



POI

Lechmündung



POI

Schloss Oberndorf



POI

Augsburger Dom



LADESTATION

Ladestation 1

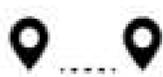


AKTUELLE ETAPPE

Marxheim bis Augsburg



VORHERIGE | NÄCHSTE



Etappen



Karte



Planer



AroundMe



Unterwegs



POI

Lechmündung



POI

Schloss Oberndorf



POI

Augsburger Dom



LADESTATION

Ladestation 1



AKTUELLE ETAPPE

Marxheim bis Augsburg



VORHERIGE | NÄCHSTE



Etappen



Karte



Planer



AroundMe



Unterwegs

gehen wie ein frontender

Welche Elemente sind Aufwandstreiber bei der Frontend-Entwicklung? Mit welchen Lösungen kann man schmerzhafteste Aufwände vermeiden? Und wie denkt ein Frontender generell bei der Analyse eines Screendesigns? Wie jeder sehr einfach mit Risikokarten eine grobe Risikoanalyse durchführen kann.



entwickler denke

30%

Sauber strukturierter Code

Abstraktion und der Einsatz von Frameworks

Qualitätssicherung und Versionskontrolle

Datenintegration und Deployments

Verzicht auf Portalentwickler dank JAM Stack



die welt ist komplexer geworden

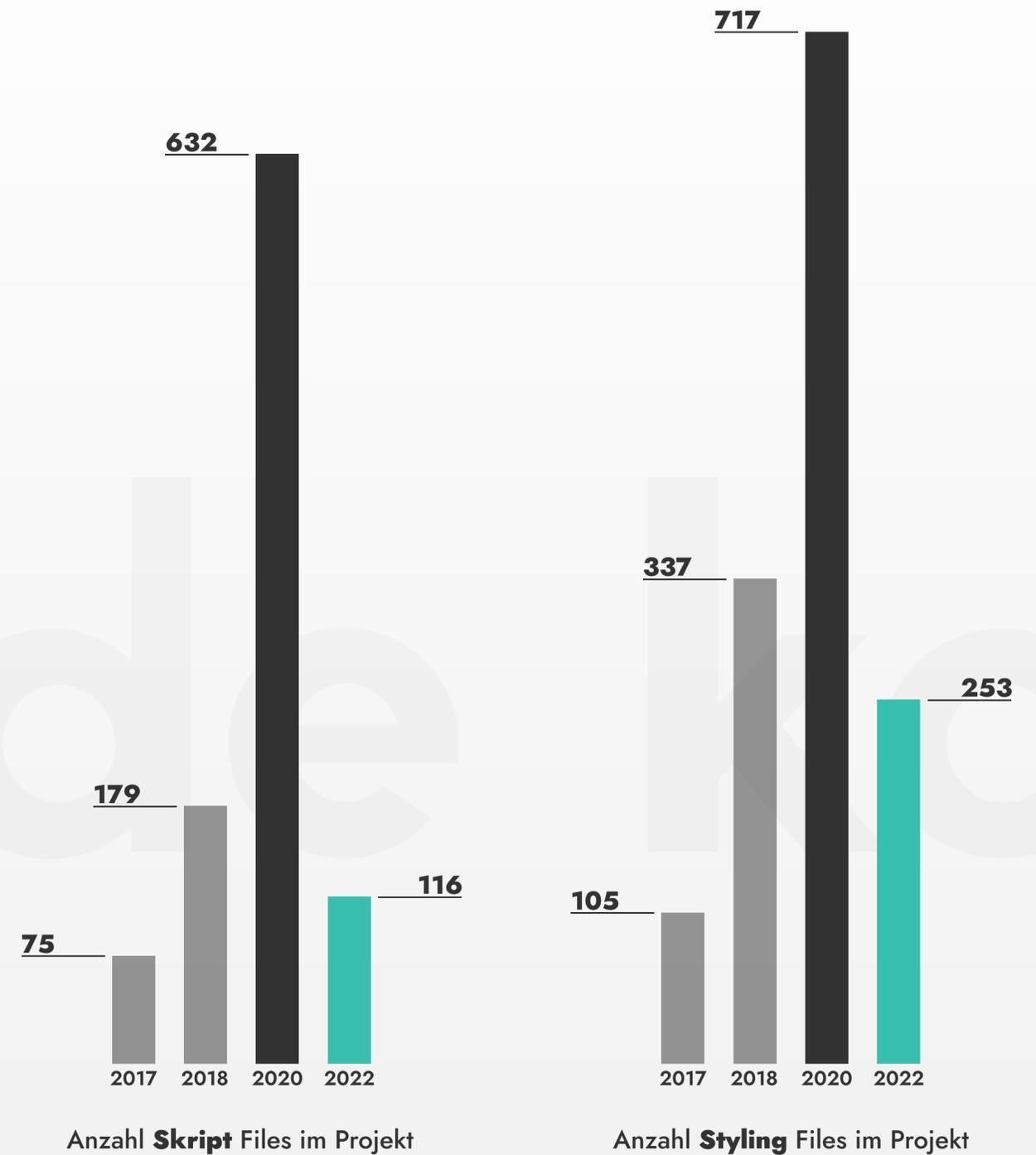
Ende der 2000er wurden statische HTML Dokumente in
kostenlosen Tools erstellt.
Sehr direkte aber auch langsame/fehleranfällige Vorgehensweise.

**Moderne Webprojekte sind zu komplex und langlebig für
diesen Ansatz.**



steigende komplexität in zahlen

Beinhane exponentielle Steigerung um den Exponenten 1,2.
Komplexität in 2020 gefährdet Wartbarkeit.
Paradigmenwechsel bei infomax nach 2020 um dem entgegen
zu wirken.
Komplexität dadurch in mehr Abstraktion verlagert.



was ist passiert?

abstraktion

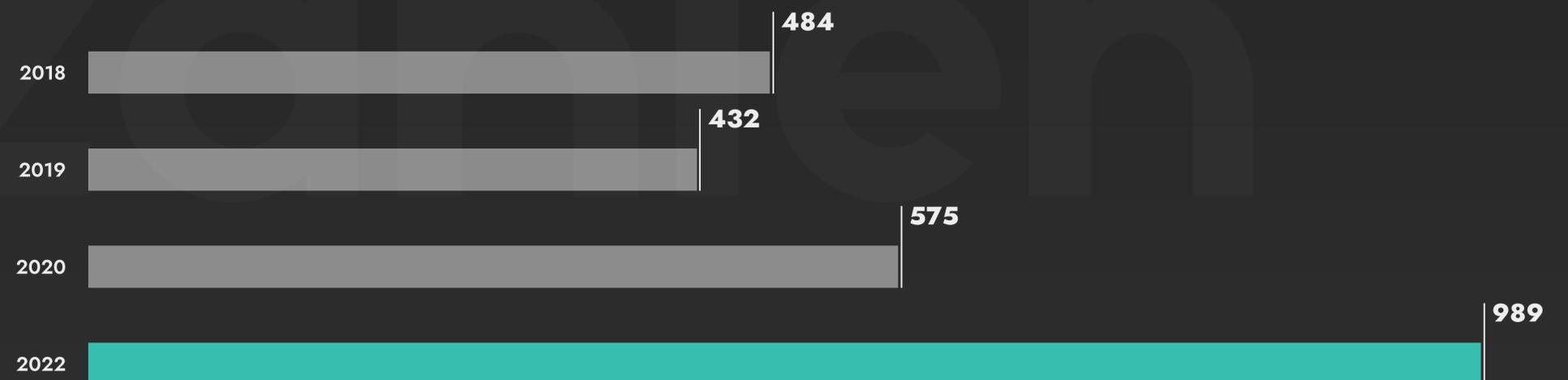
Einzelkomponenten statt Seiten
Dynamischer Seitenaufbau
Externe Komponenten (NPM)



steigende komplexität in zahlen

Annähernde Stagnation bis 2020.

Paradigmenwechsel nach 2020 verlagert Komplexität zugunsten effizienter
Wartung in die Verwendung moderner Frontend Frameworks.



Anzahl **NPM Module** Files im Projekt



designer auge

70%

Pixelgenaue Umsetzung von Designs

Interpretation dessen was in statischen Designs zu sehen ist

Definition von Haptik und Verhalten von Oberflächen

Abweichungen vom Design erkennen

Wo nötig oder sinnvoll vom Design abweichen

(korrekte Schriftgrößen!!!)



einfach mehr

mehr als nur code

Oberflächen müssen sauber nach Designvorgaben umgesetzt werden:

gutes Auge für Details (Schriftgrößen!!!)

ein sicheres Gefühl für Abstände, Maße und Verhältnisse

grundlegendes Wissen zur Bedienung von Designtools

Haptik und Verhalten von Oberflächen

Wie interagieren Menschen mit verschiedenen Eingabegeräten

Gefühl für Animation und Timing

Wissen über Browser und Endgeräte

Featuresupport über Browser und Endgeräte hinweg

Wie funktioniert ein Browser - wie kann ich performante Oberflächen bauen?

Entscheidungskompetenz bzgl. Abweichungen vom Design

An welchen Stellen macht es Sinn sich von Designvorgaben zu entfernen?

Grundlagenwissen zu Gestaltungsregeln



“ das brauchen wir nur auf smartphone ”

Unfertige Konzepte und Designs aus unterschiedlichen Gründen in die Frontend Entwicklung gegeben - können schnell zum Aufwandsgrab werden.



aufwandsgrab

unfertige konzepte

Einzigartige Elementkombinationen

Elemente sehen in bestimmten Situationen anders aus oder verhalten sich anders als sonst
häufig hat dies eine Neuentwicklung zur Folge um Abhängigkeiten zu vermeiden

Elemente werden nur für einen Anwendungsfall konzipiert

aufgrund von redaktioneller Pflege können diese Elemente dennoch oft auch anders genutzt werden
der Frontender muss daher alle möglichen Fälle vorsehen, in welchen ein Element vorkommen kann

späte Weiterentwicklung des Designs

eine gewisse Parallelität ist nicht zu vermeiden und oft sogar gewünscht werden bereits umgesetzte Elemente nachträglich im Design geändert - ist eine enge Abstimmung mit dem Frontender nötig um widersprüchliches Verhalten zu vermeiden



“ das element gibt es so ähnlich schon,
kann man wiederverwenden ”

Redundanzen gilt es in der Entwicklung üblicherweise zu vermeiden.

In der Frontend Entwicklung hingegen kann es sinnvoller sein wiederkehrende Komponenten mehrfach umzusetzen.



unnötige redundanz

nur mehr vom immer gleichen?

falsches Bedürfnis nach Verallgemeinerung
wiederverwendbare Komponenten auf Basis
vermeintlich komplexer Eigenschaften - Bsp. Slider

Wartung mit Seiteneffekten

Einsatz wiederverwendbarer Komponenten schafft
Abhängigkeiten zwischen allen Module, welche
diese Komponenten nutzen

Minimale Vorteile

Überdeckung in Aussehen, Features und Verhalten
jenseits dieser einen Eigenschaft meist minimal



redundanz

minimales aufwandsrisiko

Welche Elemente sind Aufwandstreiber bei der Frontend-Entwicklung? Wie jeder sehr einfach mit Risikokarten eine grobe Risikoanalyse durchführen kann.



2



SCHIZOPHREN

Das Element verändert sein Verhalten oder Aussehen, entweder in Abhängigkeit vom Bedienkontext (Touch/NoTouch, Bildschirmgröße) oder in Reaktion auf eine Benutzeraktion (Click, Scroll)

1



STICKY

Das Element befindet sich immer im sichtbaren Bereich. Alle anderen Elemente auf der Seite, halten den benötigten Platz frei.

2



HIDDEN STICKY

Das Element befindet sich bei Bedarf im sichtbaren Bereich, wird jedoch situationsabhängig ausgeblendet. Alle anderen Elemente auf der Seite, halten den benötigten Platz frei, aber nur wenn das Element zu sehen ist.

3



AREA STICKY

Das Element wird innerhalb eines bestimmten Scrollbereichs der Seite, im sichtbaren Bereich gehalten. Darüber und darunter bleibt das Element auf seiner Position.

1



SLIDER

Weil dieser Klassiker zum verstecken unerwünschten Inhalts immer gut funktioniert, haben wir hier eine Gruppe von Elementen welche in einer Slidergalerie angeordnet sind.

1



REDUNDANTE BEDIENUNG

Der Button mit dem großen Pfeil der nach Rechts zeigt wird nicht verstanden? Kein Problem, wir ergänzen noch Apple Dots, Autoslider und Wischgesten.



4

REIHENFOLGE? EGAL!

Eben noch auf Platz 1 dann wieder ganz woanders. In einer Gruppe von Elementen, ändern diese scheinbar beliebig ihre Positionen untereinander.

1

KLEINTEILIG

"Und hier brauchen wir noch einen Link - und in dem Link kann es noch ein Infolcon und ein fett gedrucktes Wort geben."

Ein Element besteht aus sehr vielen kleinen, individuell gestylten Texten, Links und mehr

3



SKYLINE

Gerade Kanten sind langweilig? Dieses Element wächst oben, unten, links und rechts über seine eigenen Grenzen hinaus.



2

BACKSTEINLAYOUT

Backsteine sind cool und das schon seit 10 Jahren. Deshalb werden Teaser, Bilder und weiteres in einer Masonry artigen Struktur angeordnet.

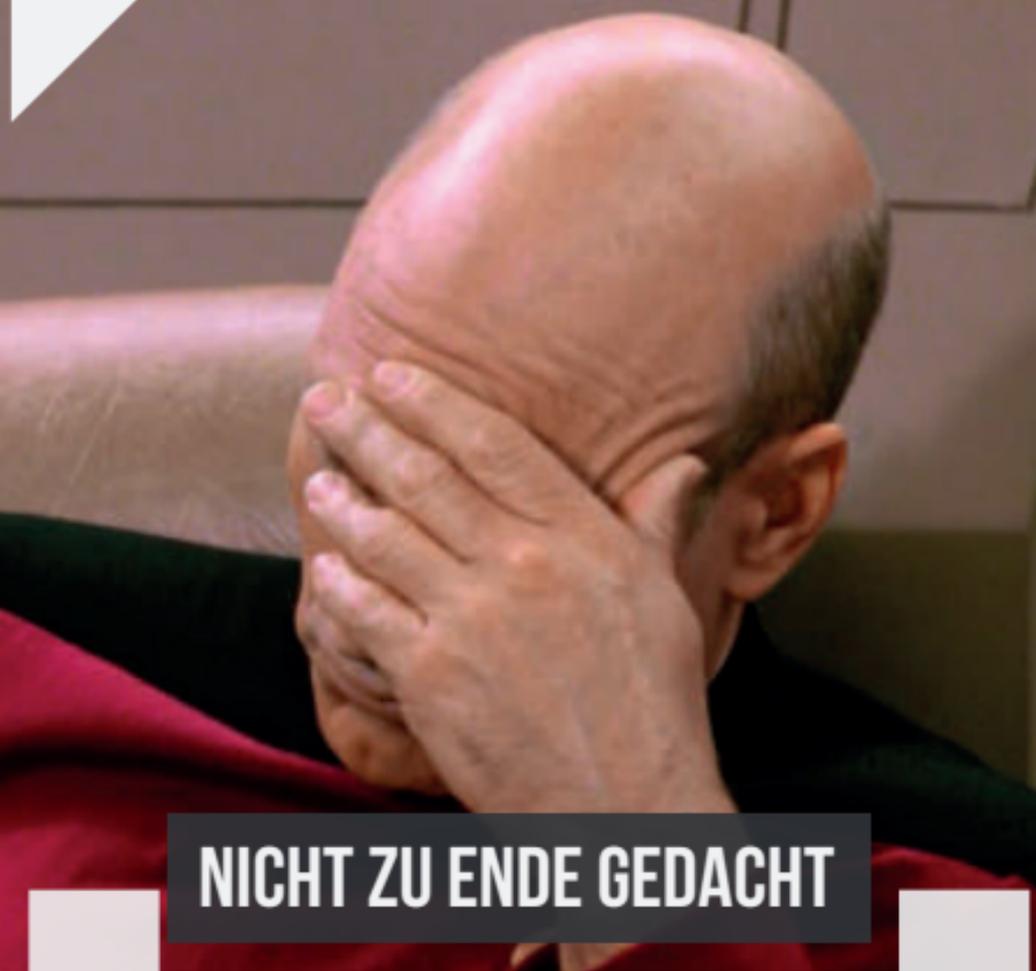
1



HIDE AND SEEK

Hier muss der User noch für seinen Content arbeiten wie ein Gorilla für seinen Fischkeks. Bei Klick auf einen Button hier, fällt da der bisher versteckte Inhalt raus.

5



NICHT ZU ENDE GEDACHT

Die Zeit war knapp? Das Meeting zu früh zu Ende? Diesen Fall hätte es gar nicht geben sollen oder wurde als unwichtig definiert?

Egal warum, aber dieses Element wurde nicht zu Ende konzipiert und das holen wir nun in der Frontend Entwicklung nach.