

<b>Intro</b>	<b>Abstract</b>
	<b>Purpose</b>
	<b>Past</b>
<b>Concepts</b>	<b>Attitudes</b>
	<b>Strategies</b>
	<b>Modalities</b>
<b>Partners</b>	<b>Students</b>
	<b>Teachers</b>
	<b>Institutions</b>
<b>Spaces</b>	<b>Buildings</b>
	<b>Exhibitions</b>
	<b>Interspaces</b>
<b>Output</b>	<b>Workshops</b>
	<b>Projects</b>
	<b>Events</b>
<b>Extras</b>	<b>Business</b>
	<b>Statements</b>
	<b>Glossary</b>



**magnolia**

For digital Media Management  
powered by QuickTime

## 1. IDEE

### 1.a Ausgangslage

Multimediale Inhalte auf dem World Wide Web besitzen heute einen sehr hohen Stellenwert, Contentprovider haben die Macht von bewegtem Bild und Ton längst erkannt und es wird zunehmend auf diese Mittel zurückgegriffen um Inhalte und Botschaften interessanter und/oder klarer zu vermitteln.

Auch viele Ereignisse und Veranstaltungen sind heute medial erst über das Internet erfahrbar.

Consumer und Prosumer bedienen sich eifrig digitaler Kameras und Handys um so auf einfache Art und Weise (bewegte) Bilder aufzuzeichnen. Dieses Material drängt sich für eine Verbreitung über das Internet geradezu auf.

Breitbandanschlüsse verzeichnen Zuwachsraten von bis zu 30% was auch für weiterhin steigenden Bedarf an online Medien garantiert.

Aus einer Studie<sup>1</sup> von PwC:

The major impact of broadband during the next three years will be the addition of video to existing applications such as conferencing, messaging and gaming and the development of applications that rely on user- and community-provided video content. As applications such as Weblogs make clear, the two-way interactive nature of the Internet allows consumers to create new content, and broadband will extend this capability beyond text to video.

<b>1.b Aktuelle Technik</b>	<p>Unverständlich erscheint, dass sich bislang opensource<sup>2</sup> Content Management Systeme<sup>3</sup> (CMS) dieser Thematik nicht ernsthaft angenommen haben. Bei SourceForge<sup>4</sup> ist bis zum heutigen Datum keine Webapplikation erhältlich, die sich dediziert mit der Verarbeitung von Film, Bild und Ton auf dem Netz befasst. Dies basiert wohl auf der Komplexität, welche die Materie mit sich bringt, speziell für plattformübergreifende Lösung.</p>
<b>1.c Bedürfnisabklärung</b>	<p>In diversen Gesprächen mit Film- und Medienschaffenden, CMS-Providern, Softwareentwicklern, Künstlern, Studenten, Fotografen, Schauspielern und Privatpersonen habe ich die Sinnhaftigkeit einer solchen Lösung beleuchtet und die Anforderungen welches so ein System zu erfüllen hat erörtert. Die allgemeine Interessensbekundung der befragten Personen brachte mich zu dem Entschluss Planung, Konzeption und Realisation eines Publikationsmechanismus für digitale Medien zum Ziel meiner Diplomarbeit zu ernennen.</p>
<b>1.d Ziel</b>	<p>Während meiner Diplomarbeit erstelle ich eine Erweiterung zu einem CMS, dass das Verwalten, Editieren und Publizieren von digitalem Bild, Film und Ton auf das Internet einer breiten Zielgruppe einfach ermöglicht.</p>
<b>1.e Anforderungen an das System</b>	<ul style="list-style-type: none"> <li>- Die Handhabung ist intuitiv und einfach.</li> <li>- Der Zyklus von Erfassen über Bearbeiten bis zur Webpublikation wird erleichtert und verkürzt.</li> <li>- Sämtliche Operationen sind Browserbasiert.</li> <li>- Im Interface wird immer eine Voransicht der selektierten Datei angezeigt.</li> <li>- Das Produkt baut auf bewährten Softwarepaketen auf.</li> <li>- Möglichkeit zur Kompression von Daten für eine webgerechte Präsentation.</li> <li>- Ablage der Originaldaten soll auf dem Server erfolgen um somit auch Teamarbeit zu ermöglichen.</li> <li>- Das Ergebnis liegt im „quelloffenen Format“ vor.</li> <li>- Eine interessierte Entwicklergemeinde kann sich aktiv an dem Projekt beteiligen.</li> </ul>

## 2. PLANUNG

### 2.a Grundlagen

Magnolia als junges Produkt der obinary ag, Basel schien mir eine sehr interessante Option. Die Intensität mit der die Softwareschmiede ihr CMS vorantreibt fasziniert. Magnolia ist Java<sup>5</sup> basiert unterstützt JCR 170<sup>6</sup> und kann mit einem attraktiven Interface punkten, dies waren mitunter Gründe die für die Wahl sprachen. Auch bekundete obinary grosses Interesse an meinem Vorhaben und sagte mir strategische und technische Unterstützung zu.

QuickTime<sup>7</sup> for Java (QT4J) ist eine Java Bibliothek die durch eine Kommunikationsschicht Zugriff auf die von Apple entwickelte QuickTime Software zulässt. Somit eröffnet QT4J praktisch unbeschränkte Möglichkeiten zur Bearbeitung von multimedialen Dateien. Diese Bibliothek wählte ich aufgrund ihres Leistungsumfanges, der Markterprobtheit und freien Zugänglichkeit.

### 2.b Team

Um der Komplexität von QT4J in Kombination mit J2EE<sup>8</sup> Herr zu werden und bei allfälligen Problemfällen Rückendeckung zu erhalten, beschloss ich eine externe Person in das Vorhaben einzubinden. Hier konnte ich Anthony Rogers, head of science bei tomato in London und ehemaliger Entwickler von QT4J bei Apple, gewinnen.

Auch innerhalb der Mauern von HyperWerk stiess mein Vorhaben auf Interesse und so wurde ich von der „crème de la crème“ der Programmierfront aus den studentischen Reihen unterstützt.

Somit ergab sich folgendes Team:

Intern:

Thomas Bach, cinque (Konzept, Applikationsdesign, Programmierung Backend)

Roman Bohrer, cinque (Produktion Medienbeitrag Nomadix)

Mariana Erzinger, cinque (Produktion Medienbeitrag Nomadix)

Andreas Hupfer, cinque (Konzept, Applikationsdesign, Programmierung QT Klassen)  
Dominik Imark, sei (Konzept, Design & Programmierung Interface)  
Miriam Risolé, (ehem.) sei (Konzept & Design, Planung Workshops)  
Mischa Schaub, Leitung HyperWerk (Interner Diplomcoach)

**Extern:**

Philipp Bracher, Software Engineer obinary ag (Unterstützung Integration Interface)  
Sameer Charles, Software Engineer obinary ag (Unterstützung Integration Backend)  
Jan Knopp, Freier Texter (Text)  
Pascal Mangold, CEO obinary ag (Mentor Interface & Integration)  
Anthony Rogers, head of science tomato (Technischer Mentor Backend)  
Markus Schnell, Mediatektur (Externer Diplomcoach)  
Sven Spreng, kommunikation mit rotem faden (Design)

## 2.c Pflichtenheft



Der Leistungsumfang der Bearbeitungsfunktionalitäten für magnoliaQT 1.0 wurde in etlichen Gesprächen mit beteiligten und externen Personen wie folgt festgelegt:

**Bild:** Skalieren  
Rotieren  
Spiegeln  
Ausschnitt definieren  
Komprimieren (JPEG)

**Film:** Skalieren  
Schnitt (In- und Outpoint setzen)  
Soundtracks entfernen/hinzufügen  
Komprimieren (verschiedene Formate/Qualitäten)

**Ton:** Schnitt (In- und Outpoint setzen)  
Komprimieren (verschiedene Formate/Qualitäten)

### 3. UMSETZUNG

#### 3.a Technische Grundlagen

Bei einem ersten Besuch im Februar in London konnte ich Details über Handling und Vorgehensweisen im Umgang mit QT4J und magnolia mit Anthony Rogers besprechen. Ebenfalls erörterten wir Formen der Zusammenarbeit. Anthony Rogers zeigte mir einige Anwendungsbeispiele, wies auf Stolpersteine hin und gab Lösungsansätze. Wir diskutierten über Nutzungsbedingungen, Usability und Zukunftsmöglichkeiten von magnoliaQT. Gemeinsam kamen wir zum Schluss, dass die Prozessabläufe und Ergonomie des Interfaces Schlüssel zur Akzeptanz des resultierenden Produktes sein werden und eine enge Zusammenarbeit mit obinary sinnvoll ist, um eine möglichst nahtlose Integration in magnolia zu gewährleisten.

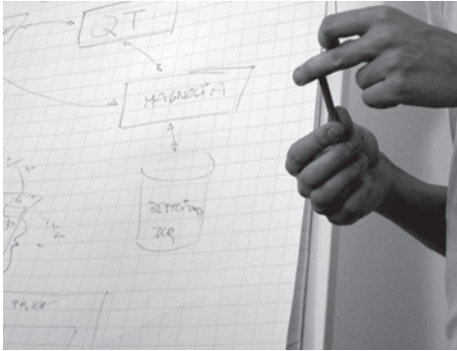
#### 3.b Einarbeitung des Teams

Um alle innerhalb des Teams auf den aktuellen Stand der Technik zu bringen, wurde zuerst eine Einarbeitung in die Java und J2EE Technologien durchgeführt. Als IDE<sup>9</sup> kam Eclipse<sup>10</sup> zum Einsatz und zur Versionenverwaltung des Quelltextes erhielten wir Zugang zum svn<sup>11</sup> Server von obinary, wo ein repository für das Projekt magnoliaQT eingerichtet wurde.

#### 3.c Hürden als Chancen

Im bereits fortgeschrittenen Stadium der Entwicklung stiessen wir auf zwei grundlegende Probleme, die den anfänglichen Zeitplan gehörig durcheinander warfen:

Das Abarbeiten von Prozessen (threading) in QT4J blockierte aus unerfindlichen Gründen immer wieder die Java Virtual Machine<sup>12</sup> (JVM). Da der Webserver (Tomcat) in derselben JVM ausgeführt wird, kam es bei einem solchen Absturz zum totalen Stillstand von magnolia. So blieb uns keine andere Wahl, als die komplette Medienbearbeitung ausserhalb von magnolia abzuhandeln, um das CMS somit wieder stabil und lauffähig zu haben.



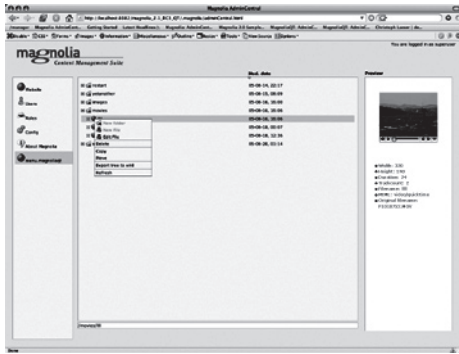
Hierzu entwickelten wir eigens einen QTFileProcessorServer, der für das Abarbeiten von Manipulationen an QuickTime Dokumenten zuständig ist. Die Brücke von magnolia zu diesem Dienst wurde über Java Remote Method Invocation<sup>13</sup> (RMI) realisiert.

Dies zog jedoch eine weitere Problematik nach sich: ein konkurrenzierendes Schreiben auf den magnolia unterliegenden Datenbehälter (JCR repository) ist wohl möglich, aber der Caching Mechanismus<sup>14</sup> kann von einer externen Applikation nicht aktualisiert werden. Somit gelangten die geänderten Files in das repository, aber magnolia konnte von deren Existenz nicht benachrichtigt werden und hatte entsprechend keinen Zugriff darauf. In langen Testserien, zusammen mit den Entwicklern von obinary ag, kamen wir zum Schluss, dass das Schreiben der Daten wieder über die magnolia Instanz geschehen muss.

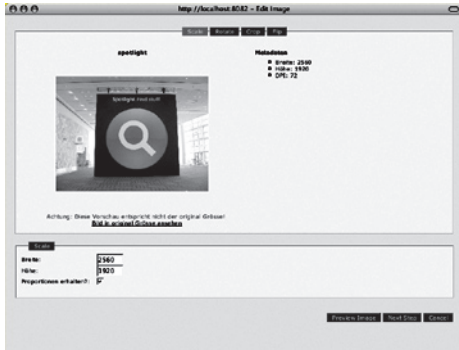
Das Resultat dieser unerwarteten Entwicklung war, dass die Architektur gründlich überarbeitet werden musste, dies eröffnete jedoch auch eine grossartige Chance: die vorhandenen Funktionalitäten sind nun soweit abstrahiert, dass lediglich das Interface von magnoliaQT noch an magnolia gekoppelt ist. Hiermit besteht die Möglichkeit, dass Produkt in ein beliebiges CMS einzubauen - eine Option, die in der ursprünglich geplanten Version nicht realisierbar gewesen wäre.

### 3.d Internationalisierung

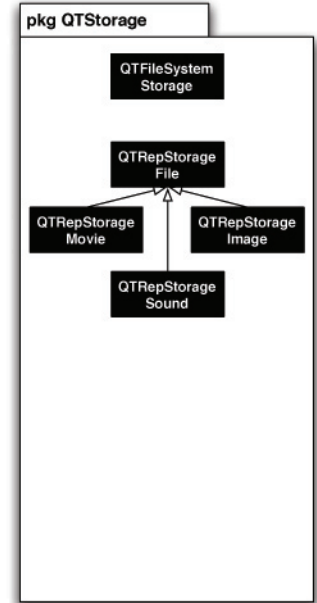
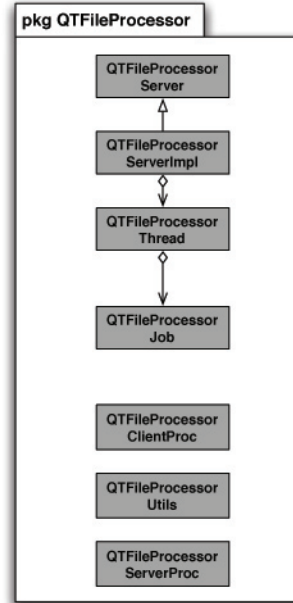
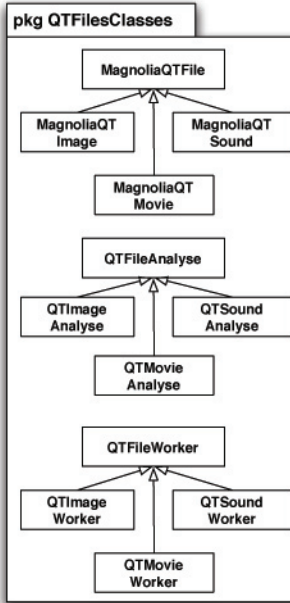
Ein letztes Feature, welches noch zusätzlich in die Entwicklung der Version 1.0 integriert wurde ist Mehrsprachigkeit des Interfaces (I18n<sup>15</sup>). Eine Option die gerade im Zuge einer Softwarelösung, welche ein internationales Publikum ansprechen soll, sehr sinnvoll ist. So wird magnoliaQT bei Veröffentlichung bereits in Deutsch und Englisch zur Verfügung stehen, wobei weitere Sprachen ohne ein erneutes Kompilieren<sup>16</sup> hinzugefügt werden können.



Screenshot der Dokumentenübersicht mit Vorschau des selektierten Objektes.



Screenshot der Editieransicht.



- Basisklassen zur Bearbeitung der Medienfiles 2581 Zeilen Code
- Zusätzliche Klassen zur externen Abarbeitung der Prozesse 872 Zeilen Code
- Zusätzliche Klassen für das Handling des Speichervorgangs der extern verarbeiteten Files 785 Zeilen Code

Stark vereinfachte Darstellung der Klassen im backend Bereich von magnoliaQT.



### 3.e Rechtliche Fragen

Um im Dickicht der opensource Lizenzen einigermaßen den Durchblick zu erhalten, besuchte ich bereits zu Anfang der Diplomarbeit ein Seminar bei Openlaw<sup>17</sup>. Dies hinterliess eine vage Vorstellung über den Einsatz dieser Lizenzen. Da die rechtlichen Gegebenheiten in magnoliaQT jedoch sehr speziell<sup>18</sup> sind, beschloss ich eine professionelle Juristin mit dieser Frage zu konfrontieren. Michelle Wahl kämpfte sich durch den Paragraphendschungel und kam zum Schluss, dass magnoliaQT unter der LGPL<sup>19</sup> vertrieben werden kann.

### 3.f Veröffentlichung von magnoliaQT

Im Laufe des Septembers wird die erste Version von magnoliaQT der Öffentlichkeit zum Download angeboten. Hierzu wird der Quellcode mit Anleitung auf SourceForge hinterlegt und von der Website <http://www.magnoliaqt.info> verlinkt. In Zusammenarbeit mit obinary ag ist ein Pressebericht geplant, der in diversen Onlinezeitschriften zur Veröffentlichung kommt.

## 4. AUSTAUSCH

### 4.a Allgemein

Die Entwicklung des Produkts machte einen grossen Teil meiner Diplomarbeit aus. Doch einen ebenso hohen Stellenwert mass ich dem aktiven Austausch mit einer potentiellen Zielgruppe, Know-How Trägern und zukünftigen Partnern zu.

Dieser Austausch fand an den folgenden drei Anlässen statt, bei welchen ich jeweils mein Vorhaben einer interessierten Gruppe näher bringen konnte:

#### 4.b Bern

Bereits ende Februar gab es einen ersten öffentlichen Austausch: anlässlich dem LOTS<sup>20</sup>, wurde in diversen Gesprächen mit Teilnehmern Vor- und Nachteile der angestrebten Lösung diskutiert. Hierbei wurde grosses Interesse bekundet, was mich in meinem Vorhaben bestärkte.

#### 4.c San Francisco

Die grösste Herausforderung und gleichzeitig Höhepunkt meiner Diplomarbeit bildete zweifelsfrei die Reise nach San Francisco, wo ich die Gelegenheit wahrnehmen konnte die WWDC<sup>21</sup> zu besuchen. Dies bedeutete für mich eine optimale Gelegenheit mich mit einer möglichst grossen Anzahl von Spezialisten mit entsprechendem technischem Background austauschen zu können. So war es mir auch möglich etliche Kontakte zu knüpfen, Feedback zu meinem Vorhaben zu sammeln, sowie Tipps und Hinweise zu erhalten wie das Produkt weiter verfeinert oder erweitert werden kann.

Interne Entwickler von Apple brachten mir in zahllosen Workshops diverse neue Technologien, User Interface Guidelines und Erneuerungen des Betriebssystems näher. Hierbei konnte ich mir viel Wissen aneignen und auch deutlich mehr Verständnis für das Innenleben von QuickTime und Kompressionsalgorithmen im allgemeinen gewinnen.



Die Krönung des Besuchs bildete das Gespräch mit Anant Sonone, dem Leiter des QT4J-Teams. Ich konnte ihm die Architektur und eine Alpha Version von magnoliaQT präsentieren. Sein Interesse war enorm: wir unterhielten uns längere Zeit, analysierten den Code und die grundlegenden Überlegungen. Er freute sich über das Vorhaben und die kreative Art und Weise wie magnoliaQT mit den QT4J Bibliotheken interagiert. Auch konnte er mir weitere Tipps zur Performancesteigerung geben. Offensichtlich hatte bislang niemand die Idee verfolgt, diese Tools in ein Content Management System zu integrieren. Nach seinen Angaben werde die Bibliothek im Allgemeinen hauptsächlich für Clientapplikationen und sehr selten auf dem Server eingesetzt. Wir unterhielten uns auch über die threading Problematiken und Anant Sonone eröffnete mir, dass QuickTime 7 unzählige Bugfixes zu dieser Problematik mit sich bringt.



Ich versprach ihm über den Stand des Projektes auf dem Laufenden zu halten und erhielt das Angebot bei gravierenden Hürden mit QT4J auf ihn zukommen zu dürfen.

Ein weiteres Highlight war das Gespräch mit Eric Zelenka, dem Head of OS X Server und Michael Robertson von obinary ag. Das Thema handelte sich um das fehlende CMS in Mac OS X Server. Mit dem Serverpaket werden nicht nur Fileservices sondern auch Webservices angeboten. Dutzende opensource Produkte sind Teil der Softwarelösung, dies beinhalten auch Blog- Chat- und Mailserver. Ein Content Management System jedoch ist bislang nicht mit an Board. Man sei auf der Suche und habe die Augen offen, meinte Eric Zelenka. Magnolia sei kein unbeschriebenes Blatt und das Vorhaben von magnoliaQT würde den Wert dieses CMS aus der Sicht von Apple steigern, liess Eric Zelenka verlauten.

#### 4.d Stuttgart



Die apachecon<sup>22</sup> in Stuttgart bildete die letzte Station meiner Reihe von Interaktionsereignissen. Auch hier konnte ich viele hochinteressante Gespräche zu der Thematik magnoliaQT und der Zukunft von multimedialen Inhalten auf dem Web führen. Besonders herauszustreichen ist die Konversation mit Noel Bergman von der ASF<sup>23</sup>. Wir unterhielten uns über Portlets<sup>24</sup> und die Möglichkeiten serverbasierter Manipulationen von Medienfiles in Kombination mit dieser Technologie. Portlets werden sehr oft durch Bilder und Animationen unterstützt - ein weiteres mögliches Einsatzgebiet für eine adaptierte Version von magnoliaQT.

## 5. KONKLUSION

### 5.a Workshops

Die ursprünglich geplanten Workshops zu der Thematik „Multimediale Inhalte auf dem Internet“ mussten aufgrund der Verzögerung in der Entwicklung von magnoliaQT verschoben werden. Ich kam zum Schluss, dass ein Austausch mit einer Zielgruppe erst Sinn macht, wenn die Grundfunktionalität von magnoliaQT gewährleistet ist und somit das System innerhalb der Workshops auch eingesetzt werden kann.

Da ich bereits Kontakt mit etlichen Institution aufgenommen habe und das Interesse seitens der Vertretern dieser Ausbildungsstätten nach wie vor besteht, freue ich mich sehr auf die Durchführung dieser Workshops. Momentan bin ich im Gespräch mit Gerd Wippich von der FH Aarau. Er ist sehr interessiert und wir werden im November konkret für das kommende Jahr planen. Ebenfalls Marcel Humair, Schulleiter der Kreisschule Thierstein West, ist seit längerem am Angebot interessiert und möchte über eine längere Dauer die Workshops in den Informatik Unterricht einfließen lassen.

### 5.b Finanzen

Lizenzkosten für die IDE, Reise- und Repräsentationsspesen, Bücher sowie Material und Spesen für den Medienbeitrag verschlangen ca. CHF 10'000.00. Diesen Betrag finanzierte ich aus eigener Tasche und werde ihn mit zukünftigen magnoliaQT Implementierungsaufträgen decken.

Den Wert des Know-Hows und der Unterstützung durch obinary ag und tomato kann nicht beziffert werden.

Externe Grafik- und Textaufträge wurden im Austausch gegen noch zu leistende Programmierarbeit erledigt.

Ein spezieller Dank geht an obinary ag, welche Kost und Logie in San Francisco gesponsert haben.

### 5.c Bezug zu nomadix

Es hat sich gezeigt, dass das Bereisen verschiedener Orte (Bern, London, San Francisco, Stuttgart) zum Know-How Transfer, für mich essentiell war. Viele der Bekanntschaften, Optionen und Lösungswege aus scheinbaren Sackgassen, die sich in diesem Jahr ergaben, kamen nur durch direkte Konfrontation vor Ort zustande. Dies trotz der Möglichkeiten von Chat, Newsgroups, Foren, Mailinglisten etc. Die Verbindlichkeit und Relevanz eines persönlichen Gespräches lässt sich zum Glück tatsächlich nicht durch einen Smiley im Chat ersetzen.

### 5.d Medienbeitrag nomadix

Als Beitrag für die Wanderausstellung wurden sieben Spots gedreht. Bewusst wurde in diesen Beiträgen nur das Erfassen von Filmmaterial durch Consumer gezeigt, auch wenn sich magnoliaQT nicht ausschliesslich an diese Zielgruppe richtet. Hiermit wird die einfache Handhabung des Produkts unterstrichen. Durch die direkte Überblendung von wirklichem Geschehen auf das Web wird die Publikation als beinahe transparenter Schritt beschrieben. 29 Personen waren an der Herstellung der Insgesamt 7 Spots involviert. Die Filme wurden ebenfalls als DVD an alle Beteiligten ausgehändigt.

### 5.e Fazit

Neben dem enormen Lerneffekt in Sachen Teamleitung, verteiltem Entwickeln von Software und nicht zuletzt technischem Wissen konnte ich in meinem Diplom etliche zukunftssträchtige Kontakte knüpfen und ein Produkt entwickeln, welches mir als Basis für eine eventuelle kommende Selbstständigkeit dienen könnte. Die Erfahrung, verschiedene Produkte zu vereinen, war sehr lehrreich. Gerade der Umgang mit magnolia hat gezeigt, dass wenn auf Software aufgebaut wird, welche selbst noch in Entwicklung ist, grosszügiges Kalkulieren angebracht ist. So muss immer wieder mit Verzögerungen und Zusatzaufwand gerechnet werden, um mit den Neuerungen der letzten Version Schritt halten zu können. Dank dem ausgezeichneten Verhältnis zu obinary ag konnte dies aber auch immer wieder als Chance wahrgenommen werden und Erweiterungswünsche unsererseits wurden rasch implementiert.

## 6. ZUKUNFT

### 6.a Erste Auftragsarbeiten

Bereits liegen Anfragen auf dem Tisch um magnoliaQT produktiv einzusetzen. So wird das System z.B. im praktischen Teil der Doktor Arbeit von Adnan Hadzi zum Thema „collective documentaries“ an der University of London zum Tragen kommen. Auch erste Offerten für kommerzielle Nutzungen (Goldschmied, Video-Kunstsammlung) können in kürze gestellt werden. Zudem konnte ich Nuada AG, Basel als Hostingpartner gewinnen und habe somit ein äusserst konkurrenzfähiges Angebot auf welches ich bauen kann.

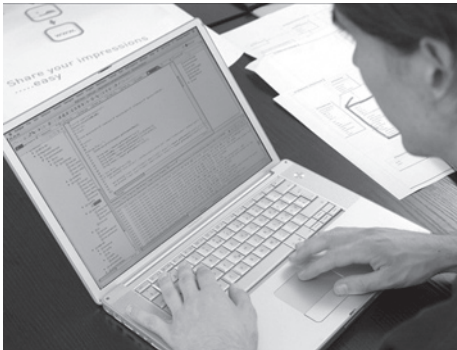
### 6.b Weiterentwicklung

Sobald der letzte Bug aus der Version 1.0 verbannt ist, geht es daran über die Weiterentwicklung von magnoliaQT nachzudenken.

Die Liste von möglichen Erweiterungen ist nahezu endlos. Batchbearbeitung von Ordnern, weitere Bildmanipulationen (z.B. Redeye Reduction), Filter (z.B. Fade) für Filme, Equalizer Funktionalitäten für Sound oder eine Anbindung an einen Streamingserver könnten eingeführt werden.

Überlegungen das Ausgabeformat in Flash zu ändern (was technisch ebenfalls ohne weiteres realisierbar ist und noch ein grösseres Zielpublikum erreichen würde) oder Implementierungen für andere Content Management Systeme (was die Fixiertheit auf magnolia auflösen würde) stehen zur Diskussion.

Doch zuerst muss die Reaktion auf die Version 1.0 abgewartet werden.



## 7. GLOSSAR

<b><sup>1</sup>Studie</b>	PricewaterhouseCoopers White Paper Details Key Applications for Broadband Internet Mehr: <a href="http://www.pwc.com/extweb/ncpressrelease.nsf/docid/F528D0637B7D01B585256E30006EEA4F">http://www.pwc.com/extweb/ncpressrelease.nsf/docid/F528D0637B7D01B585256E30006EEA4F</a>
<b><sup>2</sup>opensource</b>	Software, welche im quelloffenen Format vorliegt und somit Einblick in den Quelltext ermöglicht. Mehr: <a href="http://de.wikipedia.org/wiki/Open_Source">http://de.wikipedia.org/wiki/Open_Source</a>
<b><sup>3</sup>Content Management System</b>	Ein System zur Verwaltung von Inhalten einer Website. Mehr: <a href="http://de.wikipedia.org/wiki/Content-Management-System">http://de.wikipedia.org/wiki/Content-Management-System</a>
<b><sup>4</sup>SourceForge</b>	Das weltweit grösste Downloadportal für opensource Projekte. Mehr: <a href="http://sourceforge.net/">http://sourceforge.net/</a>
<b><sup>5</sup>Java</b>	Eine von der Firma Sun Microsystems entwickelte Programmiersprache. Mehr: <a href="http://java.sun.com">http://java.sun.com</a>
<b><sup>6</sup>JCR 170</b>	Content Repository for Java Application Programming Interface zur Standardisierung des Zugriffs auf Datenbehälter. Mehr: <a href="http://www.jcp.org/en/jsr/detail?id=170">http://www.jcp.org/en/jsr/detail?id=170</a>

<b><sup>7</sup>QuickTime</b>	Eine Produktfamilie der Firma Apple für das Anfertigen, Bereitstellen und Wiedergeben digitaler Medien. Mehr: <a href="http://www.apple.com/quicktime/">http://www.apple.com/quicktime/</a>
<b><sup>8</sup>J2EE</b>	Java 2 Platform, Enterprise Edition, ist die Spezifikation einer Standardarchitektur für die Entwicklung von verteilten, mehrschichtigen Anwendungen mit modularen Komponenten. Mehr: <a href="http://java.sun.com/j2ee/">http://java.sun.com/j2ee/</a>
<b><sup>9</sup>IDE</b>	Integrated Development Environment, Programm zur Entwicklung von Software. Mehr: <a href="http://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung">http://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung</a>
<b><sup>10</sup>Eclipse</b>	Opensource IDE. Mehr: <a href="http://eclipse.org">http://eclipse.org</a>
<b><sup>11</sup>svn</b>	Eine opensource Software zur Versionsverwaltung von Quellcode. Mehr: <a href="http://subversion.tigris.org/">http://subversion.tigris.org/</a>
<b><sup>12</sup>Java Virtual Machine</b>	Eine Laufzeitumgebung und virtuelle Maschine für Software. Mehr: <a href="http://java.sun.com/docs/books/vmspec/">http://java.sun.com/docs/books/vmspec/</a>
<b><sup>13</sup>Java Remote Method Invocation</b>	Aufruf einer Methode, eines Java-Objekts in einer anderen JVM. Mehr: <a href="http://java.sun.com/products/jdk/rmi/">http://java.sun.com/products/jdk/rmi/</a>



<b><sup>14</sup>Caching Mechanismus</b>	Zwischenspeicher für Objekte und Informationen zur Performanccesteigerung.
<b><sup>15</sup>I18n</b>	Ein Mechanismus zur Extraktion sämtliche vom Endbenutzer sichtbaren Textinformation aus dem Programm Quelltexte. Mehr: <a href="http://java.sun.com/j2se/corejava/intl/index.jsp">http://java.sun.com/j2se/corejava/intl/index.jsp</a>
<b><sup>16</sup>Kompilieren</b>	Übersetzungsvorgang von Quellcode zu Bytecode. Mehr: <a href="http://de.wikipedia.org/wiki/Kompilieren">http://de.wikipedia.org/wiki/Kompilieren</a>
<b><sup>17</sup>Openlaw</b>	Plattform für Recht und freie Software. Mehr: <a href="http://www.openlaw.ch">http://www.openlaw.ch</a>
<b><sup>18</sup>Produktechachtung in magnoliaQT</b>	Einbindung eines closedsource Produkts [QT4J] in ein opensource Produkt [magnoliaQT], welches wiederum in einer weiteren opensource Software [magnolia] zum Einsatz kommt.
<b><sup>19</sup>LGPL</b>	GNU Lesser General Public License. Mehr: <a href="http://www.gnu.org/copyleft/lesser.html">http://www.gnu.org/copyleft/lesser.html</a>
<b><sup>20</sup>LOTS</b>	(Let's open the source) Eine Tagung zur Nutzung und Entwicklung freier und opensource Software. Mehr: <a href="http://www.lots.ch/">http://www.lots.ch/</a>

<sup>21</sup>**WWDC**

(Worldwide Developers Conference) Eine einwöchige Konferenz für Apple Softwareentwickler.

Mehr: <http://developer.apple.com/wwdc/index.html>

<sup>22</sup>**apachecon**

Offizielle Konferenz der Apache Software Foundation.

Mehr: <http://apachecon.com>

<sup>23</sup>**ASF**

Apache Software Foundation.

Mehr: <http://www.apache.org/>

<sup>24</sup>**Portlets**

Standardisierte, wiederverwendbare Java-Komponente, welche in Portalseiten zur Anwendung kommen.

Mehr: <http://jcp.org/en/jsr/detail?id=168>

**share your impression**

**...easy**



## 8. KONTAKT

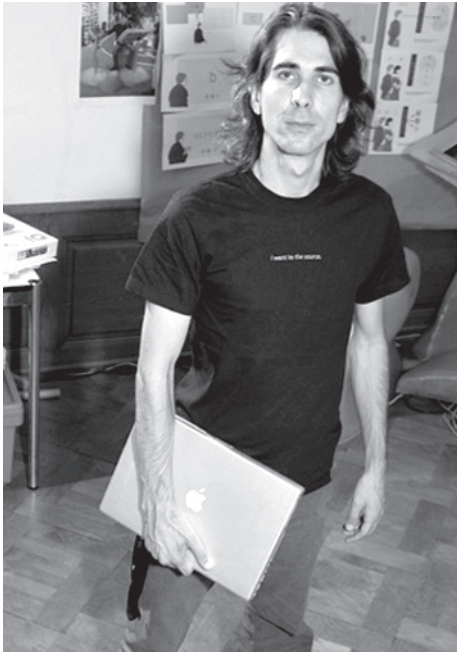
Thomas Martin  
Dipl. Interaktionsleiter FH

Mörsbergerstrasse 7  
CH-4057 Basel

+41 76 558 67 90

t.martin@hyperwerk.ch  
tm@thomas-martin.ch  
info@magnoliaqt.info

<http://www.thomas-martin.ch/>  
<http://www.magnoliaqt.info>



University of  
Applied Sciences  
Basel

**FHBB**

## **Department HyperWerk**

### **Thomas Martin**

Dipl. Interaktionsleiter FH

Mörsbergerstrasse 7  
CH-4057 Basel

+41 76 558 67 90

t.martin@hyperwerk.ch  
tm@thomas-martin.ch  
info@magnoliaqt.info

<http://www.thomas-martin.ch/>  
<http://www.magnoliaqt.info>