

Analyse, Konzeptionierung und prototypische Implementierung einer Anwendung für Software-Monetarisierung im Intralogistiksoftwareumfeld der SYSKRON GmbH

Franziska Heigl B.Sc.
Professor Dr.-Ing. Frank Herrmann
Ostbayerische Technische Hochschule Regensburg
Innovationszentrum für Produktionslogistik und Fabrikplanung (IPF)
E-Mail: Franziska_Heigl@web.de und Frank.Herrmann@OTH-Regensburg.de

Schlüsselwörter:

Software-Monetarisierung, Lizenzierung, Lizenz, Dongle, Lizenz-File

Abstract—Die vorgelegte Bachelor-Thesis befasste sich schwerpunktmäßig mit dem Thema Software-Monetarisierung. Die Frage nach der Umsetzung einer geeigneten Preisstrategie ist für Unternehmen ebenso von Bedeutung wie die Vertriebsart. In der Vergangenheit wurde dem Thema des Öfteren zu wenig Aufmerksamkeit geschenkt. Man entschied sich in vielen Fällen zu fahrlässig für Strategien, bei denen der Profit auf der Strecke blieb. Die größten Probleme, die hinsichtlich einer falsch gewählten Strategie auftraten waren beispielsweise der Verlust von Einnahmen sowie die unkontrollierte Vergabe von Lizenzen oder Berechtigungen. Es wurden Lizenzen freigeschaltet, die der Kunde nicht bezahlen musste. Des Weiteren wurde eine Lizenzgenerierung von nicht befugten Mitarbeitern nicht unterbunden. Ein zusätzlicher Aspekt der im heutigen Zeitalter nicht unterschätzt werden darf, ist der Schutz der Software vor Diebstahl. Heutzutage werden Unternehmen von vielen Faktoren positiv gedrängt, z.B. wünschen sich viele Kunden flexiblere Lizenzmodelle. Dadurch sehen sich Unternehmen gezwungen, sich ausführlich mit dem relevanten Thema auseinanderzusetzen. Aus diesem Grund war das Ziel der Arbeit eine prototypische Anwendung für Software-Monetarisierung zu implementieren. Der Erreichung des Zieles ging die Entwicklung eines Konzeptes voraus. Dieses basiert auf vorherigen Analysen, in welcher geeignete Lizenzierungsmechanismen analysiert wurden. Die entstehende Anwendung soll es dem Unternehmen ermöglichen, seine aktuell vergebenen Lizenzen und Features kontrollieren, überwachen und verwalten zu können.

Das Thema Software-Monetarisierung bzw. Lizenzierung wurde gewählt, um spannende Einblicke in eine neue Thematik zu gewinnen. Diese Aufgabenstellung ermöglichte es, sich mit einem essenziell wichtigen Thema im IT-Umfeld zu beschäftigen. Für jedes Software-Unternehmen ist das Thema Lizenzierung unumgänglich. Der Praxisbezug sowie der Erwerb von fachlichen Kenntnissen hat die Annahme der Arbeit nur weiter bestärkt.

I. THEORETISCHER TEIL

A. Allgemeine Begriffsdefinitionen

Im Verlauf der Artikels sind fünf zentrale Begriffe, die kontinuierlich auftauchen, erkennbar. Da diese nicht für alle selbstverständlich erscheinen, werden sie zu Beginn erläutert. Zu diesen zählen Software-Monetarisierung, Features bzw. Module, Lizenz, Lizenzierung und Lizenzierungsarten.

Zunächst steht der Begriff „Software-Monetarisierung“ im Vordergrund. Dieser Begriff beschäftigt sich einerseits mit Marktanforderungen, andererseits auch mit der Maximierung von Software-Erlösen. Vor allem Letzteres ist für Unternehmen von enormer Bedeutung. Um den Absatz zu erhöhen, sollten die Methoden so konstruiert werden, dass sie den Bedürfnissen der Endkunden entsprechen. Beispielsweise wünschen diese mehr Flexibilität im Bereich der Software-Paketerstellung. Man sollte sich darüber im Klaren sein, dass eine einzige Methode zur Monetarisierung nicht ausreicht. Daher ist der optimale Weg eine Kombination aus diversen Software-Monetarisierungsmethoden. Zusammenstellung, Kontrolle, Verwaltung und Verfolgung, diese vier Faktoren sind ausschlaggebend für die Strategie, die in Bezug auf die Monetarisierung realisiert werden soll. Hier ist es entscheidend, wie effektiv diese Elemente umgesetzt werden können. [14]

Diese vier Begriffe wie folgt definiert:

Im ersten Faktor „Zusammenstellung“ liegt der Fokus auf der Software-Paketerstellung. Hierfür sind Smartphone „Apps“ das beste Beispiel aus dem alltäglichen Leben. Viele Anbieter stellen meist eine kostenlose Testversion zur Verfügung. Leider wird der Kunde einer ständigen Publikation von Werbung ausgesetzt. Erst nach dem Erwerb der Vollversion erhält man zusätzliche Funktionen und ist die lästige Werbung los.

Dieses Prinzip spiegelt sich in vielen Unternehmen wider, die Software vertreiben. Hier muss allerdings nochmals un-

terschieden werden, ob es sich beim Endkunden um einen Privatkunden oder einen Geschäftskunden handelt.

Ebenso unbedingt notwendig für die Software-Monetarisierung ist die Kontrolle. Hier steht im Vordergrund, die nicht autorisierte Verwendung oder Distribution der Software zu unterbinden. Durch eine effiziente Kontrolle kristallisieren sich folgende Vorteile heraus: Compliance in Bezug auf Softwarelizenzen, Schutz vor Softwarepiraterie und Code-Diebstahl wird unterbunden. Zusätzlich bietet sie den Unternehmen auch die Möglichkeit, die Integrität ihrer Produkte und Marken zu sichern.

Im Faktor Verwaltung geht es darum, dass eine Integration in alle verfügbaren Systeme mühelos möglich sein sollte. Zusätzlich sollten die Systeme transparent, benutzerfreundlich und einfach sein. Nicht nur für unternehmensinterne Personen, sondern auch z.B. für externe Kunden sollte die Nutzung verständlich sein. Zentralisierung, Automatisierung und die Selbsthilfe für Endbenutzer sind die Basis für ein zielführendes Software-Monetarisierungsmanagement. Für die Automatisierung von verschiedenen Prozessen in Bezug auf Lizenzvergabe, Lizenzierung oder Softwarebereitstellung gibt es entsprechende Tools. Eine solche Lösung würde nachfolgende Vorteile mit sich bringen:

- Ausschließung menschlicher Fehler
- Automatisierung von manuellen und zeitaufwändigen Aufgaben
- Zurückverfolgung über den gesamten Zeitraum von Lizenzierungstrends und auf das Nutzungsverhalten der Endanwender

Im letzten Unterpunkt spielt der Faktor Verfolgung eine große Rolle. Er wird als Hauptaspekt einer umfangreichen Monetarisierungsstrategie genannt. Beschäftigt wird sich mit Berechtigungen und dem Nutzungsverhalten der Endanwender, dazu dient nicht nur die Nachverfolgung, sondern auch die Berichterstattung. Die Produktnutzung wird nachverfolgt, um dem Unternehmen Einblicke in die Einhaltung der Lizenzvereinbarungen zu verschaffen. Des Weiteren sind diese erforderlich, um zukünftige Paketerstellungsstrategien voranzutreiben. Die Vermarktung der Lösungen, entsprechend dem tatsächlichen Verhalten der Endanwender, stellt einen Vorteil dar. Doch nicht nur für das Unternehmen sind die Nutzungsdaten wertvoll, auch für den Endanwender sind diese von Interesse. Wenn diese zugänglich gemacht werden, erhält der Kunde z.B. Informationen, die ihm helfen können, die Einhaltung von Lizenzvereinbarungen sicherzustellen. [7]

Der Begriff Feature wird des Öfteren unterschiedlich für verschiedene Bereiche definiert. Da der Begriff im vorliegenden Artikel im Zusammenhang mit Software verwendet wird, sollte dieser zuvor verständlich abgegrenzt werden. Im Allgemeinen handelt es sich bei Features um spezielle Funktionen in einer Software. Darunter werden meist nur außergewöhnliche bzw. besondere oder neue Funktionen eingeordnet.

Eine weitere Begrifflichkeit, die zuvor definiert werden muss, ist das Modul. Unter dem Bezug zur Software beschreibt der Autor Bernd Kahlbrandt den Begriff Modul als eine physische oder logische Einheit, die eine klar definierte Aufgabe besitzt. Zusätzlich besteht sie aus mehreren Bestandteilen. [9, vgl. S. 57 f.]

Das Wort Lizenz stammt aus dem Lateinischen und wird abgeleitet von Licentia, welches soviel bedeutet wie Erlaubnis, Freiheit. [5] Im Bereich von Softwareanwendungen stellt eine Lizenz die Erlaubnis dar, eine Software zu nutzen. Erst nachdem eine Lizenz erworben wurde, kann die gewünschte Software genutzt werden. Ein Alltagsbeispiel ist der Kauf eines Microsoft Office Paketes, dies ermöglicht es, z.B. mit Word zu arbeiten. Zu betonen ist hier, dass mit einer Lizenz ein Nutzungsrecht erworben wird, jedoch nicht die Software, diese bleibt das Eigentum des Unternehmens. Man kann eine Lizenz dauerhaft und unbeschränkt kaufen. Im Gegensatz dazu ist es auch möglich, dass eine Erlaubnis beschränkt ist. Mögliche Schranken sind hier die Festlegung einer limitierten Zeitdauer oder Rechte, die nicht jeder Nutzer erhalten darf. Die Dauer der Nutzung zu begrenzen ist sinnvoll, wenn man z.B. möglichen Neukunden für vier Wochen die Anwendung zum Testen zur Verfügung stellen möchte.

Der Autor des Buches „Erfolgreiche Software-Lizenzierung“ Reimer Bürkner, definiert den Begriff Lizenzierung in seinem zweiten Kapitel wie folgt. Er nennt den Begriff Lizenzierung als Sammelbegriff für Maßnahmen. Diese müssen einerseits zum Softwareschutz vor illegaler Nutzung, andererseits um flexible Vermarktungsmodelle zu schaffen, ergriffen werden. Fälschlicherweise geht die Mehrheit davon aus, dass der Begriff Lizenzierung dasselbe bedeutet wie Kopierschutz. Der Kopierschutz wird als Bestandteil der Lizenzierungsmaßnahme gesehen, er wird meist benötigt, um das Ziel der Lizenzierung zu erreichen. Früher verwendete Methoden, bei denen die Lizenzierung rein auf einem Blatt Papier festgeschrieben wurde, sind längst nicht mehr zeitgemäß. Im heutigen Zeitalter gibt es dafür Softwaretools, die es ermöglichen, die Einhaltung der Vertragsbestimmungen zu kontrollieren. Versucht der Kunde, die Anwendung auf mehr Systemen, als ihm im Vertrag erlaubt ist, einzusetzen, so muss er mit Konsequenzen rechnen. In den härtesten Fällen kann dies dazu führen, dass er die Nutzungsberechtigung für die Software verliert. [2, vgl. S. 3]

Das große Angebot an diversen Lizenzierungsarten ermöglicht es jedem Kunden, individuell seine Wunschart frei zu wählen. Die Sammlung aller Arten basiert auf dem Werk „Erfolgreiche Software-Lizenzierung“, diese wurde zusätzlich durch aktuell verfügbare Arten ergänzt. Die nachfolgende Table I beinhaltet jedoch nur eine ausgewählte Anzahl an Lizenzierungsarten. [2, vgl. S. 36 ff.] [13, vgl. S. 5]

Table 1
AUFLISTUNG LIZENZIERUNGSARTEN.

<u>Lizenzart</u>	<u>Definition</u>
Einzelplatzlizenz	Die Lizenz kann entweder an eine virtuelle Maschine gebunden sein oder für einen bestimmten Rechner definiert werden. Eine weitere Möglichkeit ist, dass sich die Lizenz auf einem Dongle befindet, der lokal angeschlossen wird.
Netzwerklicenz	Diese ist meist besser bekannt als Floating-Lizenz. In diesem Fall befindet sich die Lizenz auf einem zentralen Server im Netzwerk. Die Anzahl der erworbenen Lizenzen gibt an, wie viele Lizenzen gleichzeitig im Netzwerk verwendet werden können. Hier ist zuvor auch eine Definition einer Rechnerliste möglich. Diese enthält alle zugelassenen Rechner. Die Lizenzen könnten ebenfalls auch an eine Nutzerliste gebunden sein.
Unbefristete Lizenz	Der Kunde erwirbt eine dauerhafte Lizenz, deren Nutzungszeitraum nicht abläuft.
Demo- und Testlizenz	Im englischen Sprachgebrauch steht die Bezeichnung TBYB für diese Lizenzart. Dem Anwender kann die Software im vollen oder mit eingeschränktem Funktionsumfang zur Verfügung gestellt werden. Meist ist die Lizenz zeitlich begrenzt.
Feature-on-Demand-Lizenz	Eine Software kann durch eine Vielzahl von zusätzlichen Features erweitert werden. Dies unterbreitet dem Kunden eine flexible Lizenzierung hinsichtlich seiner Featurewahl. Für das Software-Unternehmen ist dies von großer Bedeutung, da auf diese Weise ein zusätzlicher Umsatz generiert werden kann.
Verbrauchsabhängige Lizenz	Die Kosten der Lizenzierung sind abhängig von der tatsächlichen Nutzung des Kunden. Die Nutzung kann entweder zeitbasiert oder aktionsbasiert festgehalten werden. Beispielsweise könnte es sein, dass der Kunde zehn Euro pro Stunde bezahlen muss.
Benutzerabhängige Lizenz	Bei dieser Möglichkeit wird die maximale Anzahl gleichzeitiger Benutzer festgehalten. Diese Aufzeichnung stellt die Grundlage zur Abrechnung dar.

B. Analyse von technischen Lösungen

Nachdem im vorherigen Kapitel die theoretischen Begrifflichkeiten eindeutig erläutert wurden, werden im nachfolgenden Abschnitt diverse technische Lösungen für den Bereich Software-Monetarisierung analysiert. Zunächst wurde hier zwischen Hardware- und Software-Lösungen unterschieden. Im Nachgang wurden konkrete Beispiele für eine Umsetzung dargestellt, hier werden verschiedene Anbieter genannt. Wichtige Aspekte sind, die Software vor Weiterverbreitungen zu schützen, sie nach bestmöglichen Umsatzkriterien zu lizenzieren und eine möglichst einfache Integration in das bestehende System zu schaffen.

1) *Hardware-Lösung:* Sollte sich für eine hardwareorientierte Lösung entschieden werden, gibt es eine Möglichkeit, die im Folgenden beschrieben wird. In diesem Bereich könnte sich für eine portable Lizenz entschieden werden. Größtenteils ist dies so, dass beim Kauf einer Software ein zusätzliches Medium im Paket vorhanden ist. Hinsichtlich dessen gibt es verschiedene Hardware-Ausführungen, es könnte sich daher beispielsweise um einen Universal Serial Bus (USB)-Stick (Dongle) oder eine SD-Karte handeln. Um die Software funktionell nutzen zu können, muss das Medium an einer geeigneten Schnittstelle des Rechners eingesteckt werden. Somit kann die Software auf jedem PC, an dem der Dongle sich befindet, genutzt werden. Darüber hinaus besteht die Möglichkeit, nicht für jeden einzelnen PC einen Dongle zu verwenden, sondern für mehrere PCs nur einen Dongle im Netzwerk in Betrieb zu nehmen. Auf diesen dann alle Geräte, die sich im Netzwerk befinden, zugreifen. Falls das Medium nicht vorhanden ist, kann nicht mit der zu lizenzierenden Software gearbeitet werden, da die Anwesenheit beim Aufruf der Software überprüft wird. Mehrere Anbieter bieten ihren Kunden trotzdem eine eingeschränkte Funktionalität der Software an, selbst wenn der Dongle nicht vorhanden ist. Dies setzt allerdings voraus, dass eine einmalige Verifizierung des Dongles gemacht wurde. Für die Lösung gibt es unterschiedliche Anbieter, mit jeweils spezifischen Vor- und Nachteilen. In den späteren Abschnitten wird sich genauer mit dem USB-Dongle und seinen Merkmalen befasst.[10]

2) *Software-Lösungen:* Neben der Hardware-Lösung wurden auch verschiedene Software-Lösungen untersucht. Dieser Teil gliedert sich im Wesentlichen in drei unterscheidbare Lösungen. Diese wären:

- Lizenzfile
- Lizenzserver
- Cloud

Lizenzfile: Wie man in der Bezeichnung bereits erkennt, handelt es sich beim Lizenzfile um eine generierte Datei, die verschlüsselte Informationen enthält. Der Unterschied zum Dongle besteht darin, dass diese Datei an keine externe Hardware gebunden ist. Überwiegend ist die Datei direkt an die Hardware eines PCs beim Endkunden gebunden.

In einer virtuellen Software-Umgebung wird es meist direkt der Anwendung zugeordnet. Es können für die Bindung verschiedene Eigenschaften gewählt werden z.B. IP-Adresse, Merkmale der Festplatte oder der Netzwerkkarte. Üblicherweise erhält der Endkunde einen Aktivierungscode, um die Software nutzen zu können. Sobald die Aktivierung durchgeführt worden ist, erhält er einen weiteren Schlüssel. Mit diesem Lizenzschlüssel kann die Software dann nur auf dem PC genutzt werden, auf dem sie auch aktiviert wurde. Daraus ergibt sich ein wesentlicher Vorteil gegenüber der Hardware-Lösung. Die Handhabung mit Lizenzfiles ist einfacher zu gestalten, da ein solches auf elektronischem Weg

verschickt werden kann. Im Gegensatz dazu muss der Versand bzw. der Austausch von Dongles auf dem Postweg erfolgen. [2, vgl. S. 35] [10]

Lizenzserver: Mit einem Lizenzserver werden die zur Verfügung stehenden Lizenzen verwaltet. Dies bedeutet, dass eine einmalige Installation im Netzwerk des Kunden umgesetzt werden muss. In der Praxis wird dies sehr häufig in Verbindung mit Floating-Lizenzen eingesetzt, da hier die maximale Anzahl gleichzeitig eingeloggter User auf unterschiedlichen PCs lizenziert wird. Es ist möglich, jeden beliebigen Rechner im Netzwerk zu verwenden, da der Lizenzserver im Prinzip nur einen Server-Prozess darstellt. Für die Kommunikation zwischen der lizenzierten Software und dem Lizenzserver werden meistens eine oder mehrere APIs vom Hersteller der Lizenzierungstools zur Verfügung gestellt. Diese Schnittstelle muss zuvor in die bestehende Software integriert werden. [2, vgl. S. 35]

Cloud: Im Bereich Cloud Computing gewinnt der Service-Vertrag immer mehr an Bedeutung. Der Wartungsvertrag sowie die Software-Lizenz wird durch ihn ersetzt. Im Umfeld der Cloud wird zwischen drei unterschiedlichen Services unterschieden. Diese werden wie folgt bezeichnet: Software as a Service (SaaS), Infrastructure as a Service (IaaS) und Platform as a Service (PaaS). Um sich einen Überblick über die Unterschiede der Begrifflichkeiten zu verschaffen, werden diese kurz definiert.

Der Autor Troy J. Strader definiert SaaS wie folgt. Beim Bereitstellungsmodell SaaS wird die Software in einem Rechenzentrum betrieben. Dies kann entweder intern oder über einen externen Anbieter geschehen. Dem Kunden wird die Software über das Internet zur Verfügung gestellt. Darüber hinaus unterstützt SaaS Multi Tenancy, dies bedeutet, dass jedem Kunden die gleiche Infrastruktur bereitgestellt wird. [12, vgl. S. 156]

Unter dem Begriff IaaS versteht man die Zurverfügungstellung der Hardware z.B. eines Servers. Für die Bereitstellung und Kontrolle der Anwendungen bzw. des Betriebssystems ist das Unternehmen, das die Software zur Verfügung stellt, verantwortlich. Im Vergleich zum traditionellen Hosting erfordert es keine langfristige Verpflichtung. [1, vgl. S. 62]

Der letzte Service PaaS kann wie folgt erläutert werden. Dienste die zwischen SaaS und IaaS existieren werden als PaaS bezeichnet. Diese Dienste bieten Plattformen an, auf denen Entwickler Programme ausführen können. Diverse Faktoren legen fest, wieviel Freiraum der Entwickler auf der Plattform hat. [8, vgl. S. 2]

Nach der Begriffsklärung werden kurz Vor- und Nachteile beschrieben. Die Cloud ist in Bezug auf Raubkopien bzw. Weitervertrieb der Software das genaue Gegenteil des Lizenzfiles. Der Grund dafür ist, dass der Server in den meisten Fällen vom Anbieter betrieben wird und dieser so den Nutzungsumfang festlegt und kontrolliert. Ein weiterer Vorteil ist die Flexibilität der Services. Dadurch können Anpassungen an den tatsächlichen Bedarf des Anwenders realisiert wer-

den. Als Abrechnungsgrundlage wird meist die tatsächliche Nutzung verwendet. Sollte diese Umsetzung gewählt werden, stehen den Vorteilen auch negative Aspekte gegenüber. Des Weiteren muss sich mit dem Thema Datenschutz in der Cloud gründlich auseinandergesetzt werden. [4, vgl. S. 52 ff.]

3) *Lösungsmöglichkeiten der Anbieter:* Im Anschluss wurden Anbieter betrachtet, die sich auf Software-Monetarisierung bzw. Lizenzierung spezialisiert haben. Es wurden vier externe Lieferanten zur Auswertung herangezogen. TechnoData Interware, WIBU-SYSTEMS, Gemalto und Flexera Software bieten eine breite Auswahl an Produkten, die für die Umsetzung denkbar wären.

a) *Entscheidungsfindung:* Nachstehend wird zusammenfassend eine Gegenüberstellung der Lösungen präsentiert. Anhand derer wurde sich für eine Lösung entschieden, die dann im Anschluss prototypisch umgesetzt werden soll. Zuvor wurden die Vorbedingungen für die Implementierung definiert. Da sich für den Dongle oder das Lizenzfile entschieden wird, wurden diese nochmals genauer beleuchtet.

b) *Vor- und Nachteile:* In folgender Table II werden die Vor- und Nachteile der zwei Lösungsmöglichkeiten Dongle und Lizenzfile aufgezeigt.

Table II
VOR- UND NACHTEILE DER LÖSUNGSMÖGLICHKEITEN.

Dongle		Lizenzfile	
Vorteile	Nachteile	Vorteile	Nachteile
Anwendung ohne Dongle funktionslos	Defekt/Verlust des Dongles erfordert Ersatzlieferung (Lieferzeit)	Handling einfacher	Veränderung von PC-Merkmalen veranlasst Neukauf einer Lizenz
Hardwaredaten müssen nicht ausgelesen werden	Höhere Kosten durch Verwaltung, Lagerkosten	Weit verbreitet	Erstellung von Lizenzen wird intern jedem ermöglicht
Hohe Sicherheit	Funktion muss beim Migrieren einer VM erhalten bleiben	Kostengünstiger: File 39 €, Dongle 78 €	Hardwaredaten müssen zur Verfügung stehen
Keine neue Lizenzgenerierung	Kosten für physische SST liegen bei ca. 100 €	Gute Sicherheit	Duplikation in VM möglich
Anwendung sofort nutzbar	Kundenspezifische Erstellung sichern		Anwendung ohne File funktionslos

Aufgrund der Abwägungen zwischen Pro und Contra hat sich die Firma SYSKRON dafür entschieden, die Lizenzierung mit der Dongle-Lösung umzusetzen. Im nächsten Schritt wurde sich für einen Dongle-Anbieter entschieden werden. Um die Entscheidungssituation anschaulicher darzustellen, wurde zusätzlich eine Nutzwertanalyse angefertigt. Hierfür wurden alle Muss- und Sollkriterien aufgeführt, sowie die drei Anbieter.

II. PRAKTISCHER TEIL

A. Technisches Konzept

Im ersten praktischen Teil wurde sich mit der Erarbeitung eines Konzeptes, für die Integration der zu implementierenden Lizenzierungskomponente in das bestehende System, beschäftigt. Dafür wurden Abbildungen bzw. Prozessabläufe konzipiert, diese sollen zu einem besseren Verständnis beitragen. Es wurde die Systemarchitektur, Beziehung zwischen Benutzerverwaltung und Lizenzierung, Integration- und Schnittstellendarstellung sowie der Prozessablauf für die Lizenzprüfung von Floating-Lizenzen und Features konzipiert. Sie beschreiben wie und an welchen Stellen die Integration eines Dongles in das bestehende System erfolgen kann.

B. Prototypische Implementierung

Im wichtigsten Teil wurde die prototypische Implementierung der Dongle Applikation entwickelt. Da die SYSKRON im Entwicklungsumfeld mit Java programmiert, war es naheliegend, die Anwendung für Software-Monetarisierung ebenfalls in dieser Programmiersprache zu implementieren. Darüber hinaus wurde JavaFX als Framework eingesetzt, dieses ermöglicht grafische Java-Oberflächen zu entwickeln. [11, vgl. S. 7 f.] Das Aussehen der Applikation wurde mit CSS an den bestehenden Client angepasst. Die Abkürzung steht für Cascading Style Sheets (CSS), sie wird zur Gestaltung der GUI benutzt. Für die eindeutige Zuordnung der Features in die zu beschreibenden Felder des Dongles wurde eine Properties-Datei verwendet. Diese ermöglicht eine einfache Handhabung, falls die Featureliste erweitert wird. Ein einmaliger Eintrag in die Datei und das Feature wird ohne Änderungen im Java Code hinzugefügt. Allgemein handelt es sich bei einer Properties-Datei um eine Textdatei. Diese enthält eine Liste von Schlüsselwörtern, in unserem Fall Version, Lizenzanzahl und verschiedene Features. Diesen Wörtern wird ein eindeutiger Wert zugewiesen, damit wird das Feld festgelegt, das im Dongle mit der Lizenzanzahl beschrieben wird. [3, vgl. S. 197] Benötigt wird außerdem die vom Dongle-Anbieter zur Verfügung gestellte API. Mit dieser können die grundlegenden Funktionen in die Anwendung eingebaut werden. Zur Vervollständigung sei erwähnt, dass für die Umsetzung der Applikation die Entwicklungsumgebung Eclipse gewählt wurde.

Um einen ersten visuellen Entwurf der Anwendung zu erhalten, wurde diese im Vorfeld der Programmierung mit einem Mockup-Tool grob skizziert. Danach wurde sich intensiv mit dem gewählten Dongle beschäftigt. Im letzten Schritt wurde der Prototyp implementiert.

III. FAZIT

Im einleitenden Teil des Artikels wurden explizite Gründe für die Einführung einer geeigneten Software-Monetarisierung

Lösung genannt. Da die Lösung bereits in kommenden Projekten eingesetzt werden soll, wurde dem Thema eine hohe Wichtigkeit erteilt.

Das Ziel der verfassten Bachelorarbeit bestand darin, eine prototypische Implementierung für den Bereich Software-Monetarisierung zu erstellen. Um die vorgegebene Zielsetzung zu erreichen, wurde ein Überblick über die verschiedenen Lösungsmöglichkeiten erarbeitet. Dazu erfolgte zuerst die Betrachtung der Hardware- sowie Software-Lösungen und deren Anbieter. Die Ergebnisse der Analyse dienten im weiteren Verlauf als Grundlage für die Entscheidung. Anschließend wurden Vor- und Nachteile erläutert sowie eine Nutzwertanalyse verfasst. In diesem Teil wurde sich dafür entschieden, das Thema Lizenzierung mit der Dongle-Lösung in Angriff zu nehmen. Der vorgelegte Konzeptentwurf bildete die Basis zur Integration in das bestehende System. Durch die Erstellung des Konzeptes wurde deutlich, dass in der Umsetzung viele Aspekte berücksichtigt werden müssen. Durch die prototypische Implementierung wurde die Schnittstelle zum Lizenz-Anbieter geschaffen. Damit wird ermöglicht, dass der Dongle sowohl ausgelesen als auch beschrieben werden kann. Im weiteren Verlauf muss die SYSKRON entscheiden, ob die bisher prototypische Implementierung der Dongle-Applikation bereits intern für anstehende Projekte verwendet werden soll.

Zusammenfassend lässt sich feststellen, dass die SYSKRON mit dem erarbeiteten Thema einen großen und vor allem wichtigen Schritt in Richtung Software-Monetarisierung gemacht hat. Im Ausblick auf die Zukunft bedarf es einer stetigen Weiterentwicklung der Software-Monetarisierungsstrategie.

REFERENCES

- [1] Bhardwaj, Sushil and Jain, Leena and Jain, Sandeep. 2010. *Cloud computing: A study of infrastructure as a service (IAAS)*. International Journal of engineering and information Technology.
- [2] Bürkner, Reimer M. 2003. *Erfolgreiche Software-Lizenzierung: Electronic License Management - Von der Auswahl bis zur Installation*. Berlin and Heidelberg: Springer.
- [3] Czernohous, Christoph. 2011. *Pervasive Linux: Basistechnologien, Softwareentwicklung, Werkzeuge*. Berlin and Heidelberg: Springer.
- [4] Dr. Undine von Diemar. 2010. *Cloud Computing - Abschied von der Software-Lizenz*. IP Manager - Journal for the Knowledge Economy.
- [5] Dudenredaktion, Wissenschaftlicher Rat. 2013. *Duden - Die deutsche Rechtschreibung: Auf der Grundlage der aktuellen amtlichen Rechtschreibregeln ; [das umfassende Standardwerk auf der Grundlage der amtlichen Regeln]*. Dudenverlag, Bibliographisches Institut F.A. Brockhaus.
- [6] Evercoder Software S.R.L.. 2017. <https://app.moqups.com/edit/page/ad64222d5>.
- [7] Gemalto. 2014. *Was ist Software-Monetarisierung? :: SafeNet, Inc.*. <http://www.safenet-inc.de/software-monetization/what-is-software-monetization/>.
- [8] Haselmann, Till and Vossen, Gottfried. 2010. *Abgrenzung des Database-as-a-Service-Begriffs aus Benutzerperspektive*.
- [9] Kahlbrandt, Bernd. 2013. *Software-Engineering mit der Unified Modeling Language*. Berlin and Heidelberg: Springer.

- [10] mirabyte GmbH & Co. KG. 2016. *mirabyte Wissensdatenbank: Was ist der Unterschied zwischen der hardware-gebundenen Lizenz und der portablen Lizenz (mit USB-Dongle)?*. <http://www.mirabyte.com/de/support/wissensdatenbank.html?kb=5>.
- [11] Steyer, Ralph. 2014. *Einfuehrung in JavaFX: Moderne GUIs fuer RIAs und Java-Applikationen*. Berlin and Heidelberg: Springer.
- [12] Strader, Troy J. 2010. *Digital Product Management, Technology and Practice: Interdisciplinary Perspectives: Interdisciplinary Perspectives*. IGI Global.
- [13] WIBU-SYSTEMS. 2015. *Schutz, Lizenzierung und Sicherheit von Software: CodeMeter: CodeMeter-SoftwareSchutzBroschuere*.
- [14] WIBU-SYSTEMS. 2016. *Maximieren Sie Ihren Umsatz - nutzen Sie CodeMeter zur Software-Monetarisierung: Wibu Systems*. <http://www.wibu.com/de/software-monetarisierung.html>.