprints als Optionen	24

4.4	Proof of Concept, Präsentation	24
4.5	Softwarequalität: Bewertung der technischen Schuld und der Marginal Cost of Ownership	25
4.6	Vertrag	27
<b>5</b>	<b>SCHLUSSFOLGERUNGEN UND AUSBLICK</b>	<b>28</b>
	<b>ANHANG A</b>	<b>29</b>
	<b>LITERATURVERZEICHNIS</b>	<b>31</b>
	<b>DARSTELLUNGSVERZEICHNIS</b>	<b>34</b>
	<b>SELBSTÄNDIGKEITSERKLÄRUNG</b>	<b>35</b>
	<b>VERÖFFENTLICHUNG DER ARBEIT</b>	<b>36</b>

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
BBL	Bundesamt für Bauten und Logistik
BöB	Bundesgesetz über das öffentliche Beschaffungswesen
CAS	Certificate of Advanced Studies
GATT	General Agreement on Tariffs and Trade
ICT	Information and Communication Technology
ISO	International Organization for Standardization
MCO	Marginal Cost of Ownership
Org-Vöb	Verordnung über die Organisation des öffentlichen Beschaffungswesens der Bundesverwaltung
PoC	Der Proof of Concept stellt eine Möglichkeit dar, die technische Machbarkeit (häufig anhand eines Prototyps) zu prüfen.
TCO	Total Cost of Ownership
TüV	Der technische Überwachungsverein umfasst eingetragene Verein, welche in der Rolle einer technischen Prüforganisation Kontrollen durchführen und Zertifikate verleihen können.
VöB	Verordnung über das öffentliche Beschaffungswesen
WTO	World Trade Organization

# **1 Einleitung**

Agile Softwareentwicklung und öffentliche Beschaffung – beides sind Begriffe, welche in Fachkreisen, wie auch in der Öffentlichkeit in den letzten Jahren eine gesteigerte Resonanz erfahren. Auf der einen Seite steht eine moderne und flexible Softwareentwicklungsmethode, auf der anderen Seite stehen vermeintlich unflexible und starre Gesetzestexte. Wie passt dies zueinander, wie können diese zwei Themen konstruktiv kombiniert werden? In den nachfolgenden Kapiteln wird diese Herausforderung schrittweise erörtert.

## **1.1 Ausgangslage / Problemstellung**

Die agile Softwareentwicklung befindet sich seit deren Entstehung im Vormarsch. Vor allem bei privaten Unternehmen erfreut sich die Agilität grosser Beliebtheit. Auch bei den Behörden ist das Bedürfnis nach agilen Entwicklungen vermehrt spürbar, die Durchführung wird durch die gesetzlichen Vorgaben des Beschaffungsrechts jedoch erschwert: Da bei einer klassischen Beschaffung nach dem GATT/WTO-Abkommen die Anforderungen bereits vorgängig bekannt gegeben werden müssen, stellt dies in der agilen Welt, welche Änderungen am Umfang und Inhalt auch während des Projekts zulässt, vor allem im Zusammenhang mit einer Beschaffung im offenen Verfahren eine nicht zu vernachlässigende Schwierigkeit dar. Es ist jedoch nicht unmöglich, auch unter Berücksichtigung des Beschaffungsrechts, eine erfolgreiche, agile Softwareentwicklung einzukaufen.

## **1.2 Zielsetzung**

Die vorliegende Arbeit soll aufzeigen, wo die Schwierigkeiten bei der Beschaffung einer agilen Softwareentwicklung im Rahmen einer offenen WTO-Ausschreibung liegen und entsprechende Lösungsansätze aufzeigen, wie diese gelöst werden können.

## **1.3 Abgrenzung**

Die Beschaffung von agilen Softwareentwicklungen im Rahmen eines Dialogverfahrens wird nicht beleuchtet. Es wird zwar aufgezeigt, dass das Dialogverfahren selten durchgeführt wird, auf die verantwortlichen Gründe wird jedoch nur am Rande eingegangen. Es werden zudem lediglich Lösungswe-

---

ge und -ideen aufgezeigt und keine direkt verwendbaren Vorlagen oder Textbausteine erarbeitet.

## **1.4 Vorgehen**

- A. Ideen-Abstimmung mit dem Experten Daniel Wild
- B. Umfang und Abgrenzung der Arbeit präzisieren
- C. Vorgehen strukturieren
- D. Terminplan erstellen
- E. Evaluation relevanter Literatur, Dokumente und Publikationen im Internet
- F. Analyse der Daten nach Evaluation – Konklusion aus Analyse
- G: Abgabe eines ersten Entwurfs an den Experten zum Review
- H: Einarbeiten der Rückmeldung aus dem Review
- I. Einleitung, Konklusion und Zusammenfassung

## 2 Agile Softwareentwicklung und öffentliche Beschaffung

In den nachfolgenden Abschnitten werden die wesentlichen Basisbegriffe erläutert. Dies soll zu einem besseren Verständnis führen, in welchem Zusammenhang diese Themen untereinander, und insbesondere zum Kern dieser Arbeit stehen.

### 2.1 Agile Softwareentwicklung

Agilität ist in der Softwareentwicklung nicht mehr wegzudenken. Doch was ist das überhaupt und welchen Einfluss hat es auf heutige Softwareentwicklungsprojekte? Nachfolgend ein kurzer Überblick.

#### 2.1.1 Was ist agile Softwareentwicklung?

Bei den klassischen Softwareentwicklungsmethoden wird ein neues System im Voraus sehr genau geplant und anschliessend in einem Schritt (oder ein paar wenigen Schritten) programmiert. Bei einer agilen Entwicklungsmethode wechseln sich hingegen kurze Planungs- und Entwicklungsphasen ab. Zuerst wird eine grobe Vision der Software entwickelt, wobei die Ziele in groben Zügen definiert und gewichtet werden. Danach werden einige dieser Ziele in einer ersten Version der Software realisiert und getestet. Wenn notwendig, können anschliessend die Anforderungen für nächste Versionen der Software angepasst werden. Dies hat den Vorteil, dass Änderungen und Anpassungen an den Anforderungen während des Projekts ohne grosse Kostenfolge oder Zeitverlust berücksichtigt werden können<sup>1</sup>. Diese Methode solle es ermöglichen, auch während der Entwicklung eines Programms noch Änderungen an den Anforderungen ohne grossen Aufwand vornehmen zu können.

Dass Änderungen während eines klassischen Projekts nach dem Wasserfallmodell sehr hohe Kosten verursachen können, zeigt die folgende Grafik.

---

<sup>1</sup> it-agile, 2016



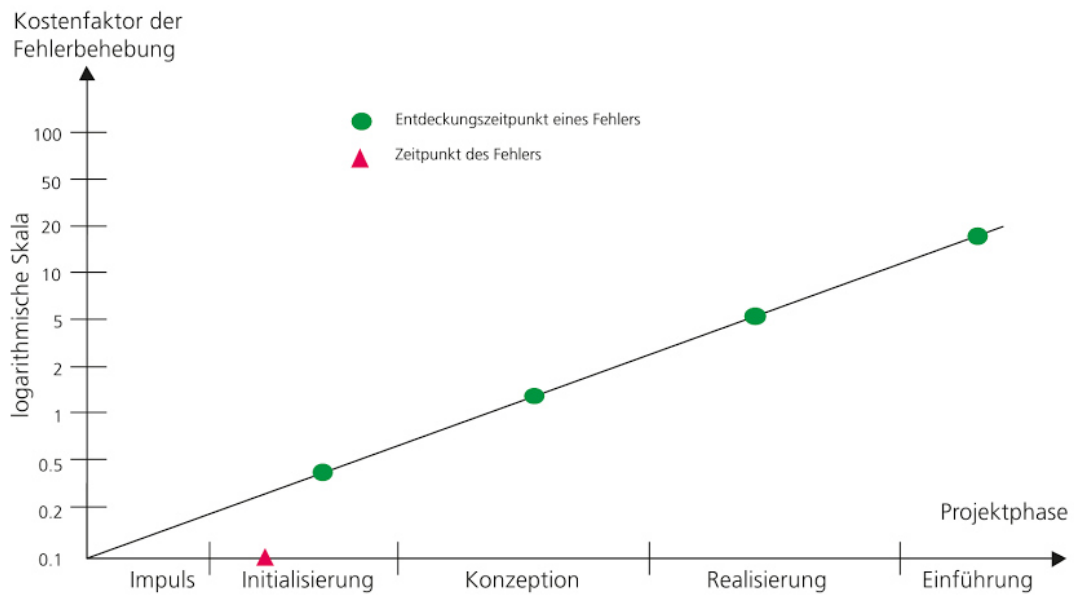


Abbildung 1: Relative Fehlerbehebungskosten, bezogen auf den Entstehungszeitpunkt<sup>2</sup>

Um daraus ein Beispiel abzuleiten: Bei einem Entwicklungsprojekt nach Wasserfallmodell wird gegen Ende der Entwicklungszeit die Software dem Kunden präsentiert. Wird dabei entdeckt, dass die Software noch früh entstandene Fehler aufweist, sind die Kosten für die Behebung dieser Fehler enorm hoch. Ein solcher „Fehler“ kann in diesem Zusammenhang aber auch als neu hinzugekommene Anforderung angesehen werden, oder als eine Anforderung, welche entgegen der ursprünglichen Planung angepasst werden muss.

Ist die Software durch die Fehler sogar unbrauchbar und auf Grund der hohen Behebungskosten nicht anpassbar, kann dies im schlimmsten Fall zu einem Totalverlust der im Projekt eingesetzten Mittel führen.

Ein solches Szenario ist bei einer agilen Vorgehensweise weniger wahrscheinlich: Da hier die Anforderungen nach jeder Iteration, resp. bei jedem Sprint angepasst werden können, sind Änderungen in den Anforderungen (resp. Fehler in der Planung) während des Projekts nicht so kostenintensiv, wie beim Wasserfallmodell. Zudem ist jede Version der Software lauffähig. Gehen die Mittel also bereits vor der Umsetzung aller geplanten Anforderun-

<sup>2</sup> Jenny 2003, S. 169, zitiert nach Gloor, 1985

gen aus, kann die Software trotzdem bis zu einem gewissen Grad eingesetzt werden.

### **2.1.2 Erfolg und Verbreitung agiler Entwicklungsprojekte**

Agil durchgeführte Projekte sind überdurchschnittlich erfolgreich: Gemäss Umfragen liegt die Erfolgsrate agiler Projekte bei 60%, resp. 70%, wohingegen die Erfolgsrate bei Projekten durchgeführt nach dem Wasserfallmodell lediglich auf 40%, resp. 45% zu liegen kommt<sup>3</sup>. Ungeachtet der leicht unterschiedlichen Ergebnisse lässt sich erkennen, dass mit der klassischen Methode weniger als die Hälfte der Projekte als erfolgreich bezeichnet wird, bei einer Durchführung mittels agiler Methoden dieser Wert jedoch bei deutlich über der Hälfte liegt.

Aus diesem Grund ist es nicht verwunderlich, dass Projekte in den letzten Jahren immer mehr mit agilen Entwicklungsmethoden durchgeführt wurden: Seit den Anfängen der Agilität im Jahr 2001 gab es eine enorme Zunahme an Unternehmen, welche agile Entwicklungsmethoden anwenden. Im Jahr 2010 setzten gemäss einer Umfrage 35% der befragten Unternehmen agile Methoden ein<sup>4</sup>, während es im Jahr 2013, resp. 2016 (je nach Quelle) bereits 84% sind<sup>5</sup>. Gleichzeitig zeigen die Resultate, dass die

Agilität häufig nicht in Reinform angewandt wird, sondern in einem Mix von traditionellen und agilen Praktiken und Tools<sup>6</sup>.

Doch auch wenn nicht überall in Reinform angewandt, zeigt die grosse Verbreitung, dass das Thema Agilität bei den heutigen Beschaffungen nicht vernachlässigt werden darf.

Die Verbreitung von Beschaffungen agiler Softwareentwicklungen beim Bund lässt sich aktuell leider nicht aussagekräftig messen, da diese nicht explizit als solche erfasst werden. Simap kann zwar nach gewissen Schlüsselworten durchsucht werden (wie „Scrum“ oder „agil“), damit können jedoch nicht alle Beschaffungen von agilen oder teilagilen Entwicklungsprojekten gefunden

---

<sup>3</sup> oose Innovative Informatik eG, 2009, S. 7; SwissQ 2016, S. 13

<sup>4</sup> Neumann Alexander, 2011

<sup>5</sup> Versionone, 2013, S. 4; SwissQ 2016

<sup>6</sup> SwissQ 2016, S. 15

werden<sup>7</sup>. Nachfolgend eine Übersicht der Beschaffungen in den letzten Jahren mit den Stichworten „Scrum“ und „agil“.

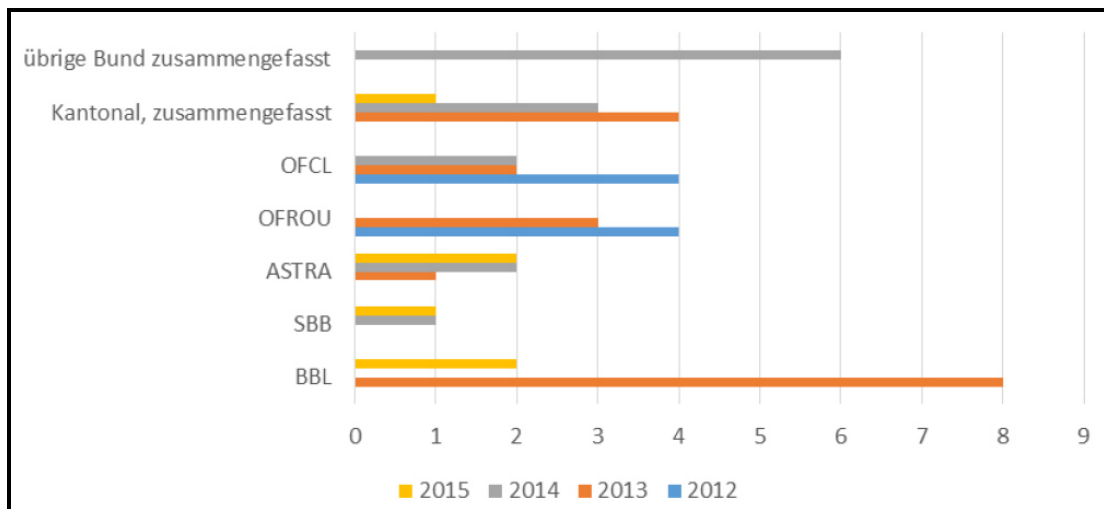


Abbildung 2: Beschaffungen zwischen 2012 und 2015 mit den Stichworten „Scrum“ oder „agil“<sup>8</sup>

Dies hat zur Folge, dass kein eindeutiger Trend bezüglich der Zunahme von Beschaffungen von agilen Entwicklungsprojekten nachgewiesen werden kann. Nichtsdestotrotz lassen die Zahlen aus der Privatwirtschaft Schlussfolgerungen auf den Verwaltungsbereich zu, übergeordnete Trends im privatwirtschaftlichen Bereich werden mit gewisser Verzögerung auch beim Bund aufgenommen. Unterstützung erhält diese Aussage nicht zuletzt auch durch die Themenwahl an den jährlichen IT-Beschaffungskonferenzen: Die Agilität ist dabei jedes Mal ein wichtiges Traktandum<sup>9</sup>.

## 2.2 Öffentliches Beschaffungsrecht

Das Bundesgesetz über das öffentliche Beschaffungswesen hat zum Ziel, öffentliche Beschaffungen transparent zu halten, den Wettbewerb unter den

<sup>7</sup> Blaser, 2015, S. 5

<sup>8</sup> Blaser 2015, S. 5, zitiert nach Stürmer, 2015

<sup>9</sup> Siehe beispielsweise das Programm der Konferenz von 2016 ([http://www.digitale-nachhaltig-keit.unibe.ch/ueber\\_uns/veranstaltungen/veranstaltungen\\_2016/it\\_beschaffungskonferenz\\_2016/index\\_ger.html](http://www.digitale-nachhaltig-keit.unibe.ch/ueber_uns/veranstaltungen/veranstaltungen_2016/it_beschaffungskonferenz_2016/index_ger.html)), 2015 (<https://www.ch-open.ch/it-beschaffungskonferenz>) und 2014 (<https://www.ch-open.ch/events/archiv/2015/180815-it-beschaffung/>)

Anbietern zu fördern, die Wirtschaftlichkeit der Vergaben zu sichern, sowie die Gleichbehandlung aller Anbieter zu gewährleisten. Nachfolgend eine Einführung ins Thema.

### **2.2.1 Was ist das öffentliche Beschaffungsrecht?**

Das öffentliche Beschaffungsrecht basiert auf dem WTO-Übereinkommen über das öffentliche Beschaffungswesen, die Schweiz hat zudem ein eigenes Abkommen mit der EU über bestimmte Aspekte desselben. National sind die Gesetze in verschiedenen Dokumenten verankert (BöB/VöB/Org-VöB u.a.). Diese kommen zum Tragen, sobald eine öffentliche Beschaffung vorliegt; das heisst, wenn der Staat auf dem Markt in Erfüllung seiner Aufgaben und unter Einsatz finanzieller staatlicher Mittel Bauleistungen, Lieferungen (Güter) oder Dienstleistungen beschafft. Die darin beschriebenen Texte zeigen auf, wie eine Beschaffung ablaufen hat und bieten dafür verschiedene Verfahren:

- **Offenes Verfahren** nach Art. 14 BöB: Alle interessierten Anbieter können an der Ausschreibung teilnehmen und ein Angebot einreichen.
- **Selektives Verfahren** nach Art. 15 BöB; Art. 12 VöB: Ein zweistufiges Verfahren, wobei nur die mittels einer Präqualifikation ausgewählten Anbieter ein Angebot einreichen dürfen.
- **Freihändiges Verfahren** nach Art. 16 BöB; Art. 13 und 36 VöB: Der Zuschlag wird direkt an einen Anbieter vergeben.
- **Dialogverfahren** nach Art. 20 BöB; Art. 26a VöB: Bei komplexen Beschaffungen oder der Beschaffung von intellektuellen Dienstleistungen kann der Auftraggeber die Lösungswege oder Vorgehensweise gemeinsam mit dem Anbieter (weiter-)entwickeln. Dies ist kein eigenständiges Verfahren, sondern kann als zusätzlicher Schritt im Kontext eines der anderen, offiziellen Verfahren angewendet werden. Dabei wird jeweils mit mehreren (mindestens aber drei) Anbietern ein eigenes Konzept erarbeitet, schlussendlich jedoch nur eines ausgewählt und umgesetzt.

- **Einladungsverfahren** nach Art. 35 VöB: Es müssten mindestens drei vergleichbare Offerten eingeholt werden, wobei ein Angebot ortsfremd sein muss.

Nachfolgend wird auf die für diese Arbeit relevanten Verfahren eingegangen.

### **2.2.2 Dialogverfahren und offenes Verfahren**

Grundsätzlich ist das Beschaffungsgesetz nicht für die Beschaffung agiler Entwicklungen konzipiert, auch wird sich dieses in naher Zukunft nicht in diese Richtung ändern.<sup>10</sup> Damit während dem stark eingeschränkten Beschaffungsverfahren im öffentlichen Bereich trotzdem eine gewisse Flexibilität beibehalten werden kann, wird im Allgemeinen das Dialogverfahren empfohlen.<sup>11</sup> Auch von Seiten der Anbieter wird ein der Einsatz des Dialogverfahrens gewünscht<sup>12</sup>.

In der Realität kommt das Dialogverfahren jedoch nur selten zur Anwendung: Gemäss Auskunft von Beschaffungsverantwortlichen Personen beim Bundesamt für Bauten und Logistik (BBL), der zentrale Beschaffungsstelle des Bundes, wurde das Verfahren seit dessen Kreation nicht mehr als fünf bis zehn Mal angewendet. Nachfolgend einige mögliche Gründe dafür.

Das Dialogverfahren

- stellt hohe Anforderungen (Kompetenzen) an die Beschaffungs- und Bedarfsstelle.<sup>13</sup>
- ist sehr aufwändig, es wird von einem Mehraufwand von 400% ausgegangen.<sup>14</sup>
- dauert dadurch wesentlich länger.
- rentiert sich auf Grund des hohen Aufwands auf Auftraggeber-, wie auch auf Anbieterseite erst bei einem Beschaffungsvolumen von über ca. 5 Mio. Franken. Allein 50% der Volumina von schweizerischen

---

<sup>10</sup> Blaser, 2015, S. 21

<sup>11</sup> Blaser, 2015, S. 27; Maduz, 2016, als Gastdozent im Rahmen des CAS ICT-Beschaffungen

<sup>12</sup> Grütter, 2016 ICT Beschaffungskonferenz 2016

<sup>13</sup> Fetz et al., 2012

<sup>14</sup> Fetz et al., 2012

Projekten im öffentlichen Bereich sind jedoch unterhalb 1 Mio. Franken.<sup>15</sup>

Erschwerend kommt hinzu, dass wie der folgenden Grafik entnommen werden kann, die Anzahl der Ausschreibungen in den letzten Jahren zugenommen hat.

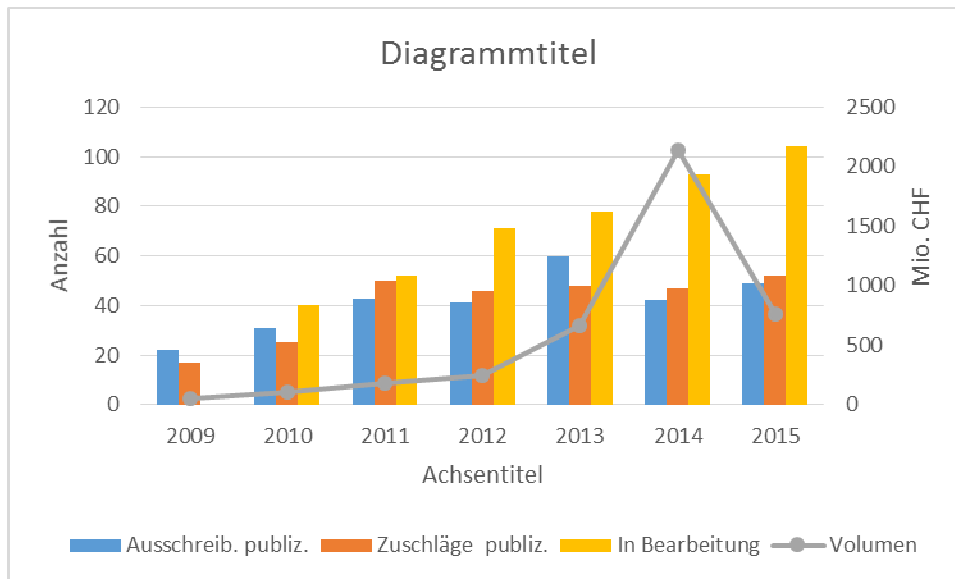


Abbildung 3: Entwicklung der Ausschreibungen seit 2009 im BBL<sup>16</sup>

Dieser Umstand, verbunden mit den gleichbleibenden Ressourcen beim Beschaffungspersonal (zumindest im BBL), ist die Möglichkeit zur Durchführung von Dialogverfahren auch dadurch begrenzt.

<sup>15</sup> Maduz, 2013

<sup>16</sup> Purtschert [intern], 2016, siehe Anhang A

### 3 Potentielle Problemzonen

Eine gemeinsame Lösungsfindung und die Kommunikation zwischen Auftraggeber und Anbieter ist ein wichtiger Bestandteil der agilen Entwicklung.<sup>17</sup> Bei einer offenen Ausschreibung ist dies in der ersten Phase (Erstellung der Ausschreibungsunterlagen) jedoch nur sehr eingeschränkt möglich. Nachfolgend werden die daraus entstehenden Schwierigkeiten beschrieben.

#### 3.1 Erstellung der Ausschreibungsunterlagen

Die Beschaffungsstelle muss sich in den Ausschreibungsunterlagen darauf festlegen, was beschafft werden soll, sowohl in Bezug auf die anbietende Firma, wie auch auf das gewünschte Produkt. Bei letzterem liegt die Schwierigkeit darin, einerseits die gewünschte Lösung gut zu beschreiben, andererseits diese nicht zu detailliert vorzugeben.<sup>18</sup> Eine zu detaillierte Vorgabe der Lösung macht einen Grossteil der Vorteile einer agilen Entwicklung zunichte.

#### 3.2 Unsicherheiten bei der Aufwandschätzung

Ein Hauptmerkmal einer agilen Entwicklung ist, dass das Endresultat noch nicht komplett bekannt ist, resp. sich während des Projekts noch verändern kann. Aus diesem Grund ist eine zentrale Herausforderung, den Gesamtaufwand eines Entwicklungsprojekts abschätzen und sich schliesslich verbindlich und vergleichbar offerieren lassen zu können. Eine denkbare Variante wäre zum Beispiel, dass man die Story Points aller User Stories schätzt und sich dafür einen Preis geben lässt. Dies scheitert jedoch an der Individualität von Story Points: Diese sind nämlich keine fest definierte Grösse, sondern werden von jedem Anbieter ein wenig anders geschätzt.<sup>19</sup> Deshalb ist es nicht empfehlenswert, dass der Auftraggeber die Komplexität – also die Anzahl Story Points – seiner eigenen User Stories selber festlegt und in der Ausschreibung vorgibt. Ausserhalb von öffentlichen Beschaffungen ist dies einfacher: User Stories können gemeinsam vom Auftraggeber und Lieferanten diskutiert und zusammen festgelegt werden. Da bei einem offenen Aus-

---

<sup>17</sup> Opelt et al., 2012, S. 6

<sup>18</sup> Maduz, 2016

<sup>19</sup> Opelt et al., 2012, S. 95

schreibungsverfahren vor dem Zuschlag abgesehen von der Fragerunde aber kein Kontakt zwischen den Parteien erlaubt ist, erschwert dies im öffentlichen Beschaffungsprozess die Einschätzung des Aufwands einzelner User Stories. Zudem ist es nicht möglich oder sinnvoll, bereits alle User Stories zu erarbeiten und mit der Ausschreibung zu veröffentlichen, es muss einen anderen Weg geben, wie die Anbieter eine Gesamtkostenschätzung abgeben können.

### **3.3 Prototyping / Proof of Concept**

Das Prototyping, resp. ein Proof of Concept (PoC) ist wichtig fürs Requirement Engineering und wird von den Anbietern zum Beispiel im Rahmen von Dialogverfahren auch sehr begrüsst.<sup>20</sup> Bei einer offenen Ausschreibung ist ein gemeinsames Erarbeiten eines Proof of Concepts jedoch nur bedingt möglich, da vor dem Zuschlag ein Kontakt mit dem Anbieter auch hier nur sehr eingeschränkt erlaubt ist. Die Schwierigkeit liegt hier also darin, einen Proof of Concept durchführen und bewerten zu können, ohne vorgängig Kontakt mit dem Anbieter zu haben.

### **3.4 Werkvertrag**

Es liegt in der Natur des Werkvertrags, dass das Resultat, die endgültige Lösung möglichst genau beschrieben wird. Zudem müssen entsprechende Abnahmekriterien definiert werden, welche vom Anbieter exakt erfüllt werden müssen. Dies steht in einem grossen Widerspruch zur Ausprägung einer agilen Entwicklung, wo ja genau das nicht erwünscht ist und nicht erwartet wird. Hier kann also nicht nach der klassischen Vorstellung eines Werkvertrags vorgegangen werden.

### **3.5 Bezahlung nach Time & Materials**

Gerade aus diesen Gründen kann es vorkommen, dass beim Anbieter lediglich die Arbeitszeit eingekauft wird (auch „Time & Materials“ genannt). Dies hat zwar den Vorteil einer grossen Flexibilität während der Entwicklung, da man sich nicht bereits auf gewisse Resultate festgelegt hat, birgt jedoch auch grosse Risiken in sich. Der Anbieter wird bei seinem Angebot seine besten

---

<sup>20</sup> Murkowski, 2016



Mitarbeiter anbieten; ist das Projekt aber erst mal im Gang, hat er dafür keinen Anreiz mehr. Vermutlich wird er nicht unfähige Mitarbeiter einsetzen, aber vielleicht mittelmässige. So kann das Projekt unter Umständen trotzdem „erfolgreich“ abgeschlossen werden, jedoch womöglich nicht in der gleichen Zeit oder mit dem gleich grossen Funktionsumfang, wie das sonst möglich gewesen wäre.<sup>21</sup> Bei der Gestaltung einer Ausschreibung gilt es also die Risikoverteilung genau abzuwägen.

### **3.6 Softwarequalität, technische Schuld und Marginal Cost of Ownership**

Unter der technischen Schuld versteht man kompliziert programmierten oder schlecht strukturierten Code. Der Wert, resp. der Verlust durch technische Schuld lässt sich beziffern: Es sind die Kosten die anfallen, um unzulänglich programmierten Code wieder zu verbessern, resp. die Qualität wieder auf das gewünschte Niveau zu bringen. Und wie bei einer Schuld im herkömmlichen, monetären Sinne gibt es auch hier Zinsen. Diese zeigen sich dadurch, dass im weiteren Lifecycle der Software Mehrkosten entstehen, wenn auf Grund der technischen Schuld die Produktivität bei Anpassungen an der Software und bei deren Wartung sinkt, resp. ein höherer Aufwand entsteht.<sup>22</sup> Für die Entstehung von technischer Schuld gibt es vor allem zwei Gründe:

- Bewusst: Wegen rascher, „Quick & Dirty“-Programmierung zum Einhalten eines Termins.
- Unbewusst: Wegen fehlendem Können oder Wissen.

Das bewusste Eingehen von technischer Schuld muss kein Zeugnis einer schlechten Arbeitsweise oder Firma sein. Wie nachfolgende Grafik zeigt, kann dies bewusst in Kauf genommen werden, um eine temporäre Produktivitätssteigerung zu erreichen, welche unter Umständen unabdingbar und wichtig für den weiteren Verlauf eines Projekts ist.

---

<sup>21</sup> Opelt et al, 2012, S. 39

<sup>22</sup> Müller, Eberhard, 2013, S.

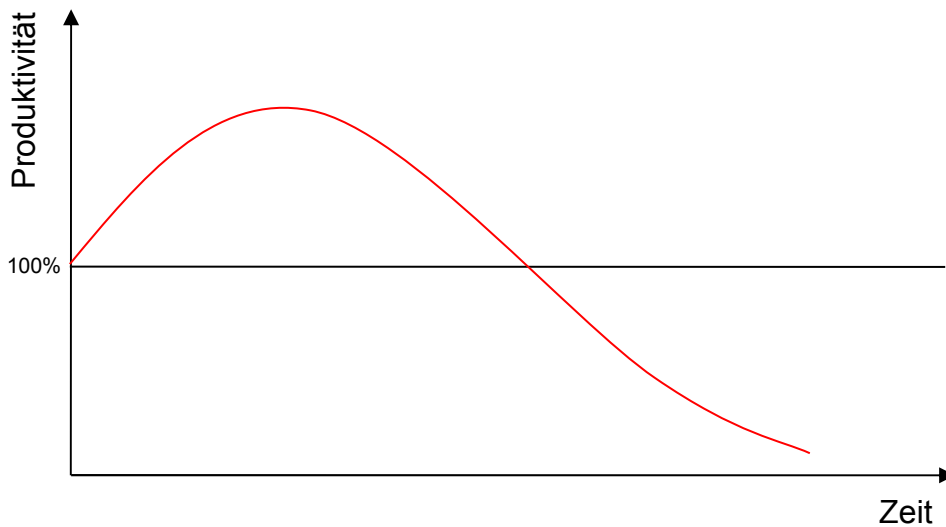


Abbildung 4: Kurzfristige Produktivitätssteigerung beim Eingehen technischer Schuld

Ob bewusst oder unbewusst: Die Folgen davon können gravierend sein. So kann es sein, dass bei einer schlecht programmierten Software eine einfache Änderung – wie zum Beispiel das Hinzufügen eines einfachen Dialogfelds – zu ungewöhnlich langen Entwicklungszeiten und damit auch zu hohen Kosten führen. Wie auch der Grafik zu entnehmen ist, sinkt die Produktivität, resp. steigen die Zinsen der technischen Schuld, je länger diese nicht behoben wird.<sup>23</sup> Dieser Umstand wird verschärft durch die Tatsache, dass die Entwicklungskosten lediglich 20%, während die Wartungskosten 80% der Kosten ausmachen.<sup>24</sup>

Vladimir Riecký hat anlässlich der IT-Beschaffungskonferenz 2016 darauf hingewiesen, dass die technische Schuld und damit auch die Marginal Cost of Ownership häufig vernachlässigt wird (unter der Marginal Cost of Ownership (MCO) versteht Riecký die Kosten, welche bei der Wartung, einer Anpassung oder Erweiterung einer Software anfallen und bei einer schlechten Softwarequalität entsprechend höher sind). Stattdessen wird nur auf die Total Cost of Ownership (TCO) geachtet, also auf den Gesamtpreis der Anschaffung. Auf dem Papier fällt das schlussendlich womöglich gar nicht auf, solange der Gesamtpreis nicht überschritten wird. Letztendlich würde der

---

<sup>23</sup> Hachmann, 2014

<sup>24</sup> Hahn, 2016

Auftraggeber bei einer qualitativ einwandfreien Software aber mehr Leistung oder mehr Funktionen für sein Geld erhalten.

Des Weiteren führt Riecicky aus, dass der Anbieter auch gewisse Vorteile aus einem schlecht strukturierten Code gewinnen kann. So kennt nur er den Code gut genug, um diesen weiter (mit einer akzeptablen Produktivität) pflegen zu können. So steigt die Wahrscheinlichkeit, dass Folgeaufträge wieder an ihn vergeben werden, ein Vendor Lock-In entsteht. Zudem ergeben sich aus der Erhaltung eines gewissen Qualitätsgrads auch Kosten: Je nach Projekt und Branche bewegen sich diese zwischen 10% und 3%. Es ist deshalb nicht verwunderlich, wenn die Anbieter aus Kostengründen absichtlich nicht eine zu hohe Softwarequalität abliefern.

### **3.7 Bewertung der Angebote**

Auf Grund der in den vorherigen Kapiteln genannten Unsicherheiten und Schwierigkeiten gestaltet sich die Bewertung entsprechend schwierig. Etliche Aspekte könnten in einem gemeinsamen Gespräch mit dem Anbieter geklärt werden, im offenen Verfahren ist dies durch die restriktiven Kommunikationsvorgaben jedoch erschwert. Eine vernünftige und faire Bewertung ist also nur möglich, wenn für alle obgenannten Punkte eine Lösung gefunden und in der Ausschreibung entsprechend umgesetzt wurde.

### **3.8 Vorgabe der Entwicklungsmethode**

Der Auftraggeber wünscht sich zwar ein agiles Vorgehen, jedoch gibt es auch hier Unterschiede, wie einzelne Firmen dieses Konzept verstehen und einsetzen. So gibt es einerseits nebst dem allseits bekannten Vorgehensmodell Scrum noch weitere, wie etwa Kanban oder eXtreme Programming (XP), andererseits werden diese Vorgehensmodelle von den Anbietern zum Teil auch unterschiedlich eingesetzt. Der Auftraggeber sollte hier also die Definition der Methode (z.B. Sprintlänge, Grösse der Teams etc.) nicht zu stark vorgeben, zumal die agilen Methoden häufig auch in Kombination mit klassischen Entwicklungsmethoden eingesetzt werden.<sup>25</sup>

---

<sup>25</sup> SwissQ, 2016, 26

---

### **3.9 Formulierung des Vertrags**

Alle obgenannten, kritischen Punkte zeigen ihre Wirkung bei der Erstellung des Vertrags. Denn hier versucht sich der Auftraggeber abzusichern und den Anbieter in die Pflicht zu nehmen. Dies ist jedoch nur möglich, wenn die Vorarbeiten korrekt vorgenommen und die heiklen Punkte vollständig und sinnvoll analysiert und gelöst wurden. Ein Vertrag kann dementsprechend nur so gut sein, wie die grundlegenden Fragen vorgängig gelöst wurden und wie dieser unter Berücksichtigung der Eigenschaften vom Angebot des Zuschlagsempfängers erstellt wurde.

## 4 Gestaltung der Ausschreibungsunterlagen

Die Gestaltung (auch genannt das Design) der Ausschreibungsunterlagen ist ein wichtiger Bestandteil einer erfolgreichen Beschaffung. Unter der Gestaltung versteht man sowohl den Aufbau der Ausschreibung (wie das Verfahren, die Gewichtung, die Anzahl Lose und Zuschläge, die Bewertung), als auch den Inhalt (die einzelnen Kriterien und Vorgaben, die möglichen zu bewertenden Komponenten wie Referenzen, eine Präsentation, ein PoC etc.). Nachfolgend werden potentielle Lösungswege aufgezeigt, welche dabei helfen sollen, erfolgreich eine agile Softwareentwicklung im offenen Verfahren zu beschaffen.

### 4.1 Granularität, Einteilung und Inhalt der Ausschreibungsunterlagen

Ein Hauptaspekt den es beim Erstellen der Unterlagen zu beachten gibt, ist die Konzentration auf die Frage des „Was“ und nicht des „Wie“.<sup>26</sup> Das „Was“ kann z.B. mittels einer Vision umschrieben werden.<sup>27</sup> Bezüglich dem „Wie“ sollte jedoch nicht zu viel vorgegeben werden: Da die Anbieter agiler Softwareentwicklungen jeweils individuelle Vorgehensmodelle haben (es gibt Unterschiede bei den Sprintlängen und -größen, Teams, Abläufen etc.), ist dies nicht sinnvoll. Durch zu enge Vorgaben werden die Anbieter dazu gedrängt, potenziell ungewohnte Strukturen und Abläufe zu übernehmen, was sich wiederum negativ auf das Projekt auswirkt.

Um dennoch ein gutes „Wie“ erreichen zu können, muss das Projekt mit einem qualifizierten Anbieter und einer guten Governance durchgeführt werden können. Das heisst mit guten Steuerungs- und Regelungssystemen, wie auch einer guten Aufbau- und Ablauforganisation im Projekt. Damit ein solcher Anbieter gefunden und eine gute Governance erreicht werden kann, muss vor allem die Qualifikation des Anbieters über die Kriterien gesteuert werden. Es ist also Wert auf die Standardkriterien wie Unternehmensgröße und Referenzen, als auch auf die Fachgebiete des Anbieters, die Kapazitäts-

---

<sup>26</sup> Maduz, 2013

<sup>27</sup> Opelt et al., 2013, S. 46

ten und Verfügbarkeiten des Projektteams, die Frameworks und Tools, wie auch die Skills und Erfahrungen der Mitarbeiter zu legen.<sup>28</sup>



*Fokus auf die Qualifikation des Anbieters, das Projektteam und die Governance im Projekt legen.*

Weiter kann vom Anbieter bereits verlangt werden, dass er eine grobe Architektur und Vorschläge zur Lösung bringt, so wie Aspekte bezüglich des Projekts (Team, Governance, etc.) aufzeigt. Diese Aspekte können als Zusatzkriterien aufgenommen werden. Generell empfiehlt es sich bezüglich Anforderungen, dies in funktionale und nichtfunktionale Anforderungen zu trennen.<sup>29</sup>



*Funktionale und nichtfunktionale Anforderungen trennen.*

## 4.2 Aufwandschätzung und Preisbewertung

Damit die Komplexität nicht vom Auftraggeber selbst festgelegt werden muss, ist Folgendes denkbar: Einige User Stories werden detailliert und komplett gemäss den agilen Vorgaben aufgezeichnet und den Ausschreibungsunterlagen beigelegt, diese gelten als Reference User Stories. Vom Anbieter wird verlangt, dass dieser die Komplexität in Anzahl Story Points bewertet und auch einen Preis pro Story Point angibt. Die Multiplikation dieser beiden Werte, kumuliert mit den Produkten aus den Werten der anderen User Stories gibt dann einen Preis für diese Reference User Stories. Nachfolgende Tabelle soll diesen Ansatz beispielhaft und reduziert darstellen, die hellgrün hinterlegten Felder wären dabei vom Anbieter auszufüllen.

---

<sup>28</sup> Tomczak, 2015, S. 26

<sup>29</sup> SwissQ, 2014

Reference User Story	Anzahl Story Points	Preis pro Story Point (CHF)	Preis pro User Story
US1: Dateneingabe	7	2'340	16'380
US2: Diagramm ge- nerieren	5		11'700
US3: Datenanzeige	3		7'020
Total			35'100

Tabelle 1: Einschätzung der Komplexität von User Stories

Es ist hierbei empfehlenswert mindestens 10 User Stories anzugeben, dies hilft dem Anbieter bei der Einschätzung im gemeinsamen Kontext.<sup>30</sup> Eine solche Vorgehensweise hat einerseits den Vorteil, dass der Anbieter mit seiner gewohnten Grösse von Story Points arbeiten kann, andererseits gibt es dem Auftraggeber ein Gefühl für die Einschätzungsart des Anbieters. Dies ist vor allem wichtig, damit es bei der Durchführung des Projekts und der Einschätzung von weiteren, zum aktuellen Zeitpunkt noch nicht definierten User Stories zu keinen Meinungsverschiedenheiten kommt und die Einschätzung des Anbieters als realistisch wahrgenommen werden kann. Die vom Anbieter in dieser Tabelle angegebenen Werte können schliesslich direkt im Vertrag aufgenommen werden.



*Einige User Stories (min. 10) in den Ausschreibungsunterlagen beschreiben und vom Anbieter bezüglich deren Komplexität in Story Points bewerten lassen.*

Ausschlaggebend für die Bewertung in einer Ausschreibung ist jedoch nur der Gesamtpreis, wofür man jedoch wiederum den gesamten Aufwand kennen muss. Da noch nicht alle User Stories ausgearbeitet oder überhaupt bekannt sind, muss also eine Annahme getroffen werden. Eine Möglichkeit eine solche Schätzung zu erhalten ist per Hochrechnung.<sup>31</sup> Auf Grund der fundier-

<sup>30</sup> Wiegand, 2015

<sup>31</sup> Opelt et al., 2013, S. 95

ten Fachkenntnisse sollte es der Fachstelle möglich sein abzuschätzen, wie gross der Anteil der Reference User Stories vom Gesamtumfang ist. Dadurch lässt sich mittels Multiplikation der Komplexität das Volumen für das gesamte Projekt errechnen. Gegenüber dem Anbieter könnte somit zum Beispiel folgender Hinweis in den Ausschreibungsunterlagen platziert werden: „Der Auftraggeber geht davon aus, dass die in den Ausschreibungsunterlagen dargestellten Reference User Stories etwa 10% des gesamten Projekts ausmachen.“ Im Preisblatt könnte schliesslich der Preis für die Reference User Stories dazu verwendet werden den Preis für 100% des Werks hochzurechnen. Dazu werden die vom Anbieter geschätzten Story Points für die Reference User Stories automatisch hochgerechnet (in diesem Fall multipliziert mit 9), allenfalls kann der Anbieter noch einen anderen Preis für die restlichen Story Points angeben.

Restlicher Umfang	Anzahl Story Points <small>(User Stories 1-3, mal 9)</small>	Preis pro Story Point (CHF)	Preis für den Restumfang
<b>User Stories 4 - 30</b>	135	2'210	298'350
<b>Total</b>			<b>298'350</b>

Tabelle 2: Hochrechnung auf Gesamtaufwand und Preisbildung

Dieser Betrag, zusammengezählt mit dem Preis für die Reference User Stories, ergibt somit den Gesamtpreis, welcher bewertet werden kann.



*Gesamtaufwand auf Basis der Reference User Stories schätzen und gesamthaft anbieten lassen.*

Gibt es bei der Aufwandschätzung Bedenken bezüglich des Umfangs, kann zusätzlich eine optionale Reserve (z.B. +10% mehr Story Points) eingerechnet werden.



### 4.3 Sprints als Optionen

Damit sich der Auftraggeber absichern kann, gibt es die Möglichkeit einen Teil der Entwicklung als Grundauftrag zu definieren, den Rest optional.<sup>32</sup>



*Die ersten Sprints als Grundauftrag definieren, die restlichen Sprints (resp. Story Points) als Option.*

Es empfiehlt sich dabei, dass der Grundauftrag mindestens 2 Sprints umfasst: Zu Beginn eines Projekts kennen sich die Parteien nicht, es kann eine gewisse Eingewöhnungsphase geben. Die restlichen Sprints können optional vergeben werden, mit einem Kündigungsrecht zu jeder Zeit seitens Auftraggeber.<sup>33</sup> So kann einerseits sichergestellt werden, dass mit einem unzulänglichen Anbieter das Projekt nicht um jeden Preis fertig gestellt werden muss (ein vorzeitiger Abbruch ist bei der agilen Entwicklung zwar auch unschön, aber nicht so kritisch wie bei der klassischen Entwicklungsmethode, denn es gibt nach jedem Sprint eine lauffähige Version), andererseits kann das Projekt auch vor der Realisierung aller offerierten Story Points abgeschlossen werden, wenn der Funktionsumfang aus Sicht Auftraggeber als ausreichend betrachtet wird. Damit sich der Anbieter in so einem Fall nicht übervorteilt fühlt, kann z.B. eine Teilzahlung der nicht realisierten Story Points abgemacht werden.<sup>34</sup>

### 4.4 Proof of Concept, Präsentation

Da ein umfangreicher, in Zusammenarbeit erstellter PoC durch die eingeschränkte Kommunikation kaum möglich ist, kann ein PoC in einer reduzierten Version vom Anbieter verlangt werden. So ist es möglich z. B. ein oder zwei User Stories umzusetzen und als Teil einer Präsentation vorstellen zu lassen.

---

<sup>32</sup> Opelt et al., 2015, S. 54f

<sup>33</sup> Straub, 2015, S. 16

<sup>34</sup> Opelt et al., 2013, S. 60



*Zusammen mit dem Angebot einen Proof of Concept verlangen, resp. die Programmierung einer User Story.*

In der Präsentation ist es zudem möglich, den Anbieter eine grobe Architektur und Vorschläge zur Lösung, die eingesetzte Technologie, wie auch die Projektgovernance zusätzlich mündlich vorstellen zu lassen. Dies gibt einen guten Eindruck über die Arbeitsweise und die Qualifikation des Anbieters und kann als Teil der Bewertung aufgenommen werden.



*Den Anbieter eine grobe Architektur und Vorschläge zur Lösung präsentieren, und das Resultat in die Bewertung einfließen lassen.*

## **4.5 Softwarequalität: Bewertung der technischen Schuld und der Marginal Cost of Ownership**

Vladimir Rieckick empfiehlt in seinem Referat anlässlich der IT-Beschaffungskonferenz 2016 nicht nur die TCO zu beachten, sondern auch die MCO zu bewerten. Diese lassen sich zwar (vor deren Entstehung) nicht in einem Frankenbetrag quantifizieren, es gibt jedoch Möglichkeiten diese tief zu halten. So erläuterte etwa Marc André Hahn anlässlich auch der IT-Beschaffungskonferenz 2016 das in Deutschland entwickelte Sternesystem IT-TÜV. Dabei wird der Code vom TÜV anhand internationaler (ISO-) Standards geprüft und mit einem bis fünf Sternen bewertet. Fünf Sterne bedeuten in diesem Fall, dass die Qualität des programmierten Codes zu den besten fünf Prozent gehört. Dies hat enorme Auswirkungen auf die Wartung und Weiterentwicklung des Produkts: Die Fehlerbehebung ist beispielsweise bei einem System mit 4 Sternen drei Mal so schnell wie bei einem System mit 2 Sternen, während dem die gesamte Produktivität sogar um den Faktor 11 höher ist.

Für den Auftraggeber ergibt sich dadurch die Möglichkeit mittels entsprechender Kriterien (Eignungskriterien oder technische Spezifikation) zu verlangen, dass der Anbieter den Code mit einer gewissen Anzahl Sterne zu programmieren hat.



*Über Kriterien die Einhaltung eines gewissen Qualitätsstandards im Code mittels anerkannter Prüfmethoden verlangen.*

Auch Abstufungen sind denkbar: So könnte zum Beispiel die Qualität von 4 Sternen zwingend verlangt und in einem Zuschlagskriterium für das Abliefern von Code mit 5 Sternen mehr Punkte vergeben werden. Es kann aber auch mit dem Zeitpunkt der Prüfung gespielt werden: Je nach Projekt kann es zum Beispiel sinnvoll sein, dass die geforderte Qualität bei jeder Iteration erfüllt sein muss (dies verhindert jedoch unter Umständen eine temporäre Produktivitätssteigerung, siehe auch Kapitel 3.6), oder aber erst bei der Schlussabnahme. Bei letzterer Variante wäre der Anbieter frei zu entscheiden, wann im Projekt er die technische Schuld abbauen, oder diese fortlaufend gar nicht erst entstehen lassen will. Letztere Variante ist nicht ganz unkritisch, denn wie auch in der „richtigen“ Finanzwelt ist ein Aufhäufen von Schuld nicht ungefährlich. Denkbar wäre daher zum Beispiel auch die Verpflichtung zu einer gewissen Qualität bei jedem zweiten Sprint. So hat der Anbieter nach wie vor die Möglichkeit bei Engpässen einen Teil „quick and dirty“ zu programmieren, muss diesen Code jedoch in absehbarer Frist wieder auf ein akzeptables Niveau bringen.

Nebst der Methode mit IT-TÜV gibt es noch weitere, automatisierte Messtools, welche die Software auf gewisse Aspekte (Softwarearchitektur, Einhaltung von Code Conventions, Kompatibilität mit gewissen Technologien oder Sprachen, etc.) prüfen, so zum Beispiel PMD, Checkstyle oder xradar.<sup>35</sup> Je nach Wichtigkeit der Softwarequalität kann es in einem weiterführenden Schritt aber auch sinnvoll sein, den Code von einem externen Unternehmen prüfen zu lassen. Wie auch immer eine Qualitätskontrolle vorgesehen ist, diese muss zwingend vorgängig definiert und vorgegeben<sup>36</sup>, sowie vertraglich festgelegt werden. Weiter sollte (auch vertraglich) festgelegt werden, welche Massnahmen bei Nichteinhaltung dieser Qualitätsvorgaben getroffen

---

<sup>35</sup> Fleischer, 2007, S. 43f

<sup>36</sup> Riecicky, 2016

werden. So kann zum Beispiel eine kostenlose Nachbesserung verlangt, oder die Bezahlung einer Pönale in Betracht gezogen werden.



*Die Einhaltung der Qualität des Codes vertraglich absichern und Massnahmen bei Nichteinhaltung definieren.*

## 4.6 Vertrag

Der Vertragsentwurf in den Ausschreibungsunterlagen sollte nebst dem üblichen Grundgerüst eines Vertrags (mit anwendbarem Recht, Kündigungsmöglichkeiten etc.) nicht zu viele Vorgaben machen. Wie bereits im Kapitel 4.1 erwähnt, muss dem Anbieter die Möglichkeit gelassen werden, seine Vorschläge und Arbeitsweise in das Projekt einbringen und in den Vertrag einfließen lassen zu können. Dies betrifft im agilen Umfeld vor allem Themen wie die Methodik, die Dauer von Sprints, die Regelmässigkeit von Meetings, Entscheidungsfindung etc.



*Die Methodik und die Governance (Eskalationsverfahren, Dauer von Sprints, Velocity, Meetings, Reviews, Quality Gates, Abnahmeverfahren etc.) im Vertrag festlegen.*

Der Vorschlag des Anbieters muss jedoch nicht unverändert übernommen werden, sondern kann z.B. im Rahmen einer Nachverhandlung besprochen, angepasst und implementiert werden.

Weiter sollten im Vertrag die in den vorigen Kapiteln behandelten Themen aufgefangen und definiert werden, dazu gehören unter anderem

- die Qualität des Codes,
- der Umfang der Story Points,
- die Verbindlichkeit von Sprint Reviews,
- die Einhaltung von Quality Gates,
- der Umgang mit Verzögerungen.

## 5 Schlussfolgerungen und Ausblick

Diverse Quellen befassen sich mit der Frage, wie agile Softwareentwicklungen im öffentlichen Bereich beschafft werden können. Es gibt einige gute Ansätze, jedoch deckt keine der in dem vorliegenden Dokument behandelten Quellen alle Aspekte, resp. genau die Fragen und (gesetzlichen) Anforderungen, welche sich bei der öffentlichen Beschaffung im Bereich der schweizerischen Bundesverwaltung im offenen WTO-Verfahren stellen. Häufig wird z.B. vernachlässigt, dass im offenen Verfahren ein Kontakt mit dem Anbieter nur sehr bedingt möglich ist, eine Besprechung der Inhalte vor der Publikation der Ausschreibung ist ganz ausgeschlossen. Teilweise äussern sich die Quellen auch nur sehr oberflächlich zu den Themen, resp. geben nur grobe Tipps oder Hilfestellungen. Es gibt also noch keine Beschreibung oder Anleitung, welche von der Gestaltung der Ausschreibung, über die (bedingt mögliche) Verhandlung, bis hin zur Vertragserstellung alle Aspekte hilfreich und detailliert aufzeigt.

In den vorangehenden Kapiteln konnte aufgezeigt werden, dass die Beschaffung einer agilen Softwareentwicklung im offenen Verfahren zwar möglich ist, jedoch eine fast noch grössere Herausforderung darstellt, als die Beschaffung von anderen Gütern oder Dienstleistungen. Es konnte allerdings auch erläutert werden, dass unter Anwendung einer Kombination von einzelnen Erkenntnissen aus den unterschiedlichen Quellen eine solche Beschaffung durchaus erfolgreich sollte durchgeführt werden können.

In einem weiterführenden Schritt wäre es aus Sicht des Autors denkbar, die hier aufgezeigten Lösungsansätze noch konkreter und detaillierter zu vertiefen und für den praktischen Einsatz im strategischen Einkauf beim BBL aufzubereiten.

## Anhang A

### Aktennotiz

Datum: 31-12-2014

Verfasser: Fritz Purtschert

31.10.2016 11:31

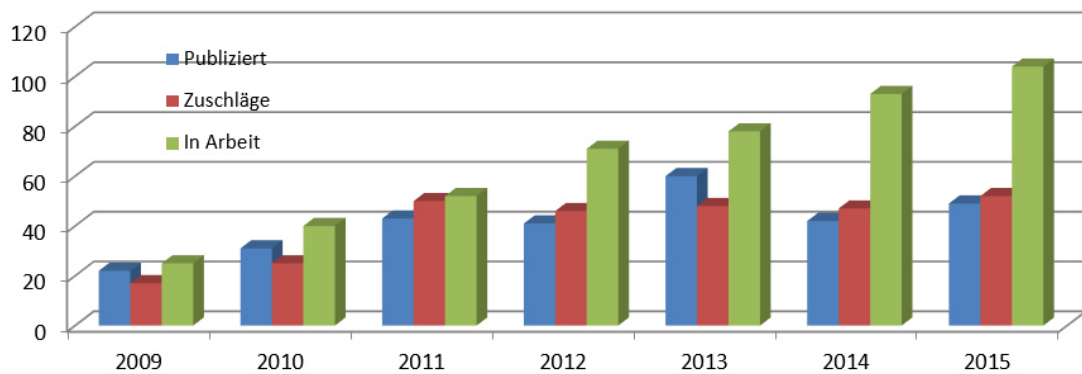
### BBI: Zahlen 2009 bis 31-12-2015

#### 1. Die "nackten" Zahlen: 2009 - 2015

Umsatzzahlen in Mio.CHF	2009	2010	2011	2012	2013	2014	2015	Bemerk.
BT Umsatz inkl. Kopiergebühren	16.4	20.9	17.8	21.3	22.4	25.4	25.5	nur SAP
HW Umsatz	28.6	26.0	22.4	20.9	19.3	12.4	12.7	nur SAP
SW Umsatz	29.4	17.3	17.8	28.2	26.8	27.6	54.7	nur SAP
Wartung HW	13.4	13.7	14.0	15.0	13.3	12.9	12.5	nur SAP
Pflege SW	59.8	63.6	61.6	59.6	60.0	60.4	45.9	nur SAP
Standard HW	85.6	60.0	68.0	44.8	35.4	45.5		nicht SAP
DL-Verträge An- zahl	84	72	137	181	153	128	101	nicht SAP
Umsatz Mio. CHF	50.4	83.7	86.1	133.2	126.4	210.5	98.7	
WTO Freihänder	42	48	81	54	52	75	69	ohne Ausschr.
Umsatz Mio. CHF	92.4	107.0	115.3	91.9	127.8	229.7	182	
WTO Zuschläge	17	25	50	46	48	47	52	ohne Freihänd
Umsatz Mio. CHF	53.2	104.1	179.6	242.9	664.1	2'137	759.2	

#### 2. WTO Ausschreibungen im BBI: 2009 - 2015

	2009	2010	2011	2012	2013	2014	2015	Bemerk.
Ausschreib. publiz.	22	31	43	41	60	42	49	
Zuschläge publiz.	17 53.2	25 104.1	50 179.6	46 242.9	48 664.1	47 2'137	52 759.2	inkl. M. Zuschl.
In Bearbeitung	~ 25	40	52	71	78	93	104	



- Der Betreuungsaufwand pro Ausschreibung / pro Geschäft nahm ab Jahr 2012 erheblich zu; die qualitativen Anforderungen sind gestiegen / Fragen komplexer.  
mehr einzelne Lose, grössere Zuschlagssummen usw.
- Anhäufung von Grossprojekten; FISCAL-IT, IT-Dienstleistungen für alle Departemente, Layer 2, UCC usw.

### 3. Geschäfte IT-Dienstleistung im BBI: 2009 - 2015

Dieser Geschäftsbereich hat sich in den letzten Jahren besonders stark entwickelt.

		2009	2010	2011	2012	2013	2014	2015
Anzahl Verträge	DL-	84	72	137	181	153	128	101
Umsatz Mio. CHF		50.4	83.7	86.1	133.1	126.4	210	98.7

### 4. IT – Wartung im BBI: 2009 - 2015

Mehrarbeit in den vergangenen Monaten und Projekte wie "Einsparungen Wartungskosten im FUB. Etliche Einsparungen wurden erzielt (vor allem bei SW), dafür kamen neue Wartungsverträge dazu.

		2009	2010	2011	2012	2013	2014	2015
Hardware Mio. CHF		13	14	14	15	13.3	12.9	12.5*
Software Mio. CHF		60	64	62	59.6	60.0	60.4	45.9*

Ab 2015 viele Kleinwartungsverträge ersetzt durch "normale" IT-Bestellungen \*

## Literaturverzeichnis

Blaser Robert (2015): „*Beschaffung agiler Entwicklungsprojekte oder Agile Beschaffungen?*“. Abschlussarbeit im Rahmen des CAS ICT-Beschaffungen.

Fleischer André (2007): „*Messen & Metriken: Ist Qualität messbar?*“. <http://xradar.sourceforge.net/downloads/metrics-isQualityMeasureable-de.pdf>. Abgerufen am 08.10.2016.

Fetz Marco, Schafer Bruno, Sutter Stephan, Maduz Reto (2016): „*Rahmenbedingungen für agile Softwarebeschaffung - Fachkonferenz IT-Beschaffung*“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2012. [http://www.ch-open.ch/fileadmin/user\\_upload/events/itbeschaffungskonferenz2012/09\\_AgileSoftwareBeschaffung.pdf](http://www.ch-open.ch/fileadmin/user_upload/events/itbeschaffungskonferenz2012/09_AgileSoftwareBeschaffung.pdf).

Grüter Franz (2016): „*Schlanke und flexible Beschaffungen*“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2016. <https://www.youtube.com/watch?v=92ESQ44A36g>.

Hachmann Ilona (2014): „*Technische Schuld(en) im Projekt*“. <http://blog.setzwein.com/2014/12/11/technische-schulden-im-projekt/>. Abgerufen am 07.10.2016.

Krypczyk Veikko (2014): „*Softwarequalität – so misst und verbessert man Software*“. <https://entwickler.de/online/agile/softwarequalitaet-so-misst-und-verbessert-man-software-114867.html>. Abgerufen am 08.10.2016.

Maduz Reto (2016): „*Agile Beschaffungen: Best Practices*“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2013. [https://www.youtube.com/watch?v=xRnJBV\\_08aw](https://www.youtube.com/watch?v=xRnJBV_08aw).



Murkowski Philipp (2016): „*Bessere Requirements durch Prototyping*“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2016. [http://www.iwi.unibe.ch/unibe/portal/fak\\_wiso/a\\_bwl/inst\\_wi/abt\\_digital/content/e90975/e462798/e462845/e462863/e463040/e473709/e473725/e474149/05bk16-prototyping-pmurkowsky\\_ger.pdf](http://www.iwi.unibe.ch/unibe/portal/fak_wiso/a_bwl/inst_wi/abt_digital/content/e90975/e462798/e462845/e462863/e463040/e473709/e473725/e474149/05bk16-prototyping-pmurkowsky_ger.pdf).

Müller Felix, Eberhard Wolff (2013): „*Qualitätsinvestitionen statt technischer Schulden*“. <http://www.heise.de/developer/artikel/Qualitaetsinvestitionen-statt-technischer-Schulden-2063864.html>. Abgerufen am 07.10.2016.

Neumann, Alexander (2011): „*10 Jahre Agiles Manifest – zur Geburt agiler Softwareentwicklung*“. <http://www.heise.de/developer/meldung/10-Jahre-Agiles-Manifest-zur-Geburt-agiler-Softwareentwicklung-1188299.html>. Abgerufen am 10.09.2016.

N.N. (2016): „*Software Development 2016*“. Umfrage, Trends und Benchmarks von SwissQ. [Kann kostenlos bestellt werden unter <http://report.swissq.it/de/>].

N.N. (N.N): „*Was ist agile Produktentwicklung?*“. Erklärungen zur Agilität auf der Webseite von it-agile. <https://www.it-agile.de/wissen/einstieg-und-ueberblick/was-ist-agile-produktentwicklung/>. Abgerufen am 15.10.2016.

Opelt Andreas, Gloger Boris, Pfarl Wolfgang, Mittermayr Ralf (2012): „*Der agile Festpreis – Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge*“. München: Hanser.

Riecký Vladimir (2016): „*Lieber Product Owner, dein Business Case fiel leider der technischen Schuld zum Opfer*“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2016. <https://www.youtube.com/watch?v=0iVQTxEoen4>.

Tomczak Artur (2015): „WTO-Ausschreibung für App-Entwicklung, „agile Beschaffung“, Mini-Tender, agiles Umfeld“. Präsentation im Rahmen der IT-Beschaffungskonferenz 2015.

<https://www.ch->

[o-](https://www.ch-)

[pen.ch/fileadmin/user\\_upload/events/itbeschaffungskonferenz2015/16\\_Referat\\_A\\_Tomczak\\_ScrumBeschaffung\\_BK.pdf](https://www.ch-).

Toth Stefan, Vogenschow Uwe, Witter Markus (2009): „*Einfluss klassischer und agiler Techniken auf den Erfolg von IT-Projekten*“.

<http://www.oose.de/wp->

[content/uploads/2012/06/Ergebnisbericht\\_Projektmanagementstudie.pdf](http://www.oose.de/wp-content/uploads/2012/06/Ergebnisbericht_Projektmanagementstudie.pdf).

Wiegand Sven (2015): „*Story-Points: Normierung der Skala und Planung*“.

<http://agilmanagen.de/story-points-normierung-und-planung/>. Abgerufen am 24.10.2016.

## Darstellungsverzeichnis

### Abbildungen

Abbildung 1: Relative Fehlerbehebungskosten, bezogen auf den Entstehungszeitpunkt.....	8
Abbildung 2: Beschaffungen zwischen 2012 und 2015 mit den Stichworten „Scrum“ oder „agil“ .....	10
Abbildung 3: Entwicklung der Ausschreibungen seit 2009 im BBL .....	13
Abbildung 4: Kurzfristige Produktivitätssteigerung beim Eingehen technischer Schuld.....	17

### Tabellen

Tabelle 1: Einschätzung der Komplexität von User Stories.....	22
Tabelle 2: Hochrechnung auf Gesamtaufwand und Preisbildung.....	23

## Selbständigkeitserklärung

„Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe o des Gesetzes vom 5. September 1996 über die Universität zum Entzug des aufgrund dieser Arbeit verliehenen Titels berechtigt ist.“

Bern, 31.10.2016



Florian Frauchiger

## Veröffentlichung der Arbeit

- ☒ Hiermit erlaube ich, meine Arbeit (ohne Anhang A) auf der Website der Forschungsstelle Digitale Nachhaltigkeit zu veröffentlichen.
- ☐ Hiermit erlaube ich, meine Arbeit den anderen Teilnehmenden des CAS ICT-Beschaffungen (alle Jahrgänge) über ILIAS zugänglich zu machen.
- ☐ Ich möchte auf eine Veröffentlichung meiner Arbeit verzichten.

Die Benotung der Arbeit erfolgt unabhängig davon, ob die Arbeit veröffentlicht werden darf oder nicht.

Bern, 31.10.2016



Florian Frauchiger