



Software-Produkte, die Benutzer lieben
Mit Living Styleguide und Continuous X zur optimalen User Experience

Software-Produkte, die Benutzer lieben

Mit Living Styleguide und Continuous X zur optimalen User Experience

Autor

Andreas Lehner – IT-Berater, Trainer und Experte für Design Thinking

Kontakt

Richard Attermeyer
richard.attermeyer@opitz-consulting.com

Impressum

OPITZ CONSULTING Deutschland GmbH
Kirchstr. 6
51647 Gummersbach
+49 (0)2261 6001-0
info@opitz-consulting.com

Disclaimer

Text und Abbildungen wurden sorgfältig entworfen. Die OPITZ CONSULTING Deutschland GmbH ist für den Inhalt nicht juristisch verantwortlich und übernimmt keine Haftung für mögliche Fehler und ihre Konsequenzen. Alle Rechte, z. B. an den genannten Prozessen, Show Cases, Implementierungsbeispielen und Quellcode, liegen bei der OPITZ CONSULTING Deutschland GmbH. Alle genannten Warenzeichen sind Eigentum ihrer jeweiligen Besitzer.

Inhalt

Vorwort	3
Wo hakt es in der Praxis?	3
UX-Optimierung für 3000 interne Nutzer weltweit	3
Wie kann ein Living Styleguide helfen?	3
Die Akteure des Living-Styleguide-Systems	4
Endbenutzer der Software-Produkte	4
Entwickler und Entwicklungsteams	6
UX-Designer	7
Mehr als nur ein technisches Werkzeug	8
Das Zusammenspiel der Akteure	8
Entwicklung mit Continuous X	9
Continuous X im Detail	10
Zusammenfassung und Fazit	12
Referenzen	12

Vorwort

Eine positive User Experience, kurz UX, ist heute entscheidend für den Erfolg von Softwareprodukten. Egal, ob im Consumer-Bereich oder bei Enterprise-Anwendungen: Scheitern die Benutzer an der effizienten Erfüllung ihrer Ziele, suchen sie nach anderen Möglichkeiten und verwenden im schlimmsten Fall sogar „selbstgestrickte“ Lösungen. Der komplette Entwicklungsaufwand war dann im wahrsten Sinne für den Papierkorb. Währenddessen wächst im Schatten der IT ein „Zoo“ an Einzeltools und Anwendungen mit hohen Risiken und Folgekosten für das Unternehmen.

Doch warum bekommen Software-Entwickler die UX nicht besser in den Griff? Die Antwort liegt in der Software selbst. Die ist heute komplex, stark verteilt, vernetzt und unterliegt schnellen und dynamischen Veränderungen. In solchen Systemen die Qualität der UX konstant hoch zu halten, ist eine der größten Herausforderungen in der gemeinschaftlichen Entwicklung von Software-Produkten.

Doch es gibt durchaus Wege, die aus diesem Spannungsfeld herausführen. Zum Beispiel mithilfe eines technologischen Hilfsmittels, das die Benutzer in den Blick nimmt und die größten Risiken und Fallstricke für eine Verschlechterung der UX minimieren hilft. Außerdem braucht es begleitende Prozesse, welche die Sichtweisen von Entwicklerteams und Stakeholdern zusammenbringen und damit auch die UX der Software-Produkte kontinuierlich verbessern. Das Living-Styleguide-System ist genau so ein Hilfsmittel. Ein System, das die UX nicht nur im Fokus hat, sondern diese zu einem wesentlichen Erfolgsfaktor für die Applikation macht.

Ein solches System erfolgreich einzurichten und zu pflegen ist allerdings nicht trivial. Für die Umsetzung benötigen Entwicklerteams eine Menge Know-how und Erfahrung und manchmal auch einen mentalen Change.

Wo hakt es in der Praxis?

Fangen wir vorne an: Wie können wir ein System für eine gute UX nachhaltig entwickeln? Oder anders gefragt, warum gelingt dies oft nicht? Wagen wir dafür mal einen Blick in die Herausforderungen im Entwicklungsalltag:

UX-Optimierung für 3000 interne Nutzer weltweit

Ein IT-Leiterin, nennen wir sie Sandra, leitet die Abteilung für die Software-Entwicklung eines Großkonzerns in der Logistikbranche und möchte eine bestimmte Software-Lösung modernisieren. Umgestellt werden soll auf neue moderne Technologien wie Angular, React, Microservices-Architektur sowie Cloud und Continuous Delivery.

Dabei möchte Sandra auch die UX wesentlich verbessern. Die Lösung ist sehr groß: 3000 User weltweit. Der komplette Konzern nutzt die Lösung. Etliche Benutzergruppen und Rollen verwenden Funktionen, die sich teilweise überschneiden.

Was kann Sandra tun, um die UX nachhaltig positiv zu gestalten und kontinuierlich zu verbessern?

Der Benutzer im Fokus

Sandra ist sich bewusst, dass sie für eine nachhaltige UX konsequent die Benutzer – und zwar vor allem Endbenutzer, die regelmäßig aktiv mit der Anwendung interagieren – mit ihren Bedürfnissen und Zielen in den Mittelpunkt stellen muss.

Sandra nutzt also die Bedürfnisse und die Ziele dieser konkreten Personen als Grundlage, um das Design der Anwendung entsprechend zu entwickeln. Dabei folgt sie den Prinzipien aus einschlägigen Design-Disziplinen. Diese Prinzipien tragen Namen wie Human Centered Design, User Centered Design, Goal-Oriented Design, Design Thinking, Interaktionsdesign usw. und haben alle eines gemeinsam: Der Mensch steht im Mittelpunkt. Also alles perfekt!?

Langsam wird's unübersichtlich

Im Laufe der Produktentwicklung erreicht Sandra einen Punkt, an dem sie sich fragt, wie sie die Vielzahl an unterschiedlichen Designs, Interaktionsmustern und Styles einheitlich und konsistent in hoher Qualität halten kann, ohne die UX einzuschränken. Irgendwann kommt der Zeitpunkt, an dem Benutzer oder Stakeholder genervt aufseufzen, weil der gleiche Button in einem Bereich so und im anderen Bereich anders reagiert. Dann kommt der Tag, an dem selbst Designer und Entwickler den Überblick verlieren: Die Designer können sich nicht mehr erinnern, welche Designs im produktiven Code wie umgesetzt wurden. Und die Entwickler fragen sich, welches Design sie wie nutzen und einbinden sollen und welche Bausteine denn nun valide evaluiert und qualitätsgeprüft waren und welche nicht.

Oder kurz: Eine Software mag am Anfang noch überschaubar und handhabbar erscheinen, mit steigender Komplexität gerät sie irgendwann aus den Fugen.

Wie kann ein Living Styleguide helfen?

Grundsätzlich lassen sich mit dem Living-Styleguide-System Designs langfristig handhabbar gestalten und dabei die Qualität von Design und UX systematisch verbessern. Das System hat viele Spielarten: Bei der einfachen Dokumentation anfangen über Farben, Typographie und Layouts bis hin zu einem dynamischen Software-Produkt, das die Implementierung von neuen Features effizienter und schneller macht.

Letztendlich ist der Living Styleguide mehr als nur ein technisches Werkzeug. Um den Living Styleguide herum etablieren sich agile Prozesse für die Zusammenarbeit von Entwicklern, Designern und weiteren Stakeholdern in einem Team, die alle ein Ziel unterstützen: dem Benutzer eine hochwertige UX zu bieten.

Um das Living-Styleguide-System begreifbarer zu machen, betrachten wir es im Folgenden aus einer etwas höheren Flughöhe, unter Berücksichtigung von UX-Strategie, Business-Aspekten und agilen Prozessen. Uns interessieren vor allem Software-Produkte im Enterprise-Umfeld, wobei es natürlich auch Überschneidungen mit Consumer-Anwendungen gibt.

Die Akteure des Living-Styleguide-Systems

Wer sind die Akteure des Living-Styleguide-Systems? Wer sind seine Benutzer? Und wer gestaltet das System? Vom Living Styleguide profitieren viele verschiedene Nutzergruppen und Stakeholder:

- Endbenutzer der Software-Produkte
- Software-Entwickler (Development Team)
- UX-Designer (Development Team)
- Scrum Master (Scrum Team)
- Produkt Manager (Fachbereich)
- Product Owner (Scrum Team)
- Business Owner, Management (Fachbereich)

Die ersten drei Gruppen schauen wir uns etwas genauer an.

Endbenutzer der Software-Produkte

Welche Personen gelten als „Endbenutzer“?

Als „Endbenutzer“ gelten in unserer Betrachtung Personen, die direkt und regelmäßig mit einer Anwendung interagieren. Für diese Nutzer stellt das Software-Produkt einen wichtigen Teil ihrer täglichen Arbeit dar.

Das Produkt hilft den „Endbenutzern“, ihre individuellen Ziele zu erreichen. Sie „befüllen“ den Living Styleguide, und machen ihn zu einer Quelle für UI-Komponenten, Patterns und Designs, die sich am Ende zu einem Feature oder ganzen Produkt vereinen.

Ob die Anwendung einen Mehrwert für das Unternehmen liefert, wird am Ende, wie man so schön sagt, mit den Füßen entschieden – nämlich durch die Akzeptanz und regelmäßige Verwendung durch die Endbenutzer.

Nicht zu den Endbenutzern in diesem Sinne gehören:

- Personen, die das Software-Produkt nur hin und wieder nutzen oder durch einen Vermittler, wie etwa System-Administratoren oder Personen, die Inhalt aus dem Produkt erhalten.
- Personen, die durch das Software-Produkt in irgendeiner Weise beeinflusst werden oder Entscheidungen über das Software-Produkt treffen müssen, wie etwa Management, Einkauf oder Personalabteilung.

Welche Ziele verfolgen die Endbenutzer?

Jeder Endbenutzer verfolgt eigene Ziele. Methoden der User Research [1] helfen, Ziele und Bedürfnisse der Benutzer zu identifizieren. Auf Basis dieser Daten lassen sich Anforderungen für ein Software-Produkt ableiten. In der Regel bewerten Endbenutzer ein Produkt anhand von allgemeinen Kriterien wie

- Einfachheit,
- Effizienz,
- Spaßfaktor,
- Geschwindigkeit,
- oder Aufwand.

Ebenfalls von großem Interesse sind Fragen, die Endbenutzern tagtäglich während ihrer Arbeit mit einer Software durch den Kopf gehen, wie

- „Ich möchte bessere Entscheidung bei der Auswahl von Bewerbern für ein Unternehmen treffen.“
- „Ich möchte meine Qualitätsprüfungen mit weniger Aufwand durchführen.“
- „Ich möchte nach dem Auftreten eines Fehlers den Produktionsprozess so schnell wie möglich wieder aufnehmen.“
- „Ich möchte eine Fernreise möglichst günstig buchen.“

Das Software-Produkt kann dabei helfen, Ziele wie diese zu erreichen, indem es dem Benutzer hilft, bestimmte Aufgaben effizienter durchzuführen.

Ein weiteres Beispiel aus dem „echten Leben“:

Frederik arbeitet in einem Service-Center. Dort wird er nach der Anzahl erledigter Service-Fälle bezahlt. Je mehr Fälle er abschließen kann, desto mehr verdient er. Sein wichtigstes Arbeitsmittel ist ein Software-Produkt zur Eingabe von Daten. Dauert die Eingabe lange oder ist die Eingabe umständlich, verdient Frederik weniger Geld. Die Höhe seines Gehalts hängt also direkt von der Effizienz des Software-Produkts ab.

Welche Hindernisse gibt es?

Endbenutzer können eine nicht passgenauer UX oder die mangelnde Usability einer Software als große Hürde empfinden. Insbesondere, wenn sie Mängel wie diese aufweist:

- Bedürfnisse, Ziele und Anforderungen der Endbenutzer werden nicht berücksichtigt.
- Ein Mehrwert ist für den Benutzer nicht erkennbar.
- Die Handhabung erscheint kompliziert und setzt Vorkenntnisse und größeren Aufwand bei Bezug, Installation und Anwendung voraus.
- Im Problemfall gibt es keine Unterstützung und Lösung.

Die Top 3 für eine schlechte Usability könnten zusammengefasst so aussehen:

1. Vor der erstmaligen Nutzung ist ein hoher Lernaufwand erforderlich.
2. Aufgaben lassen sich nur auf Umwegen oder mit viel Mühe ausführen.
3. Die Möglichkeit, Fehler zu korrigieren, ist nur sehr eingeschränkt vorhanden.

(Weiterführende Literatur zum Thema Usability unter [\[2\]](#))

Eine gute UX hingegen zeichnet sich vor allem durch das Erleben von Effizienz, Freude und Wertschöpfung bei der Nutzung eines Software-Produkts aus.

Welche Vorteile bietet das Living-Styleguide-System für Endbenutzer?

Der Living Styleguide soll als zentrale Sammlung von UI-Komponenten, Patterns und Designs vor allem eine kontinuierliche Verbesserung der UX ermöglichen. Häufig auftretende Probleme und Risiken, die die Qualität der Benutzererfahrung vermindern, können mit Hilfe des Living-Styleguide-Systems minimiert werden.

Ein Problem, das Endbenutzern in vielen Anwendungen besonders zu schaffen macht, sind inkonsistente Benutzeroberflächen. Dazu ein paar Beispiele aus der Praxis, die zeigen, wie das Living-Styleguide-System UI-Probleme löst:

Problem 1: Die Bezeichnung von Buttons unterscheidet sich von einem Systembereich zum anderen. Auf dem einen Button steht „Senden“, auf dem anderen Button mit gleicher Funktion steht „Speichern“.

Lösung: Im Living-Styleguide-System wird der Button zentral abgelegt und von dort aus automatisch in allen Bereichen des Software-Produkts genutzt.

Problem 2: Die Interaktionsmuster unterscheiden sich. Wird etwa in einem Bereich ein automatisches Speichern durchgeführt, ist im nächsten Bereich eine explizite Benutzeraktion wie ein Button-Klick notwendig.

Lösung: Im Living-Styleguide-System werden auch die Funktionen des Buttons zentral abgelegt und von dort aus in allen Bereichen des Software-Produkts genutzt.

Problem 3: Die Suchmechanismen in Suchfeldern sehen identisch aus, verhalten sich aber unterschiedlich. Das eine Eingabefeld für die Suche arbeitet mit Volltextsuche, das andere ist lediglich ein Filter.

Lösung: Im Living-Styleguide-System können Suchfunktion und Suchelemente einheitlich bereitgestellt werden.

Durch die Nutzung eines Living-Styleguide-Systems kann dem Risiko inkonsistenter Benutzeroberflächen also wirkungsvoll entgegengesteuert werden. Design-Entscheidungen, die im Entwicklungsprozess getroffen wurden, finden sich im Living Styleguide zentral wieder. Qualitätsgeprüfte UI-Komponenten, Patterns und Designs können damit fest in der Software verankert werden. Bei richtigem Einsatz profitieren die Endbenutzer von einem erprobten anwenderfreundlichen Design, das ihre Ziele und Bedürfnisse adäquat berücksichtigt.

Zusammenfassung

Der Living Styleguide kann noch so effizient wartbar, der Quellcode noch so sauber sein, wenn die UI-Elemente des Produkts für den Endanwender keinen Mehrwert liefern, ist das Software-Produkt langfristig zum Scheitern verurteilt. Deshalb sollten die Bedürfnisse und Ziele der Endbenutzer schon bei der Entwicklung des Living-Styleguide-Systems die Grundlage für alle weiteren Aktivitäten bilden.



Abbildung 1: Der Endbenutzer als relevanter Akteur des Living-Styleguide-Systems

Entwickler und Entwicklungsteams

Als Entwickler eines Software-Produkts bezeichnen wir jene Personengruppe, die das Produkt technisch implementiert. In modernen Arbeitsumgebungen mit modernen Arbeitsprozessen erfolgt die Entwicklung von Software-Produkten mit Hilfe von agilen Frameworks und Methoden wie Scrum oder Kanban. [3][4]

Warum stellen Entwickler eine relevante Benutzergruppe für das Living-Styleguide-System dar?

Entwickler „verwandeln“ die Design-Konzepte und Modelle mithilfe von Quellcode in ein lauffähiges Software-Produkt. Und am Quellcode hängt am Ende auch die Qualität und der Wert des gesamten Software-Produkts.

Über den Living Styleguide können Entwickler auf standardisierte UI-Komponenten, Patterns und Designs zugreifen. Die mühsame Implementierung von Designs entfällt und die Entwickler können sich auf die Lieferung ihrer Produktinkremente konzentrieren.

Welche Ziele verfolgen Entwickler?

Im Folgenden finden Sie typische Ziele und Bedürfnisse von Entwicklern, die häufig in Gesprächen, Retrospektiven, Reviews zutage kommen. Natürlich kann es abhängig von der jeweiligen Person noch andere Ziele geben.

- Hochwertige Software entwickeln mit modernen und innovativen Frameworks und Technologien.
- Qualitativ hochwertigen Quellcode schreiben, Komplexität reduzieren und skalierbare Architekturen entwickeln.
- Das Design bereits definiert vorfinden. Entwickler möchten sich auf das Coding konzentrieren und von tiefergehenden Designfragen verschont bleiben.
- Eine hohe Arbeitsgeschwindigkeit, denn ein einsatzfähiges Stück Software oder ein Feature wird einem fest definierten Zeitraum erwartet, zum Beispiel in bis zu vier Wochen. Dafür möchte sich der Entwickler auf seine Kernaufgabe und die Qualität des Quellcodes konzentrieren. Alles, was die Auslieferung des Produktinkrements verzögert, kommt ungelegen.
- Mithilfe von frühzeitigem Wissen über die Zielgruppe das Produkt verfeinern.
- Mithilfe von Experimenten und Testing den Code stetig optimieren und das Software-Produkt immer besser machen.

Was hindert Software-Entwickler, ihre Ziele zu erreichen?

Wie immer gibt es Stolpersteine, Hürden und Blocker, die Software-Entwickler hindern, ihre Ziele zu erreichen. Wäre es zu einfach, wäre es ja auch langweilig. Diese Hürden können sein:

- Unsicherheit bei Design-Entscheidungen. Wie soll das UI aussehen? Wie soll sich die Applikation verhalten?
- Bei Design-Fragen fühlen sie sich schnell überfordert und nicht verantwortlich.
- Kritik von Nutzern an den implementierten Designs empfinden sie als sehr unangenehm.
- Mehrdeutigkeiten bei einfachen Design-Guides oder visuellen Spezifikationen, die beispielsweise in Powerpoint oder Word dokumentiert sind, irritieren sie. Und wie schnell ist der Punkt erreicht, wo einfache visuelle Spezifikationen nicht mehr eindeutig zuordbar sind!
- Ein sehr hoher Kommunikationsaufwand mit UX-Designern, z. B. lange Diskussionen über UI-Elemente, nervt sie. Software-Entwickler möchten primär ein Produkt ausliefern und sich nicht in Design-Diskussionen und im Entscheidungsprozess verlieren.

Welche Vorteile bietet das Living-Styleguide-System für Entwickler?

Der Living Styleguide unterstützt Entwickler in vielen Punkten, die ihre Arbeit wesentlich erleichtern:

- Sie können sich zu 100 Prozent darauf verlassen, dass die UI-Komponenten, die sie für ein Software-Produkt verwenden, auf ihre Qualität geprüft wurden und effizient nutzbar sind.
- Der Living Styleguide stellt die technische Implementierung, also den Quellcode der UI-Komponenten zur Verfügung und legt eindeutig den Kontext fest, in dem die UI-Komponenten genutzt werden dürfen.

Zusammenfassung

Entwickler und Entwicklungsteams stellen neben den Endbenutzern die zweite wichtige Benutzergruppe eines Living Styleguides dar. Das Living-Styleguide-System macht die Entwicklung insgesamt leichter, weil der langwierige Prozess der Design-Entscheidungsfindung mit seiner Hilfe optimiert werden kann. Auf diese Weise können sich Entwickler auf ihre Kernkompetenzen konzentrieren und schneller und zuverlässiger neue Features für die Nutzer entwickeln

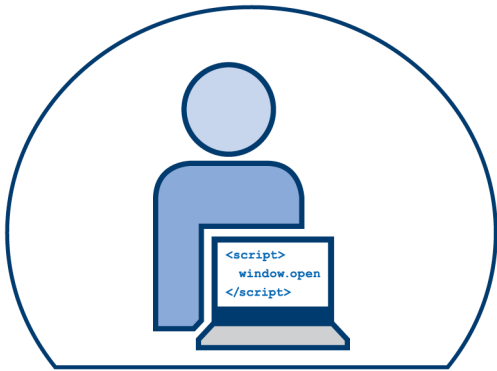


Abbildung 2: Software-Entwickler als relevante Akteure des Living Styleguide Systems

UX-Designer

Was verstehen wir unter UX-Designern und wie nutzen sie den Living Styleguide?

UX-Designer sind Spezialisten für die Gestaltung von Produkten, die dem Anwender eine positive Benutzererfahrung liefern. Zu den Schwerpunkten ihrer Arbeit gehören:

- **User Research:** Hier liegt die wichtigste Quelle für Daten und Erkenntnisse über die Endbenutzer. Auf der Grundlage von User Research kann der Designer fundierte Entscheidungen für die Gestaltung eines Software-Produkts treffen.
- **Iterative Prozesse:** UX-Designer folgen bei ihrer Arbeit einem iterativen Prozess, der folgende Komponenten umfasst:
 - User Research
 - Design (z. B. in Form von Sketches, Prototypen und Wireframes)
 - Build (z. B. in Form von technisch implementierten Prototypen oder Minimum Viable Products (MVP) [6] oder Erweiterungen bzw. Anpassungen bestehender Produkte).
- **Testen:** Das Testen der Designs ist essentiell und bildet die Grundlage für die nächsten Iterationen. Die Tests erfolgen mit echten Endbenutzern, um valide Daten und Erkenntnisse zu gewinnen. Diese Erkenntnisse nutzen UX-Designer, um ein Software-Produkt weiter zu verfeinern.

Warum bilden UX-Designer eine relevante Benutzergruppe für den Living Styleguide?

UX-Designer legen die Grundsteine für den Living Styleguide. Designs und UI-Elemente, die sie entwickelt und mithilfe von User Research und Testing validiert haben, werden im Living Styleguide aufgenommen. Erst sie erwecken den Living Styleguide zum Leben. UX-Designer verfeinern die Elemente des Living-Styleguide-Systems stetig in einem iterativen Prozess. Diese Elemente greifen Entwickler anschließend auf und setzen sie in Quellcode um.

Welche Ziele verfolgen UX-Designer?

UX-Designer möchten die Endbenutzer verstehen, sie in die Designentwicklung einbeziehen und Designs mit ihrer Hilfe validieren, um die UX kontinuierlich zu neuer „Exzellenz“ zu führen. Sie überprüfen das Design dafür vor allem auf diese vier Kriterien hin [5]:

- **Value:** Ist es nützlich?
- **Usability:** Ist es einfach zu handhaben?
- **Adoptability:** Ist die Nutzung von Beginn an einfach?
- **Desirability:** Macht es Spaß?

Was hindert UX-Designer, ihre Ziele zu erreichen?

Hürden, die UX-Designer in ihrer täglichen Arbeit behindern, erleben sie beispielsweise, wenn

- die in Code verwandelten Designs von den visuellen Designs abweichen. Oder platt ausgedrückt: „Wenn die Software-Entwickler nicht tun, was die Designer wollen.“
- zwischen UX-Designern und Software-Entwicklern unterschiedliche Erwartungshaltungen zu Konflikten führen, die in der täglichen Arbeit aufeinanderprallen.
- sich der Workflow aufwendig gestaltet und es nur langsam weitergeht, zum Beispiel infolge ineffizienter Design-Diskussionen oder Missverständnissen aufgrund von Mehrdeutigkeiten im Design.

Welche Vorteile hat das Living-Styleguide-System für UX-Designer?

Der Living Styleguide unterstützt Designer bei ihrer Zusammenarbeit mit Entwicklern sowie bei ihrer domainspezifischen Design-Arbeit. Typische Leistungen, mit denen ein Living-Styleguide-System Erleichterung für UX-Designer bewirken kann, sind:

- **Designelemente für das Prototyping:** Der Living Styleguide stellt Designern wiederverwendbare und sofort einsetzbare Komponenten zur Verfügung. Sie verfügen damit über eine qualitätsgeprüfte zentrale Sammlung an Design-Elementen für das Prototyping.
- **Gemeinsames Verständnis:** Der Living Styleguide erzeugt eine gemeinsame Sprache, die Mehrdeutigkeiten und daraus entstehenden Konflikte verhindert. Das fördert eine reibungslose Kommunikation und Zusammenarbeit.
- **Prozessoptimierung:** Der Aufwand für Prozesse und Workflows ist wesentlich schlanker. Design und Coding rücken viel näher zusammen als sonst üblich. Die zentrale Verankerung der Design-Elemente im Living-Styleguide sorgt für eine Synchronisierung von Design und Entwicklung.

Zusammenfassung

UX-Designer sind neben Endbenutzern und Software-Entwicklern eine weitere wichtige Benutzergruppe des Living-Styleguide-Systems. Der Living Styleguide kann die Kommunikation zwischen verschiedenen Benutzergruppen stark vereinfachen und UX-Design-Prozesse beschleunigen.



Abbildung 3: UX-Designer als relevanter Akteur des Living-Styleguide-Systems

Mehr als nur ein technisches Werkzeug

Der Living Styleguide bietet allen Benutzergruppen Vorteile und Erleichterungen für den täglichen Umgang mit dem Software-Produkt. Dabei ist er mehr als nur ein technisches Werkzeug: Das Living-Styleguide-System ermöglicht einen Brückenschlag zwischen den einzelnen Benutzergruppen und bündelt ihren Fokus auf den Endbenutzer. Als Ergebnis gewinnt das Software-Produkt deutlich an positiver UX für den Endbenutzer.

Das Zusammenspiel der Akteure

Wie funktioniert das Zusammenspiel der drei wichtigsten Akteure – Endbenutzer, Software-Entwickler und UX-Designer – im Kontext eines Living Styleguides?

Ein Living-Styleguide-System kann man sich wie eine übergeordnete Klammer vorstellen, die die Akteure und ihre Tätigkeiten zusammenhält und dadurch einen besonderen Mehrwert generiert. Ein konkretes Beispiel aus der Praxis:

Ein Softwareanbieter, nennen wir ihn *Sunshine AG*, beabsichtigt eine Vielzahl an neuen Produkten für seine Geschäftspartner zu entwickeln und verfolgt dabei vier wichtige Ziele:

- Erreichung einer konsistenten UX über alle Produkte hinweg
- Wiederverwendbarkeit von Design-Elementen
- Schnelles und effizientes Design sowie die Entwicklung neuer Produkte und Features
- Frühzeitige Validierung am Markt durch echte Endbenutzer und Kunden

In den folgenden Abschnitten wird beschrieben, wie der Living Styleguide bei der *Sunshine AG* zu einer zentralen Komponente im Zusammenspiel zwischen den Akteuren wird. In verschiedenen Szenarien betrachten wir den Kontext der einzelnen Akteure, und zeigen, wie sie vom Living-Styleguide-System profitieren.

Kontext der Endbenutzer

Der Endbenutzer steht im Zentrum des Zusammenspiels. Um ihn herum baut sich alles Weitere auf: die geschäftlichen Ziele, die Lösungsideen, die Features, die Art der Zusammenarbeit der Produktteams, die technische Architektur, die die Idee Wirklichkeit werden lässt. Ein Beispiel:

Markus ist Mitarbeiter eines Service-Centers und nutzt das Software-Produkt *Service24*, um seine tägliche Arbeit zu verrichten. Ein erfolgreicher Arbeitstag tritt für ihn ein, wenn er zehn Service-Anfragen positiv lösen konnte. Den Living Styleguide nutzt Markus indirekt über die Design-Komponenten von *Service24*. Über die User Research wird Markus in die Gestaltung des Designs mit eingebunden und die UX-Designer berücksichtigen seine Anforderungen in der nächsten Verfeinerung der Design-Elemente von *Service24*. Diese Design-Elemente finden wiederum Eingang in das Living-Styleguide-System.

Kontext der Software-Entwickler

Ein Entwicklerteam, bestehend aus Tim, Florian und Simone, programmiert das Software-Produkt *Service24* iterativ (z. B. in Scrum Sprints [3]) und stellt einzelne Features und Software-Inkrementen bereit, die den Mehrwert für Markus erhöhen.

Die Gestaltung der Benutzeroberfläche basiert dabei auf den Vorgaben der UX-Designer. Über ein Prototyping Tool wie „Invision“ [7] erhalten Entwickler die wichtigsten Design-Informationen für ihre Umsetzung.

Über die Komponentenbibliothek des Living-Styleguide-Systems können die Entwickler schnell und effizient qualitativ hochwertige und getestete UI-Komponenten in ihre Produkte integrieren. Auf diese Weise können sich Entwickler auf die Programmierung von Features konzentrieren, statt sich mit der Entwicklung von UI-Komponenten herumzuschlagen.

Kontext der UX-Designer

UX-Designer „gestalten“ Software-Produkte. Sie entwerfen Modelle, entwickeln Prototypen, MVPs, Mockups, Wireframes und viele weitere Design-Artefakte, die dazu dienen, Lösungsideen und Hypothesen am Nutzer zu validieren.

Um Design-Artefakte für das Software-Produkt *Service24* schnell und effizient zu erstellen, greifen die UX-Designer Tobias und Mila auf bestehende Design-Elemente aus dem Living Styleguide zurück. Sie nutzen dabei Tools wie Sketch oder Adobe Experience Design, in denen die qualitätsgeprüften Design-Elemente des Living Styleguides verfügbar sind und für das Erstellen von Prototypen genutzt werden können. Dabei ist wichtig, dass die Design-Elemente und der programmierte Living Styleguide synchron gehen mit der rein visuellen Instanz des Living Styleguides in Tools wie Sketch oder Adobe Experience Design.

Das Zusammenspiel der Akteure

Das Living-Styleguide-System wird von den Akteuren auf unterschiedliche Weise genutzt. Beginnend mit den Bedürfnissen und Zielen der Endbenutzer finden die Oberflächen ihren Weg von den Designs über den Code bis zum Endbenutzer. Zentrale Quelle für die qualitätsgeprüften Designs ist immer der Living Styleguide. Im Idealfall bekommt der Endbenutzer keine Design-Elemente zu Gesicht, die nicht im Living Styleguide ihren Ursprung haben.

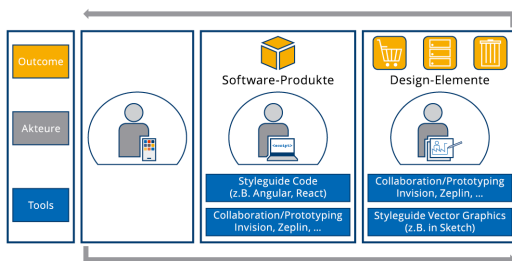


Abbildung 4: Die drei Hauptakteure bei der Nutzung des Living-Styleguide-Systems. Endbenutzer benutzen den Living Styleguide indirekt über das Produkt. Entwickler nutzen vorgefertigte UI-Komponenten zum Bauen ihrer Produktinkremente. UX-Designer nutzen visuelle Design-Elemente aus dem Styleguide zum beschleunigten Erstellen und Testen von Prototypen und MVPs.

Methodisches Vorgehen

Um den im letzten Abschnitt beschriebenen Ablauf erfolgreich zu etablieren, ist ein methodisches Vorgehen erforderlich. Das Ziel: Unsere drei Hauptakteure „Endbenutzer“, „Entwickler“ und „UX-Designer“ sollen bei ihren täglich zu bewältigenden Aufgaben eine spürbare Erleichterung erleben.

Um dieses Ziel zu erreichen, ist es hilfreich, den Living Styleguide wie ein Software-Produkt zu behandeln. Wir bedienen uns hierzu einer Methode aus der *Innovativen Softwareentwicklung*.

Etwas verkürzt dargestellt umfasst dieses Vorgehen folgende Arbeitsschritte:

1. Analyse:

- Welche Vision und welche Ziele verfolgt das Projektteam?
- Wer sind die relevanten Benutzergruppen des Projekts?
- Welche konkreten Bedürfnisse, Ziele und Anforderungen haben die Benutzergruppen?

2. Erstellen von Hypothesen für das Living-Styleguide-System und die Benutzergruppen

3. Validierung der Hypothesen durch Prototypen oder MVPs mit den jeweiligen Benutzergruppen

4. Integrieren der relevanten Design-Elemente in den Living Styleguide

5. Kontinuierliche Verbesserung des Living-Styleguide-Systems

Entwicklung mit Continuous X

Continuous X beschreibt die iterative Herangehensweise bei der Entwicklung eines Living-Styleguide-Systems. Die erfolgreiche Entwicklung hängt wesentlich von der effizienten Zusammenarbeit des Teams ab. Im Team werden dafür diese Rollen benötigt:

- **Product Owner:** kümmern sich um die Erhebung von Anforderungen, Priorisierung des Backlogs und Testen von Hypothesen.
- **UX-Designer:** Sie entscheiden, welche Design-Elemente in den Living Styleguide kommen.
- **Software-Entwickler:** Sie sind für die technische Implementierung zuständig.

Nur ein gut eingespieltes interdisziplinäres Team, das Bereiche wie UX-Design, Softwareentwicklung und Unternehmensstrategie miteinbezieht, ist in der Lage, ein Living-Styleguide-System zu entwickeln, das den Mehrwert liefert, den die beteiligten Akteure erwarten. Die Validierung erfolgt im optimalen Fall nicht nur am Endbenutzer, sondern auch an spezifischen Erfolgsfaktoren, die Management und Fachabteilungen vorgeben.

Für die Validierung werden Hypothesen aufgestellt, d. h. im Vorfeld werden bestimmte Annahmen formuliert, die kontinuierlich überprüft und verfeinert werden.

Dieses Vorgehen nennen wir Continuous X. Das X steht für die Menge aller formulierten Hypothesen. In die kontinuierliche und frühzeitige Validierung werden alle relevanten Zielgruppen miteinbezogen, also Endbenutzer, Software-Entwickler, UX-Designer, Unternehmensmanagement usw. Erweisen sich die Hypothesen als richtig, geht es von hier aus weiter.

Agile Frameworks zur Produktentwicklung wie Scrum [3][4] sind hier sehr hilfreich und bieten bewährte Prozesse und Methoden.

Wir empfehlen, das klassische Scrum um Komponenten aus Design und Lean UX zu erweitern, um den Fokus stärker auf die Validierung von Prototypen und MVPs durch echte Benutzer zu richten. Lean-Ansätze und -Methoden wie Lean Startup oder Lean UX stellen hier Rahmenwerke zur Verfügung, an denen sich die Teams orientieren können.

Der größte Vorteil von Continuous X liegt darin, dass sich die Verantwortlichen von dem Druck lösen können, zukünftige Entwicklungsprozesse vorhersagen zu müssen und auf dieser Basis Versprechen abzugeben, die das Team am Ende nicht einhalten kann. Das zyklische Vorgehen von Continuous X, das sich selbst kontinuierlich überprüft, schafft eine fundierte Basis für die nächsten Schritte bei der Entwicklung des Living Styleguides. Dieser Prozess erfolgt iterativ und transparent und bezieht alle Beteiligten mit ein, erfordert aber auch ein entsprechendes Mindset.

Continuous X im Detail

Um das Konzept des Continuous X besser zu verstehen, lohnt es sich, das Vorgehen einmal ganz praktisch Schritt für Schritt nachzuvollziehen:

Schritt 1: Continuous Research – Die Anforderungen der Benutzer eruieren

In diesem Schritt erarbeiten Software-Entwickler bzw. UX-Designer mit benutzerzentrierten Methoden die Bedürfnisse, Ziele und Anforderungen der relevanten Benutzergruppen des Living Styleguides. Das Team lernt kontinuierlich über die Nutzer.

Als primäre Benutzergruppe gelten Software-Entwickler, die sekundäre Gruppe bilden UX-Designer und die tertiäre Benutzergruppe stellen die Endbenutzer des finalen Software-Produkts dar. Für die Entwicklung des Living-Styleguide-Systems bedeutet dies, dass wir die Software-Entwickler zwar als wichtigste Zielgruppe einstufen, die Gruppe der UX-Designer und Endbenutzer jedoch ebenso berücksichtigen müssen, um ein System zu entwickeln, das für alle Benutzergruppen einen Mehrwert liefert.

So wäre es beispielsweise denkbar, dass der Living Styleguide für Software-Entwickler auf technischer Ebene einen tollen Mehrwert liefert, aber die UX-Designer einige Funktionalitäten des Living Styleguides nicht richtig nutzen können. Ein solches Manko würde das gesamte System zum Scheitern bringen.

Leitfragen für die Zielgruppe der Software-Entwickler könnten sein:

- Mit welchen Techniken schreiben Software-Entwickler den Code?
- Welche Frameworks nutzen sie?
- Wie wird die Software ausgeliefert?
- Welche automatisierten Tests nutzen sie?

Das gleiche Vorgehen empfiehlt sich für die zweite relevante Benutzergruppe des Living Styleguides: UX-Designer.

Leitfragen für UX-Designer könnten sein:

- Welche Tools nutzen sie für das visuelle Design?
- Welche Prototyping-Plattformen nutzen sie?
- Wie validieren sie die Prototypen mit den Benutzern?
- Wie diskutieren sie ihre Design-Ideen im Team?
- Wie treffen sie Design-Entscheidungen?

Die Antworten werden in Interviews mit den Benutzergruppen erarbeitet. Aus ihnen können dann konkrete Anforderungen an das Living-Styleguide-System abgeleitet werden.

Die Anforderungen werden in ein Backlog übertragen und dort von den Gestaltern des Living Styleguides für die Priorisierung, die Entwicklung und das Testen der Hypothesen genutzt. Ziel ist zunächst ein MVP des Living Styleguide als Testobjekt für die weitere Entwicklung.

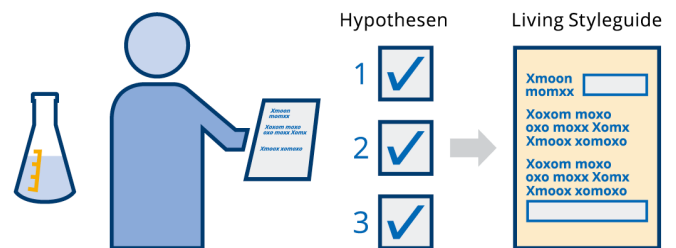


Abbildung 5: Testen von Hypothesen für ein Design-Element

Schritt 2: Minimum Viable Product – Ein Styleguide MVP erstellen und Hypothesen testen

In diesem Schritt testet das Living Styleguide Team die Hypothesen, die dem MVP zu Grunde liegen. Das könnte z. B. so aussehen:

Im Zuge der Continuous Research in Schritt 1 hat das Team erfahren, dass die Software-Entwickler Angular-Komponenten nutzen, um ein Bestellformular, das als Design-Element vorliegt, zu entwickeln. Die technische Integration funktioniert damit sehr gut. Die Software-Entwickler lehnen es hingegen ab, in zu viele konkret designbezogene Diskussionen und Entscheidungsfindungen einbezogen werden. Zum einen, weil solche Diskussionen ihrer Erfahrung nach zu Zeitproblemen führen und die Einhaltung von gesetzten Teilzielen gefährden können. Und zum zweiten fühlen sie sich durch solche Diskussionen eher überfordert und verunsichert.

Aus den in den Research-Interviews gesammelten Daten und Erkenntnissen lassen sich Annahmen treffen und Hypothesen modellieren. Die Methodenbaukasten des Lean UX Canvas [8] stellt Vorlagen für eine solche Hypothesenbildung bereit.

Zum Beispiel diese:

“We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature].”

In den eckigen Klammern befinden sich Variablen, für die jeweiligen Erkenntnisse aus der Continuous Research. Die Ergebnismerte für diese Variablen sehen in unserem Beispiel folgendermaßen aus:

- [business outcome] = die Dauer der Entwicklung eines Features bis zum Release reduziert sich um die Hälfte
- [user] = Software-Entwickler
- [benefit] = weniger Zeitaufwand für Design-Diskussionen
- [feature] = eine zentrale Sammlung an qualitätsgeprüften und technisch nutzbaren Design-Elementen für die Entwicklung von neuen Features

Ins Deutsche übertragen und gebrauchsfertig formuliert, ergibt sich daraus am Ende diese Hypothese:

„Wir glauben dass sich die Dauer der Entwicklung eines Features bis zum Release um die Hälfte reduziert, wenn wir erreichen, dass Software-Entwickler weniger Zeit für Design-Diskussionen aufwenden müssen, weil sie von einer zentralen Sammlung an qualitätsgeprüften und technisch nutzbaren Design-Elementen für die Entwicklung von neuen Features profitieren.“

Diese Hypothese lässt sich nun an der Zielgruppe der Software-Entwickler testen. In der alltäglichen Praxis werden eine Vielzahl von Hypothesen modelliert und entworfen. Im Anschluss werden die Hypothesen priorisiert, zum Beispiel mithilfe einer einfachen Matrix [9], und die Reihenfolge der Tests festgelegt.

Die anschließenden Tests lassen Erkenntnisse zu, die unmittelbar in die nächste Iteration einfließen können. Durch das kontinuierliche Testen und Verfeinern des Continuous X nähern wir uns einer idealen Lösung an.

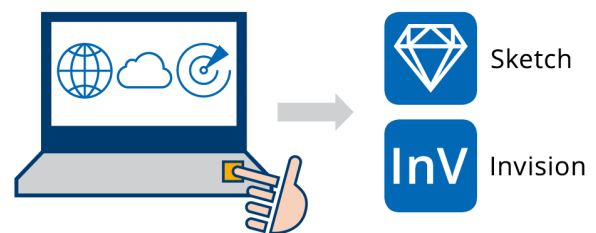


Abbildung 6 Integration der Design-Elemente in Sketch und Invision

Schritt 3: Continuous X: Kontinuierliches Verfeinern des Living Styleguides

Nach einigen Iterationen skaliert das Living-Styleguide-System. Die Kernfunktionalität liefert den erwarteten Mehrwert. Immer mehr Design-Elemente finden Eingang in den Living Styleguide. Die Elemente sind qualitätsgeprüft und validiert. Die Verfeinerung verläuft kontinuierlich in den folgenden Schritten:

- UX-Designer testen kontinuierlich die UI-Komponenten anhand ihrer Hypothesen in realen Arbeitsumgebungen, unter Berücksichtigung von Vorgaben und Richtlinien des Unternehmens wie Corporate Identity.
- UX-Designer stellen die relevanten Designs auf allen Plattformen zur Verfügung:
 - Visuelle Design Tools wie Sketch oder Adobe Experience Design
 - Prototyping- und Collaboration-Plattformen wie Invision oder Zeplin
 - Komponenten-Libraries in den relevanten UI-Frameworks und Bibliotheken wie Angular oder React
- Software-Entwickler verwandeln die Design-Elemente in Quellcode. Die programmierten UI-Komponenten stehen nun bereit, um von Entwicklern in Software-Produkten und Features getestet zu werden.
- In Reviews mit Business-Stakeholdern werden Zielsetzungen überprüft.
- Ziele und Produktvision werden unter Einbezug der Business-Strategie und den kontinuierlich gewonnen Erkenntnissen aus User-Research-Aktivitäten entwickelt.

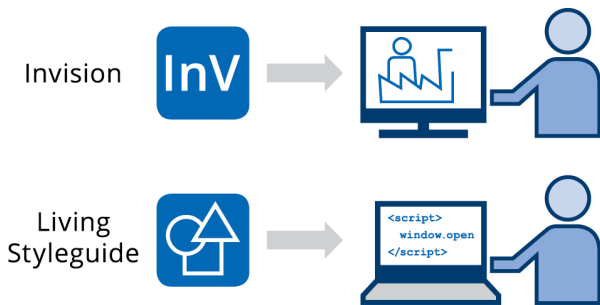


Abbildung 7: Living-Styleguide-Entwickler entwickeln UI-Komponenten, Software-Entwickler nutzen den Living Styleguide

Zusammenfassung und Fazit

Ein Living-Styleguide-System ist mehr als ein einfaches Werkzeug zur Dokumentation von Design-Elementen. Zum einen liefert es seinen Benutzern zahlreiche Mehrwerte wie

- Verstärkung der positiven Benutzererfahrung mit dem Produkt selbst,
- einfachere und effizientere Abläufe der Softwareentwicklung und des UX-Designs.

Zum anderen unterstützt es die Teamarbeit und die Kollaboration der Beteiligten.

Im Idealfall entwickelt ein interdisziplinäres Team aus Software-Entwicklern, UX-Designern und Business-Stakeholdern kontinuierlich Verbesserungen am Living-Styleguide-System. Auf diese Weise bleibt das System transparent, dauerhaft verfügbar und nutzbar. Es verfeinert sich immer weiter durch kontinuierliche Validierungen mit den Benutzern und wird zum zentralen Ort und Qualitätsgaranten für herausragende User Experience.

Für die gemeinschaftliche Entwicklung komplexer und stark veränderlicher Software-Produkte ist ein System wie der Living Styleguide unverzichtbar.

Referenzen

- [1] User Research: Baxter, Kathy; Courage, Catherine; Caine, Kelly. Understanding Your Users: A Practical Guide to User Research Methods (Interactive Technologies) (S.38). Elsevier Science. Kindle-Version.
- [2] Usability: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [3] Scrum: <https://www.scrumguides.org/>
- [4] Scrum (Diskussion) <https://thecattlecrew.net/2019/02/13/scrum-vs-kanban-eine-entscheidungshilfe/>
- [5] User Experience (UX): https://www.uxmatters.com/authors/archives/2009/10/frank_guo.php
- [6] Minimum Viable Produkt, kurz MVP: https://en.wikipedia.org/wiki/Minimum_viable_product
- [7] Invision: <http://invisionapp.com/>
- [8] Lean UX: <https://www.jeffgothelf.com/blog/leanuxcanvas/>
- [9] Priorisierungsmatrix: <https://mural.co/framework/importance-difficulty-matrix>