

Wenn Ideen Sex haben

Wie kann ich wilde Kundenanforderungen einfangen?

Maik Pfingsten, Der Troubleshooter

In meiner aktiven Zeit als Troubleshooter habe ich über 10 Jahre lang internationale Entwicklungsprojekte gerettet. Das Ziel war stets, ein Projekt in den nächsten sechs Monaten wieder auf die Gleise zu stellen und zum Erfolg zu führen. Dafür habe ich zunächst das Team wieder aufgebaut und die Lasten an das zu entwickelnde System geprüft. Schließlich brauchte ich für eine sinnvolle Strategie, die in den folgenden, harten Verhandlungen mit den Stakeholdern notwendig war, Klarheit über die Anforderungen.

In den meisten Projekten gab es bis zu meinem Eintreffen im Grunde kein Lastenheft, weder vom Hersteller noch vom Zulieferer bzw. Entwicklungsdienstleister. Entsprechend hantierten alle mit irgendwelchen PowerPoint-Folien, Mails und sonstigen Dokumenten, nicht aber mit einem wirklichen Lastenheft und einer Spezifikation.

In diesen Situationen war es für mich entscheidend, etwas in der Hand zu haben, mit dem ich schnell ein umfassendes Bild der Anforderungen an das System erschaffen konnte. Basierend darauf konnte ich erst entscheiden, was in den nächsten Monaten realistisch und sinnvoll erreichbar war. So entstand der System Footprint. Um es mit den Worten eines Komponentenverantwortlichen von Daimler zu sagen: „Das ist endlich mal ein Meta-Bild unseres System mit seinen Anforderungen.“

Von der Mindmap zur Version 4.0: Die Entwicklung des System Footprint

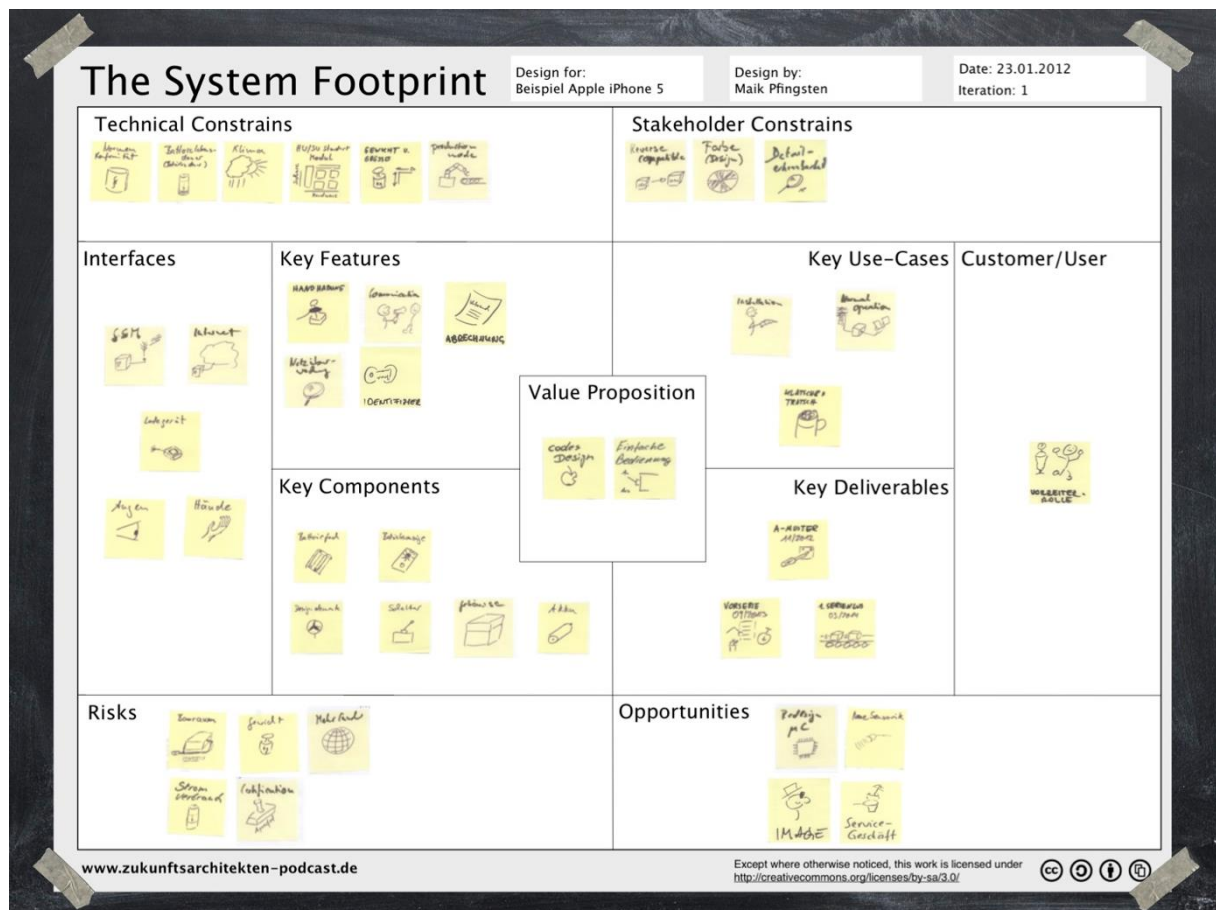
2005 hat die Entwicklung des System Footprints mit einer Mindmap begonnen. Wenn ich als Troubleshooter in ein Projekt eingestiegen bin, habe ich anhand dieser von mir aufgebauten Mindmap die verschiedenen Anforderungen an das System zusammengetragen. Grundsätzlich funktionierte das Ganze auch gut, war aber in den Workshops noch nicht optimal einsetzbar. 2007 bin ich dann über die Visualisierung mithilfe einer „Canvas“ und zahlreicher Post-Its gestolpert und habe die Mindmap in die zweite Generation des System Footprints übertragen. Mit dieser Version ließ sich in den Workshops bedeutend besser arbeiten.

Durch intensive Diskussionen mit der Hörer-Community meines Podcasts entstand dann die dritte Generation, mit der ich bis 2015 erfolgreich unterwegs war. Im gleichen Jahr habe ich mir schließlich die Zeit genommen, meine Erfahrungen aus den über hundert moderierten Workshops in die vierte Generation des System Footprints einfließen zu lassen.

Alles im Blick: Aufbau & Inhalte des System Footprints

Wenn wir ein Lastenheft erstellen und darin Anforderungen beschreiben, ist es an einem bestimmten Punkt essentiell zu entscheiden, ob eine Anforderung für das System sinnvoll ist oder nicht. Diese Entscheidungen können wir einfacher fällen, wenn wir den Kernnutzen des Systems und die wesentlichen Bestandteile im Blick haben. Dies lässt sich auf einer „Canvas“ als System Footprint sehr effektiv visualisieren: Auf einer großen, weißen Fläche, die in verschiedene Bereiche eingeteilt ist, können Teile z.B. im Rahmen eines Workshops mit Inhalten gefüllt werden.

Als Beispiel habe ich einen System Footprint für ein Smartphone aus meinen Seminarunterlagen dargestellt.



Die beim System Footprint vorhandenen Bereiche – wie auch in dem Beispiel zu sehen – sind die folgenden:

Der Benutzer

Die Frage, wer eigentlich der Benutzer ist, scheint trivial – oft ist sie es aber nicht. Bei einem Kaffeemaschinenhersteller beispielsweise sind Händler die Kunden – oder doch die Nutzer der Kaffeemaschine? Ist für einen Fahrkartenhersteller die Bahngesellschaft der Kunde? Oder der Fahrgast, der das System benutzt, um eine Fahrkarte zu kaufen? Ist in der Autobranche der Kunde des Zulieferers immer der Autohersteller? Der wirkliche Benutzer ist doch schließlich der Autofahrer, er benutzt das System und entscheidet, ob er das Auto kauft.

Das Produkt wird für den Endkunden entwickelt, also für die Käufer des Produkts bzw. die Nutzer des Systems – und das ist entsprechend auch die zu fokussierende Zielgruppe.

Der Kernnutzen unseres Produkts

Der Kernnutzen einer Kaffeemaschine ist nicht immer nur das Herstellen eines Kaffees: Es gibt beispielsweise ein paar entscheidende Gründe, warum Kunden eine Premium-Kaffeemaschine kaufen, nämlich Qualität, Status, Design etc. Das ist das Nutzenversprechen des Produkts und führt zu einer sehr emotionalen Kaufentscheidung, die auf keinen Fall zerstört werden darf.

Nutzungsfälle

Mit den Anwendungsfällen wird beschrieben, wie der Kunde das System benutzt. Die Idee kommt aus dem SysML, dort wird von „Use-Cases“ gesprochen. Das Wissen um die Nutzungsfälle hilft uns, später einen Anforderungen zu definieren, die das Nutzenversprechen in jedem dieser Fälle erfüllt.

Liefereinheiten

Wenn wir ein Produkt ausliefern, haben wir verschiedene Varianten und auch verschiedene Musterstände, die wir in einem Entwicklungsprojekt berücksichtigen müssen. Bei dem Beispiel Kaffeemaschine könnten das die Varianten „Premium Classic“, „Premium Gold“ und „Premium Future“ sein. Alle drei sind Kaffeemaschinen, die auf dem gleichen Systemdesign basieren, aber unterschiedliche funktionale Ausprägungen haben.

Kernkomponenten

Ein System wird heutzutage selten aus dem Nichts entwickelt und ist in sich absolut neu. Vielmehr arbeiten Firmen bewusst oder unbewusst mit einem Baukastensystem, in dem man als Systemingenieur immer wieder eine grundlegende System-Architektur definieren kann. Da man hierbei getestete Komponenten verwendet, werden Zeit und Risiko reduziert.

Kernfunktionen

Um mit unserem System einen Kernnutzen zu erzeugen, muss das Projekt die wichtigen Kernfunktionen kennen und beschreiben. Die Kernfunktionen können sich über eine oder mehrere Komponenten verteilen und sollten auch deshalb im System Footprint explizit betrachtet werden. Bei diesen Themen sind intensive Diskussionen mit den Entwicklern von großem Vorteil, denn dadurch bekommen alle ein besseres Systemverständnis.

System-Schnittstellen

Wenn wir über Produkte reden, dann ist eine Systemabgrenzung wichtig. Nur so kann für alle Beteiligten klar festgelegt werden, über welches System geredet wird. Gleichzeitig sind Schnittstellen auch typischerweise die Stellen, an denen die meisten Fehler entstehen. An realisierten Komponenten haben die Entwickler selbst ein großes Interesse und möchten ein positives Ergebnis schaffen – die Schnittstellen zwischen Systemen jedoch bleiben meiner Erfahrung nach sehr oft schlecht abgestimmt.

Systeme können über ihre Schnittstellen Informationen, Energie oder Stoffe austauschen – Schnittstellen sind also sehr essentiell: Bei unserer Kaffeemaschine sind nicht nur der Stecker für das Netz und die Übertragung von Energie eine Schnittstelle. Wir haben auch die Schnittstelle zur Wasserleitung, über die der Stoff Wasser ausgetauscht wird. Und schließlich gibt es auch Schnittstellen zum Menschen: Bedientöpfe sind eine Schnittstelle in Form von Druckenergie, und auch das Display stellt eine Schnittstelle dar, an der Informationen ausgetauscht werden. Oft sind es genau diese Schnittstellen zum Menschen, die gerne vergessen werden.

Bestehende Vorgaben

Auch Vorgaben an das System müssen klar definiert sein. Das können beispielsweise Vorgaben zur Versorgungsspannung sein, doch auch Vorgaben zu Standards und Normen, Klimabedingungen, Gewicht, Größe, (Software-)Architektur, Haptik, Ergonomie, Verpackung usw. sollen hier beschrieben werden. In der Regel sind Constraints nicht-funktionale Anforderungen, die man weiter in technische Vorgaben und Stakeholder-Vorgaben aufteilen kann, um sie besser auseinanderzuhalten.

Visualisierung – und noch viel mehr: Der System Footprint und Lastenhefte

Der fertige System Footprint kann dazu dienen, ein Lastenheft zu schreiben. Doch Vorsicht – immer wieder habe ich an diesem Punkt folgende Aussagen gehört: „Vieles davon haben wir schon dokumentiert. Hier sind Word-Dokumente, PowerPoint-Folien, Meeting-Minutes etc., das kippen sie einfach da rein und alles ist fertig!“

Doch weit gefehlt. Meiner Erfahrung nach sind im Schnitt gerade mal 20% der System-Footprint-Post-Its bedient, wenn wir „das alles reinkippen“, die anderen 80% jedoch bleiben mehr oder weniger leer. Doch auch um das Lastenheft sinnvoll und effizient zu füllen, bietet sich der System Footprint an: Dank seiner Visualisierungen kann jeder schnell sehen, wo welche Anforderungen noch fehlen oder nicht klar beschrieben sind.

Wenn Sie mehr über Lastenhefte wissen wollen, schauen Sie einfach bei <http://lastenhefterstellen.de/> vorbei. Dort gebe ich Tipps und Tricks zur Erstellung von Lastenheften und schreibe über mein agiles Vorgehen.

Autor

Dipl.-Ing. Maik Pflingsten ist internationaler Speaker und Mentor mit dem Schwerpunkt Systems Engineering. Mit seinem Praxishintergrund von über 12 Jahren als Systemingenieur und Troubleshooter für komplexe Embedded Projekte hat er sich zu einem Vordenker im Systems Engineering entwickelt.

Er bloggt zum Thema Lastenhefte unter <http://lastenhefterstellen.de/> und podcastet zum Thema Systems Engineering und Leadership unter <http://zukunftsarchitekten-podcast.de/>.

