

Die Benutzung des Styleguides für Software-Entwickler optimieren

Herausforderungen und Perspektiven



Maria Rauschenberger

MSP Medien Systempartner GmbH & Co. KG
Peterstraße 28–34
26121 Oldenburg
Maria.Rauschenberger@medien-systempartner.de

Heike Sinning

Hochschule Emden/Leer
Constantiaplatz 4
26723 Emden
heike.sinning@gmx.de

Jörg Thomaschewski

Hochschule Emden/Leer
Constantiaplatz 4
26723 Emden
jt@imut.de

Abstract

Der Styleguide ist als Artefakt in Software-Projekten etabliert, da er konkrete Vorgaben für die visuelle Gestaltung und das Layout bietet. In der Praxis ist zu beobachten, dass der am Anfang erstellte Styleguide nicht zwangsläufig mit dem finalen Produkt übereinstimmt. Doch wieso werden die Richtlinien aus dem Styleguide nicht übernommen? Eine genaue Problembetrachtung zeigt, dass der Styleguide ebenso wie der Software-Entwicklungsprozess iterativ zu erstellen und einfach zu nutzen sein sollte. Der hier dargestellte Lösungsansatz zeigt erste Ideen für ein optimiertes Vorgehen zu Erstellung und Benutzung des Styleguides.

Keywords:

/// Styleguide
/// Agiles Projektmanagement
/// Dokumentation

Einleitung

Von einer Idee bis zum fertigen Produkt ist es ein langer Weg. Um das Produktziel zu erreichen, werden in der Softwareentwicklung Prozesse, Artefakte und Maßnahmen zur Qualitätssicherung festgelegt. Für die Festlegung des Designs wird oftmals ein Styleguide erstellt. Dieser gehört für viele Unternehmen zur Standard-Dokumentation sowohl in der klassischen Softwareentwicklung als auch in agilen Prozessen.

Ein Styleguide wird in der Literatur wie folgt beschrieben:

- „Styleguides: Sie bestehen aus einem Satz von sehr konkreten Richtlinien und/oder Spezifikationen mit dem Ziel der Vereinheitlichung von Systemen eines bestimmten Typs oder Herstellers. Sie beschreiben ein grundsätzliches Layout-Rahmenmodell, in das die Inhalte eingefügt werden“ (Sarodnick & Brau 2011).
- „Styleguides: konkrete Vorgaben für die visuelle Gestaltung und das Layout einer bestimmten Benutzeroberfläche. Styleguides beschreiben Aussehen und Verhalten (Look & Feel) von User-Interface-Elementen, abhängig von der eingesetzten Technologie“ (Richter & Flückiger 2010).

Die Vorteile liegen in der klaren Festlegung auf ein Grunddesign für ein Produkt bzw. ganze Produktreihen. Vorab definierte Farben, Formen, Anordnungen und Größen sollen nachträgliche Auseinandersetzungen über das grafische Design reduzieren. Dem Entwickler werden konsistente Designvorlagen für Funktionen und Designprobleme vorgeschlagen (Iconstorm 2010) und die Kunden bzw. Usability-Experten zur objektiven Bewertung befähigt (Sarodnick & Brau 2011). Aus der Sicht von Ortlieb ist der Styleguide auch ein Nachschlagewerk, welches als Checkliste beim Design-Prozess verwendet werden kann (Ortlieb 2012).

1. Verwendung des Styleguides in der Praxis

In unterschiedlichen Unternehmen haben die Autoren Erfahrung in der Anwendung mit dem Styleguide gesammelt und erkannt, dass der Styleguide nicht wie geplant eingesetzt oder verwendet wird. Unternehmen und Mitarbeiter profitieren damit nicht von den prinzipiellen Vorteilen des Styleguides.

Der Umgang mit dem Styleguide wirft die folgenden Fragenstellungen auf:

- Sind es die menschlichen und teilweise organisatorischen Eigenschaften, die einen Styleguide nicht funktionieren lassen?
- Ist die Vielfalt der anfallenden Artefakte, Dokumente, Ablageorte und Programme während des Software-Entwicklungsprozesses problematisch, die den Entwickler jeweils von seiner ursprünglichen Aufgabenstellung ablenken können?
- Ist ein Styleguide eine Voraussetzung oder eher ein Arbeitsdokument auf dem Weg zu einem passenden Design für das Produkt?

In der praktischen Verwendung von Styleguides können zwei größere Problematiken den Vorzügen gegenübergestellt werden. Zum einen können **Prozessprobleme** und zum anderen **Umsetzungsprobleme** festgestellt werden. Fehler! **Verweisquelle konnte nicht gefunden werden.** soll verdeutlichen, dass Prozess- und Umsetzungsprobleme parallel zueinander existieren und damit die Benutzung des Styleguides erschweren oder sogar verhindern können. **[Abb. 1]**

Der in einer Konzeptionsphase erstellte Styleguide ist an die initialen Anforderungen der Software angepasst. Die

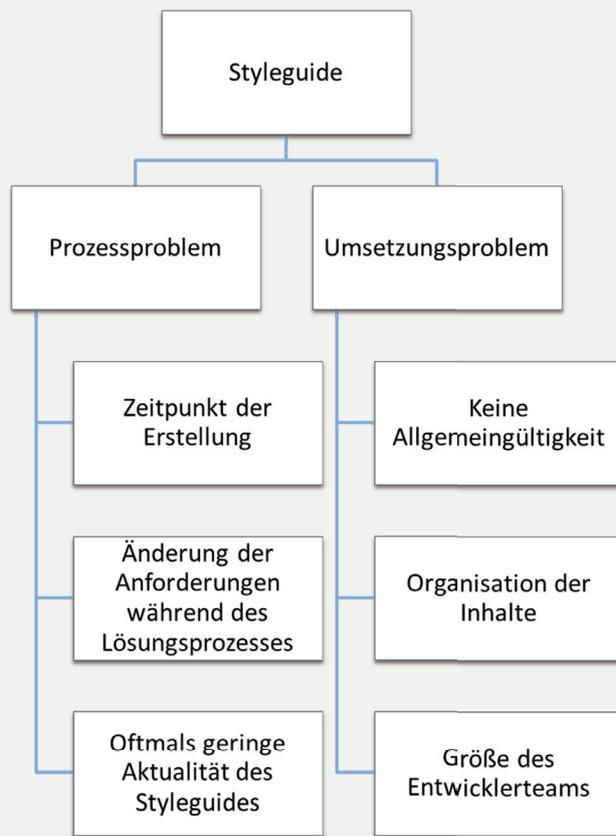


Abb. 1.
Abhängigkeit der Prozess-
und Umsetzungsprobleme

Prozessprobleme entstehen durch die sich im Entwicklungsprozess ändernden Anforderungen an eine Software. Besonders bei der Neuentwicklung von Produkten sind in der Anfangsphase nicht alle Funktionen oder Abläufe abschließend geklärt, insbesondere bei der Verwendung agiler Entwicklungsprozesse. Es werden während des gesamten Human-Centered Design Prozesses Anpassungen vorgenommen, auch nach einer Installation (Ortlieb 2012).

Gut wäre ein Styleguide für alle (ähnlichen) Produkte eines Unternehmens. Allerdings sind eventuell nicht alle Produkte eines Unternehmens durch die verschiedenen Ziele und Funktionen mit einem Styleguide vernünftig abzudecken. Dies führt in der Praxis dazu, dass innerhalb eines Unternehmens mehrere Styleguides zu verschiedenen Projekten, Produkten oder Bereichen erstellt werden müssen, die sich nicht widersprechen sollten. „Beispiele für diese typischen Probleme sind die Schwierigkeit,

relevante Informationen innerhalb des umfangreichen Materials aufzufinden und Warten der Informationen“ (Schrammel et al. 2010). Überdies ist die mangelnde Kommunikation der Aktualisierungen an die Projektteilnehmer problematisch und es bedarf einer Person, die die Dokumente auf den neuesten Stand hält, die Aktualität gewährleistet und die Kommunikation zum Projektteam vornimmt. Das Prozessproblem verhindert zusätzlich zum Auffinden eine schnelle Akzeptanz und Verbindlichkeit des Styleguides und verstärkt überdies das Umsetzungsproblem.

Schubert et al. (Schubert et al. 2010) beschreiben, dass der Erfolg eines Styleguides abhängig ist von der inhaltlichen Qualität, der organisatorischen Verankerung in dem Entwicklungsprozess und der Sensibilität der Software-Entwickler für konsistente Gestaltung der Oberflächen. Die Kommunikation des Styleguides an

das Projektteam ist so auszurichten, dass die hohe Bedeutung des Styleguides vom Projektteam anerkannt wird. „Die besondere Herausforderung an eine erfolgreiche Kommunikation und Dokumentation liegt darin, trotz unterschiedlicher Fachsprachen, Perspektiven und Dokumentationsstilen, geeignete Werkzeuge zu wählen und ein gemeinsames Verständnis zu erzeugen“ (Herdle et al. 2010). Buxton (Buxton 2010) bestätigt, dass die Kommunikation unter den Projektbeteiligten wichtig ist, um ein Design erfolgreich umzusetzen.

Beim **Umsetzungsproblem** geht es um die Akzeptanz des Styleguides bei den Projektteilnehmern, besonders bei den Software-Entwicklern. Das Umsetzungsproblem gestaltet sich deutlich diffuser als das Prozessproblem. Die dokumentenbasierte Ablage und damit statische Form des Styleguides ist für eine einfache Weiterentwicklung ungeeignet. Dieses oftmals starre Vorgehen ähnelt dem eines Pflichtenhefts, welches ebenfalls am Anfang eines Projektes festgelegt wird. In beiden Dokumententypen sind größere Änderungen nur mit hohem Aufwand möglich und entsprechen nicht dem Projektalltag in kleineren IT-Abteilungen. Während die dokumentengetriebene Softwareentwicklung (mittels Pflichten- und Lastenheften) immer mehr von agilen Prozessmodellen (Scrum, Kanban) abgelöst wird (vgl. Begel & Nagappan 2007, Salo & Abrahamsson 2008), findet man bei den Styleguides weiterhin die eher statische Dokumentationsform vor. Um den Styleguide aktuell zu halten, kann es von Vorteil sein, wenn die laufenden Änderungen durch einen dynamischen (iterativen) Ansatz festgehalten werden. In diesem Zusammenhang hat die Agentur Iconstorm festgestellt, dass die Aktualität inkl. Erstellungsprozess auf 36 Monate begrenzt ist. „Überraschend ist die Tatsache, dass die langfristige Designstrategie nur noch bei der Minderheit der teilnehmenden Unternehmen, länger als drei Jahre gilt. Lediglich 20 Prozent geben eine Ausrichtung der Strategie auf mehr als zehn Jahre an. [...] Die Halbwertszeit sinkt, zwischen Erstellungsprozess und Erneuerung liegen nur selten noch mehr



als 36 Monate. Berücksichtigt man eine durchschnittlich angegebene Dauer bei der Erstellung von sechs Monaten, gilt ein Styleguide heutzutage maximal 24 Monate“ (Iconstorm 2010).

In vielen Unternehmen werden die Styleguides meist als Dokument auf Laufwerken, im Intranet oder in Wiki-Systemen abgelegt. Dieser dezentrale und von den Arbeitswerkzeugen des Entwicklers abgegrenzte Ablageort unterbricht den Arbeitsfluss und stört die Effektivität. Für den Entwickler bedeutet dies eine aktive Suche nach neu einzubindenden UI-Elementen außerhalb seiner Arbeitsumgebung und damit eine Störung seines Arbeitsflusses und des Implementierungsprozesses. Darüber hinaus ist eine gewisse Organisation der Inhalte des Styleguides notwendig. Hier wird von der Agentur Iconstorm empfohlen, „lieber eine Vorlage statt einem Beispiel“ (Iconstorm 2010) abzubilden. Darüber hinaus sind die Vorlagen an einer zentralen und kommunizierten Stelle abzuliegen, um die Auffindbarkeit zu verbessern. „Oft ist Mitarbeitern, die neue Dokumente anlegen wollen, nicht klar, wo der Inhalt im Wiki am besten untergebracht ist. Oder aber sie sind sich unsicher, ob sie ein neues Dokument anlegen oder ein bereits bestehendes Dokument ergänzen sollen“ (Seibert et al. 2011). Dieser Umstand kann dafür sorgen, dass Redundanzen die Übersichtlichkeit verhindern.

Die Größe des Entwicklerteams kann ein Problem darstellen. Es erfordert meist mehrere Abstimmungsmeetings, wenn alle Anforderungen eines Projektes in das Projektteam getragen werden. Laut Litke geht die Effektivität tendenziell ab einer Größe von 15 Mitarbeitern zurück (Litke 2007). Ändern sich nun zusätzlich während des Projektes die Anforderungen des Styleguides müssen diese Abstimmungsrunden häufiger durchlaufen werden. „Nur wenn die Teamgröße auf die genannte Mitarbeiterzahl [5–8] beschränkt wird, ist ein vollständiger Informationsaustausch möglich. Er erlaubt es, dass ein Projektteam mit einem geringen Verwaltungsaufwand arbeiten kann und den direkten Kontakt

zwischen den Mitarbeitern aufrechterhält. Bei steigender Teamgröße nimmt der Aufwand für die Informationsvermittlung überproportional zu“ (Litke 2007).

Richter und Flückiger empfehlen den Styleguide mit guten Beispielen und Hilfsmitteln anzureichern, um ein „gemeinsames Verständnis und das sinnvolle Anwenden der Regeln zu fördern“ (Richter & Flückiger 2010). Diese Vorgehensweise wird auch von Ortlieb befürwortet (Ortlieb 2012). Überdies schlagen Richter und Flückiger eine Kategorisierung des Styleguides in verschiedene Bereiche vor. Folglich kann es einen hersteller- oder plattformabhängigen Styleguide geben, welcher „das vorgesehene **Look&Feel** der Applikationen eines bestimmten Betriebssystems mit dem Ziel einer konsistenten Anwendung aller GUI-Elemente wie Eingabefelder, Listboxen, Schaltflächen etc.“ beschreibt (Richter & Flückiger 2010). Zusätzlich wäre es möglich, einen Unternehmens-Styleguide bereitzustellen, welcher die Vorgaben aus dem Corporate-Design in den Applikationen reflektiert. Weiter wird eine Differenzierung vorgenommen, „ob es sich um Richtlinien für die firmeninterne Applikationslandschaft handelt oder um Anwendungen bzw. Produkte für externe Kunden“ (Richter & Flückiger 2010). Die Verwendung eines Projekt-Styleguides stellt die Richtlinie für „die Konsistenz der Benutzerschnittstelle bei der Entwicklung einer Applikation (z.B. beim Einsatz verschiedener UI Designer) oder von Produkten für den Endkunden sicher“ (Richter & Flückiger 2010).

2. Ideen, wie ein Styleguide vom Projektteam genutzt werden kann

Nachdem die Erwartungen an den Styleguide nicht mit der praktischen Erfahrung der Autoren übereinstimmen, soll eine Lösung für das Problem aufgezeigt werden.

Ein möglicher Lösungsansatz für das **Prozessproblem** sind die organisatorischen Voraussetzungen des Styleguides. Wie

Schubert et al. (Schubert et al. 2010) erläutern, ist eine zugeordnete Verantwortlichkeit für das Erstellen und Kommunizieren des Styleguides relevant für den erfolgreichen Einsatz. Der Verantwortliche kümmert sich darum, dass der Styleguide kommuniziert wird und allen Projektteilnehmern die allgemeinen Richtlinien und der Umgang mit der Dokumentation verständlich sind. Relevant ist, dass sich Anforderungen über den Projektzeitraum ändern können. Der Styleguide sollte deshalb immer die aktuellsten Absprachen enthalten und damit ist eine schnelle und einfache Änderung im Styleguide erforderlich. Dies wird auch von Herdle et al. (Herdle et al. 2010) und (Schrammel et al. 2010) unterstützt.

Der initiale Styleguide gilt damit als Grundlage für Erweiterungen und die Entwickler sind verpflichtet, den Styleguide-Verantwortlichen auf bisher nicht festgehaltene Regelungen aufmerksam zu machen. Der Verantwortliche kümmert sich dann wiederum um das Festlegen der neuen Richtlinien und die Kommunikation ins Projektteam. Dieses Verfahren orientiert sich maßgeblich an dem agilen Prozessgedanken von Scrum und soll das iterative Anpassen der Styleguide-Anforderung organisieren. In diesem Zusammenhang beschreiben Schubert et al. (Schubert et al. 2010), dass die „organisatorische Verantwortung des Styleguides in den Entwicklungsprozess“ und die „Sensibilität der Entwickler für ansprechende und konsistente Gestaltung“ wichtige Erfolgsfaktoren für den Styleguide sind.

Der Wirkungskreis des zukünftigen Styleguides sollte vor Erstellung festgelegt werden und es wird vorgeschlagen den Wirkungskreis einzugrenzen (Richter & Flückiger 2010). Wie bereits in dem Abschnitt der Einleitung genannt, könnte auch ein kundenspezifischer Styleguide pro Produkt etabliert werden. Hierbei ist darauf zu achten, um welche Unternehmensgröße und Produktpalette es sich handelt. Unternehmensgrößen von unter 200 oder über 7000 Mitarbeiter sind Rahmenbedingungen, welche sie auf die Anforderungen des Styleguides (Benutzergruppe,

Produktvielfalt, Anzahl der Produkte,...) auswirkt. Dieser Ansatz bezieht sich im Gegensatz zu Schrammel auf mittelständische Unternehmen.

Im Unternehmen sollte eindeutig festgelegt werden, wo die Styleguides von den Projektteilnehmern, für jeden Kunden, jedes Projekt oder andere Bereiche zu finden sind. Hierbei erscheint wichtig, dass ein zentraler Ablageort (physikalisch oder virtuell) festgehalten wird.

Bothmer (Bothmer 2011) beschreibt einen dynamischen Styleguide, welcher über den gesamten Entwicklungsprozess besteht und iterativ angepasst wird. Zudem geht er davon aus, dass die Plattform Confluence von Atlassian Software Systems als Enterprise Wiki hervorragend die Anforderungen an einen dynamischen Styleguide regelt. Damit ist allerdings nicht die Problematik der aktiven Suche nach Styleguide Vorlagen der Entwickler geregelt.

Hiermit verlagern sich die Lösungsansätze in die Richtung des **Umsetzungsproblems**, denn der Ablageort könnte direkt die Entwicklungsumgebung des Entwicklers selbst sein. Hintergrund ist die bereits

starke Suchbereitschaft der Entwickler nach Methoden oder Funktionen ihrer Programmiersprache im Internet oder innerhalb der Methodenbibliothek der Entwicklungsumgebung. Im Artikel von Kirstein et al. (Kirstein et al. 2012) wird beschrieben, dass eine Orientierung „an der Arbeitsweise der Entwickler und deren Entwicklungswerkzeuge“ bestehen sollte. Zudem schlagen Kowalik et al. (Kowalik et al. 2011) die Integration in die Entwicklungsumgebung vor.

Ein österreichisches Technologieunternehmen hat seinen Styleguide „mit dem Einsatz eines strukturierten Authoring-Prozesses basierend auf DITA-Maps“ konstruiert (Schrammel et al. 2010). Damit bieten sie eine anwenderorientierte Dokumentation des Styleguides, PDF-Dateien, die Integration in das Hilfesystem von Windows und die Integration in eine Entwicklungsumgebung an. Das von Schrammel et al. (Schrammel et al. 2010) dargestellte System verfügt über eine mächtige Struktur, eine hohe Flexibilität der Funktionen und Styleguide-Benutzergruppen, die jedoch mit einem hohen initialen Aufwand verbunden sind.

Das Unternehmen Apple erläutert in seinem Styleguide, dass das grundsätzliche Verständnis für ein Produkt und seine Gestaltung hilft, ein Produkt zu entwerfen, welches den Benutzer erfreut (Apple Inc. 2012). Apple hat Elemente seines Styleguides als Framework in seiner Entwicklungsumgebung Xcode integriert.

Zudem bieten die Open-Source Produkte wie Eclipse oder kommerzielle Produkte wie Visual Studio von der Firma Microsoft die Möglichkeit Plug-Ins zu integrieren. Vorteilhaft wäre die Integration des Styleguides in die Entwicklungsumgebung, da der Entwickler diese Software jetzt schon täglich für seine Implementierung benutzt. Zunächst kann eine simple Darstellung in der Methodenbibliothek erfolgen, wie in Abbildung 2 dargestellt. [Abb. 2]

Auch könnte die Idee der Firma Iconstorm (Iconstorm 2010) weiter interpretiert werden und eine Art **Microsoft Word Formatvorlage** für bestimmte Methoden der Programmierung in die Entwicklungsumgebung eingegliedert werden. Die Formatvorlage kann dann zum Beispiel für Formulare (Kontakt- und Bestellformulare), Buttons („Weiter“, „Abschicken“), AGBs,

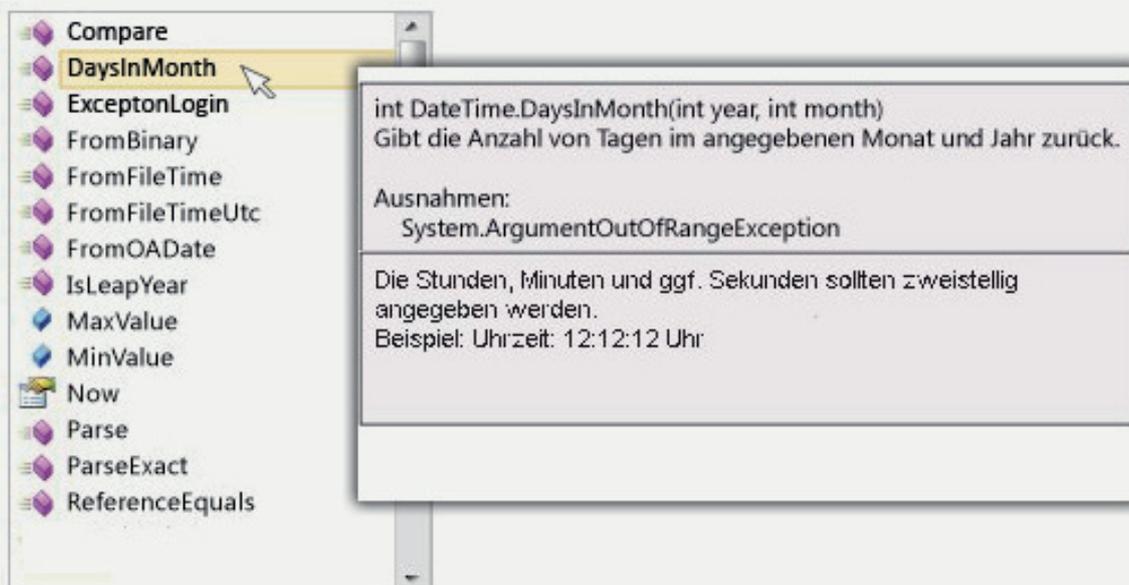


Abb. 2. Bild von Visual Studio mit einer möglichen Darstellung einer Richtlinie aus dem Styleguide



„rechte Box“ im Webseitenaufritt, Detailseiten oder Fehlermeldungen gelten. Als Beispiel ist die Fehlermeldung in Abbildung 3 dargestellt. [Abb. 3]

Diese Formatvorlagen können die heutigen UI-Patterns als Grundlage haben. Sinnvoll ist dabei eine Auswahl an zum Beispiel zwei UI-Pattern für den Entwickler und der zusätzlichen manuellen Änderbarkeit.

Damit wäre ein schnellerer und integrierter Zugriff für den Entwickler gesichert und keine aktive Suche nach Styleguide Informationen nötig. Hilfestellung könnten auch die bereits bestehenden Frameworks wie Zend oder Microsoft Visual Studio LightSwitch geben, welche ein Grunddesign mitbringen. Eine Erweiterung durch Customizing der Methoden muss noch geprüft werden.

Der Entwickler sollte an der Weiterentwicklung des Styleguides beteiligt werden und seine Ideen frühzeitig einbringen. Als weiterer Optimierungsansatz sollte die Versionierung der Artefakte (Styleguide-Dokumente/Texte/Formatvorlagen) über die bereits bestehende Versionsverwaltung (beispielsweise GIT, SVN oder TFS) der Entwicklungsumgebung nachvollziehbar abgebildet werden. Gesetzt den Fall, dass sich eine Designvorlage in der Praxis nicht

bewährt hat, ist ein Zurückkehren zu der vorherigen Version problemlos möglich.

Natürlich ist es auch möglich, eine physikalische Abbildung als Poster im Büro der Entwickler aufzuhängen. Hierbei ist zu beachten, dass man diese aktuell halten muss.

Das Ziel ist die einfache Verwendung und schnelle Verbreitung der Änderungen des Styleguides an die Softwareentwickler.

3. Fazit

Der Styleguide ist als relevantes Artefakt in der Software-Entwicklung anerkannt. Nach der genaueren Betrachtung des Styleguides können die Ursachen für die praktische Benutzung aufgeführt werden.

Es liegt an menschlichen und teilweise organisatorischen Eigenschaften, dass ein Styleguide oftmals nicht wie vorgesehen funktioniert. Hinzukommt, dass die Vielzahl an anfallenden Artefakten, Dokumenten und Programmen und der daraus resultierenden Medienbruch destruktiv für die Effektivität des Entwicklers und der Projektteilnehmer ist. Es wird deutlich, dass in Anlehnung an den agilen Software-Entwicklungsprozess der Styleguide in der Umsetzungsphase iterativ seine

Anforderungen spezifiziert, welche dann wiederum schnellstmöglich dem Projektteam zur Verfügung gestellt werden sollten. Die Probleme können in Prozess- und Umsetzungsprobleme eingeordnet und beschrieben werden.

Zusätzlich zu einer Befragung und deren Auswertung soll in der nahen Zukunft verstärkt auf Lösungsansätze eingegangen werden, um diese auf ihre Realisierbarkeit zu überprüfen. Dafür eignen sich besonders die Open-Source Entwicklungsumgebung **Java Sun Eclipse** und die kommerzielle Entwicklungsumgebung **Microsoft Visual Studio**. Beide Produkte bieten Ansätze für eine Erweiterung mittels Plugins innerhalb der Entwicklungsumgebung.

Weitere langfristige Ziele sind die Erstellung eines ersten Prototyps und die Durchführung eines Probelaufs. Die Voraussetzung für das Erreichen der Ziele stellt eine Kooperation mit einem oder mehreren Unternehmen dar, die einzelne Produkte, Projekte und Entwickler für eine Überprüfung zur Verfügung stellen.

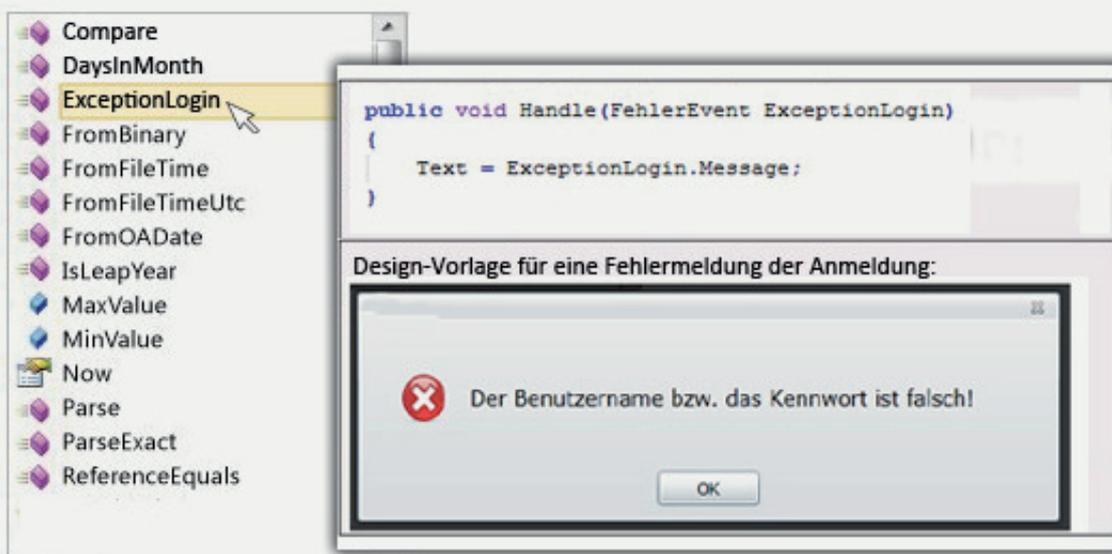


Abb. 2. Entwurf einer möglichen Designvorlage für die Fehlermeldung der Anmeldung

Literatur

1. Apple Inc. (2012). Mac OS X Human Interface Guidelines. online verfügbar unter <http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html>, zuletzt geprüft am 02.07.2013.
2. Begel, A. & Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. In: IEEE Empirical Software Engineering and Measurement, 255–264. Los Alamitos. IEEE Computer Society.
3. Bothmer, J. (2011). Tools für die aktive Markenführung. Dynamischer Styleguide. Iconstorm – Agentur für Markentechnik GmbH & Co. KG. Online verfügbar unter: http://www.style-guide.org/Kategorien/articles/Dynamischer_Styleguide.html. Zuletzt geprüft am 30.05.2013.
4. Buxton, Bill (2010). Sketching User Experiences. Amsterdam: Morgan Kaufmann.
5. Herdle, K., Kurzak, M., Kratzheller, J. & Bierkandt, J. (2010). Gemeinsames Verständnis erzeugen bei der interdisziplinären Gestaltung des neuen Bahnautomaten. In: Brau, H. (Hrsg.): Usability Professionals 2010, 108–113. Stuttgart: German Chapter der Usability Professionals Association.
6. Iconstorm (2010). Styleguide. Gestaltungsrichtlinien-Monitor 2010. Iconstorm – Agentur für Markentechnik GmbH & Co. KG. Online verfügbar unter: <http://www.iconstorm.de/news/56/Iconstorm-verffentlicht-Bulletin-11-2010-zum-Thema-Styleguide>. zuletzt geprüft am 13.02.2013.
7. Kirstein, E., Schoenherr, N. & Schubert, U. (2012). Icon Design im großen Stil. In: Brau, H. (Hrsg.). Usability Professionals 2012, 60–66. Stuttgart: German UPA e.V. .
8. Kowalik, P., Schrepp, M. & Erle, M. (2011). Eingeschränkt Sehen. Eingeschränkt Hören. In: Brau, H. (Hrsg.). Usability Professionals 2011, 66 -70. Stuttgart: German UPA e.V.
9. Litke, H. (2007). Projektmanagement: Methoden, Techniken, Verhaltensweisen. München: Hanser.
10. Ortlieb, W. (2012). Styleguides. Online verfügbar unter: <http://www.se.uni-hannover.de/pub/File/kurz-und-gut/ss2012-proseminar-inf-usabiliy/Ortlieb2012.pdf>. Zuletzt geprüft am 02.07.2013.
11. Richter, M. & Flückiger M. (2010). Usability Engineering Kompakt. Heidelberg: Spektrum Akademischer Verlag.
12. Salo, O. & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations. In: Software, IET 2. 58–64.
13. Sarodnick, F. & Brau, H. (2011). Methoden der Usability Evaluation. Bern: Huber.
14. Schrammel, J., Lugmayr, M., Hämmerle, F., Murtinger, M. & Tscheligi, M. (2010). Flexible und erfolgreiche Implementierung eines User Interface Styleguides mittels eines strukturierten Authoring-Konzeptes basierend auf DITA-Maps. In: Brau, H. (Hrsg.). Usability Professionals 2010, 141–145. Stuttgart: German Chapter der Usability Professionals Association.
15. Schubert, U., Bonhag, W. & Groß, M. (2010). Einsatz von User Interface Patterns bei der Entwicklung von Business-Software. In: Brau, H. (Hrsg.). Usability Professionals 2010, 150 – 156. Stuttgart: German Chapter der Usability Professionals Association.
16. Seibert, M., Preuss, S. & Rauer, M. (2011): Enterprise Wikis. Wiesbaden: Betriebswirtschaftlicher Verlag Gabler.
17. Wilson, C. (2010). Guidance on Style Guides. online verfügbar unter <http://www.stcsig.org/usability/newsletter/0104-style.html>, zuletzt aktualisiert am 24.10.2010, zuletzt geprüft am 24.06.2013.