

Mehr Effizienz durch eine integrative Arbeitsweise

Ist-Zustand:

Mehrfachrealisierung der Projekte

In der normalen Projektabwicklung setzt jedes Gewerk das Projekt eigens um, damit alles für die Endproduktion vorbereitet werden kann.

Allerdings ist das jeweilige Produkt der Umsetzung oft nur im abstrakten Sinne für die nächste Produktionsstufe zu verwenden.



Ist-Zustand:
Ich mache mein Ding

Im Konzept:
Projektanforderungen werden analysiert und als Wireframes für die visuelle Realisierung zur Verfügung gestellt.

Im Design:
Wireframes werden in visuellen Elementen in Form von Styleguides, Layouts und Komponenten verwandelt.



Ist-Zustand:
Ich mache mein Ding

Im Frontend:

Designelementen werden prototypisch in HTML/CSS/JS gerendert.

Im Backend:

Das Frontend-Prototyp wird zerlegt und in das produktive Endsystem integriert.



Ist-Zustand:

Diese Arbeitsweise wird sowohl in Wasserfall- als auch agilen Methoden gelebt.

Nachteile:

Sehr wenig von den digitalen Erzeugnissen aus den Vorstufen werden tatsächlich produktiv verwendet.

Änderungen und Korrekturen am Projekt müssen oft in allen Produktionsstufen integriert werden.

Fazit:

Hier können kleine Besserungen gleich 3-fach wirken!



Soll-Zustand:

Jeder Fachbereich soll möglichst Verwertbares erstellen!

Aus dem Konzept:

Keine Wireframes, sondern HTML-Elemente.

Template Editoren und Code-Libraries sollen verwendet werden.

Vorteile:

- Man kann besser beurteilen, ob das Endergebnis läuft!
- Code-Elemente sind ggf. schon im FE verwertbar.



Soll-Zustand:

Jeder Fachbereich soll möglichst Verwertbares erstellen!

Aus dem Design:

Keine Screen-Layouts, sondern einzelne visuelle Elemente sollen produziert werden.

Vorteile:

Diese können schon dem Kunden als Richtwerte dienen. Abnahmegegenstand ist jedoch das FE-Prototyp! Somit werden die Erwartungen vom Kunden von lebendigen Dokumentation geleitet.



Soll-Zustand:

Jeder Fachbereich soll möglichst Verwertbares erstellen!

Aus dem Design:

Die visuelle CSS-Regeln, die zentral für das Projekt relevant sind, sollen schon für das Frontend vorbereitet werden.

Vorteile:

Durch den Einsatz von CSS-Compiler ist es möglich, die Variablen als Text zu erstellen für den direkten Einsatz in das Frontend-Buildsystem.



Soll-Zustand:

Jeder Fachbereich soll möglichst Verwertbares erstellen!

Als Beispiel:

Der Designer arbeitet direkt in die `_settings.scss` Datei hinein und stellt diese unter Versionskontrolle dem Projekt zur Verfügung.

Vorteile:

- Aufwand für Knowledge-Transfer sinkt.
- Änderungen fließen vom Kunden direkt ins BE.
- Mögliche Fehlerquellen werden ausgeschlossen.



Soll-Zustand:

Es hört nicht im Frontend auf...

Integrationsarbeit vom FE ins BE könnte auch verringert werden.

Kopiervorgänge können durch Buildsysteme minimiert werden. Beim Aufsetzen, bitte Das Backend mit berücksichtigen!

CI-Systeme sollen auch ggf. ans Frontend gekoppelt werden, um Datenbestände überschaubar zu halten.



Soll-Zustand:

Benennung der Elementen soll einheitlich sein.

Schon in der Konzeptionsphase sollen Seiten- und Templateelemente als Einheiten definiert und benannt werden.

Die Definitionen sollen dann für das gesamte Projekt gelten – oder noch besser, darüber hinaus.



Risiken und Stolperfallen: Das, was der Bauer nicht kennt...

Erwartet Widerstand! Besonders in der Konzeption ist mit Skepsis zu rechnen. Templating tools für Bootstrap, z.B. können (noch) nicht detailreich Informationen vermitteln.

Im Frontend ist oft es gefühlt schneller, Farb- und Layoutwerte aus zu lesen, als diese vom Designer geben zu lassen.

Hier gilt es zu betonen, dass man ihnen keine Programmierkenntnisse abverlangen wird!



Vielen Dank!

d.fitzpatrick@triplesensereply.de

