

UX Schweiz Last Thursday Talks

***Wenn Software den Menschen im Weg steht:
Produktivitäts-Hindernisse in
Unternehmenssoftware,
und was wir dagegen tun können***

Timo Würsch

25. August 2022

UX Architect & Gründer, Interaktionswerk GmbH

Produktivität

Produktivität





«Produktivität» im weiteren Sinne

Nicht (nur) Stück pro Stunde, sondern:

1. Unsere Zeit sinnvoll nutzen
2. «Meine Arbeit hat eine Auswirkung»
3. Repetitive oder mühsame Tätigkeiten minimieren
4. Software kann teuer sein. Deshalb sollte sich ihr Einsatz lohnen.

Finde ich alles wichtig.

Wieso das Thema?

- Produktivität ist Teil einer guten User Experience
 - Siehe z.B. *Flexibility and Efficiency of Use: The 7th Usability Heuristic Explained*
- Unternehmen = Produktivität! (Zumindest in der Theorie)
- Hat sich für mich so ergeben, aufgrund meiner Projekte
- Oft mein Eindruck: «Das kann nicht produktiv sein»

Wieso das Thema?

“despite ongoing digitalisation, labour productivity growth has declined sharply across OECD countries over the past decades”

Digitalisation and productivity: a story of complementarities, OECD Economic Outlook, Vol 2019 Issue 1

“Digitization contains the promise of significant, productivity-boosting opportunities—but the benefits have not yet materialized at scale.”

Is the Solow Paradox back? McKinsey Quarterly, June 4, 2018

“You can see the computer age everywhere but in the productivity statistics.”

Robert M. Solow: “We’d better watch out”, New York Times Book Review, July 12 1987

Methodik

Basis: Qualitative Nutzerforschung mit:

- Geübten Nutzern
- Meist regelmässige Nutzung
- Insgesamt ca. 50 Interviews & Arbeitsplatzbeobachtungen
 - Über mehrere Projekte & Systeme hinweg

- Disclaimers:
 1. Beobachtungen im Nachhinein aus der Produktivitäts-Perspektive ausgewertet
 2. Wünschenswert wäre mehr Genauigkeit

Systeme

Ich fokussiere mich hier auf:

- Issue-Tracking-System: Tickets, Zuständige, Workflows
- Inventarsystem: Geräte, Standorte, Lager
- CRM- (Customer Relationship Management-)-System

- Alles in einem Unternehmens-Kontext
 - (keine Consumer-Software)
- In der Regel Standardsoftware

Die wichtigsten Erkenntnisse



Photo by [Some Tale](#) on [Unsplash](#)

Für etwas, das ich mit der alten Software in fünf Minuten erledigt hatte, brauche ich jetzt eine halbe Stunde.

Sogar wenn Nutzer genügend Zeit zum Lernen hatten

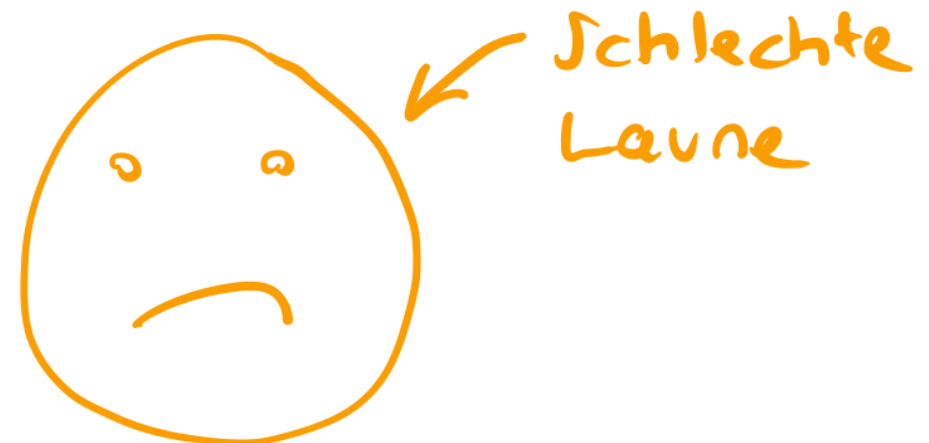
Wir werden jetzt agil. Wir werden das jeden Monat machen müssen, für jedes einzelne Projekt. Wird ein Overhead werden.

Bei der neuen Software vermischen wir Geschwindigkeit.

Oft beobachtet:

1. Überstrukturierung
2. Mangelnde Integration
3. Zu wenig Automatismen
4. Ungenügende Kollaborations-Funktionen
5. Unpassende Berechtigungen
6. Fehlende Massenoperationen

→ Drückt auf Produktivität und Laune.



1. Überstrukturierung

2. *Mangelnde Integration*

3. *Zu wenig Automatismen*

4. *Ungenügende Kollaborations-Funktionen*

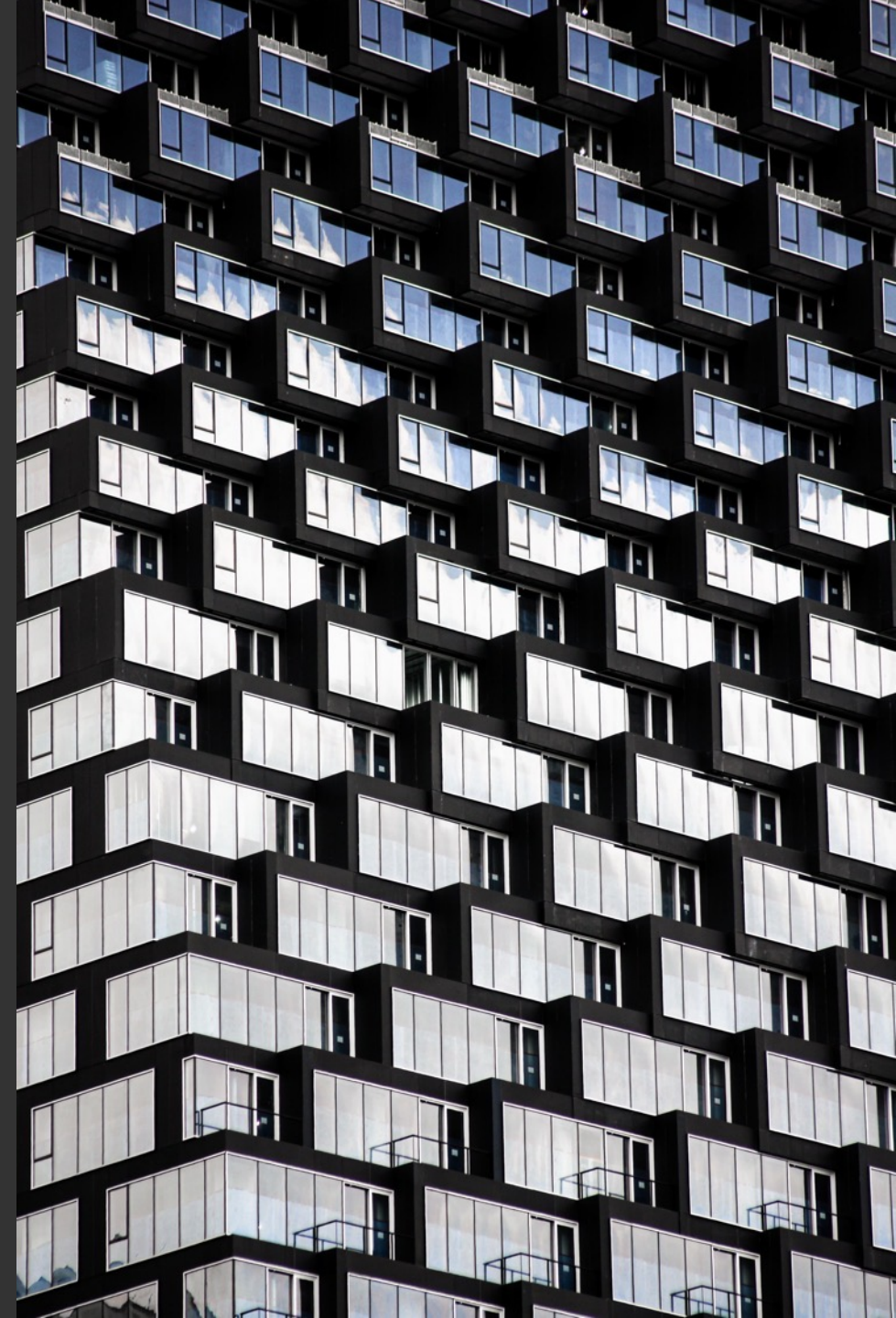
5. *Unpassende Berechtigungen*

6. *Fehlende Massenoperationen*

7. *Ausserdem...*

8. *Positives!*

9. *Verbesserungsmöglichkeiten*



Beispiel: Change ankündigen, via Issue Tracking



← Höhe des Fensters



Beispiel: Labels in einem Issue-Tracking-System

Labels

Review_Offen

Entwickler_4

Feature_2

Architekt_1

Budget_3

Abteilung

Design_Fertig

Tests_Offen

Stakeholder_1

Stakeholder_4

Umgebung_2

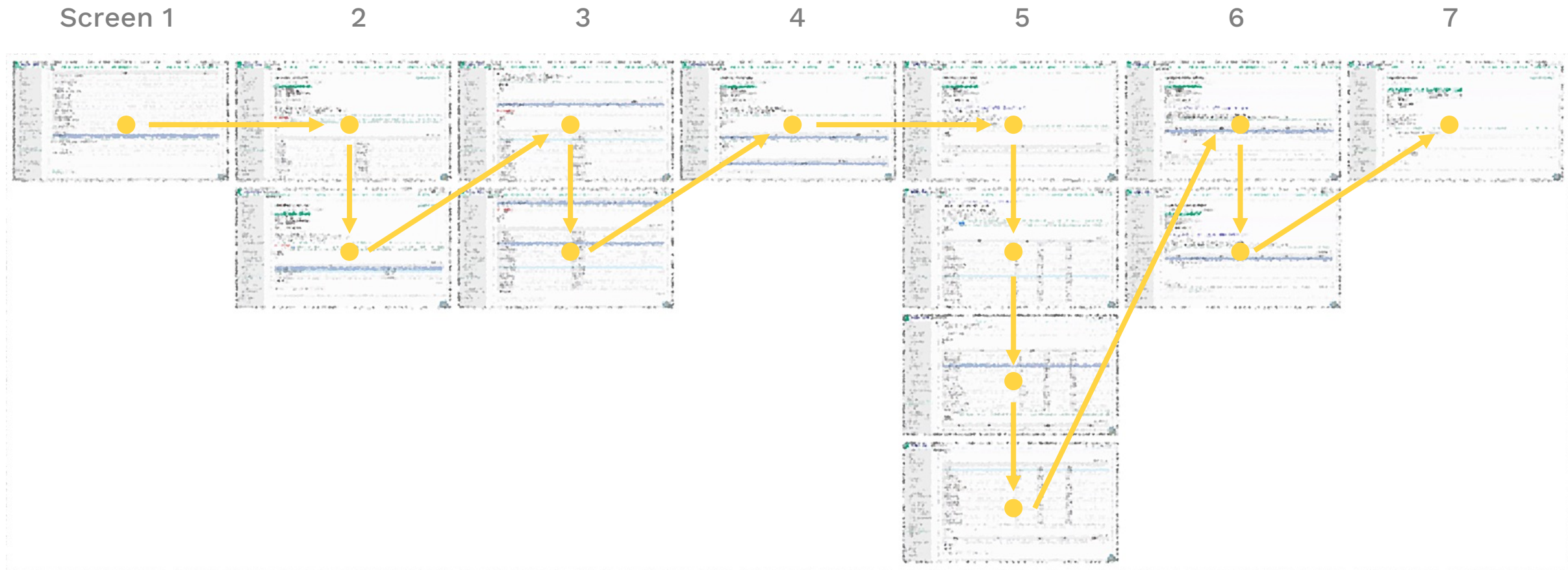
Produkt_2

Produkt_3

«Ich verbringe nicht selten Stunden damit, diese Labels zu vervollständigen.»

⚠ Viele dieser Labels bilden Dinge ab, die es in diesem System bereits gibt, z.B. Status und Zuständigkeiten.

Beispiel: Gerät von einem Lager in ein anderes verschieben



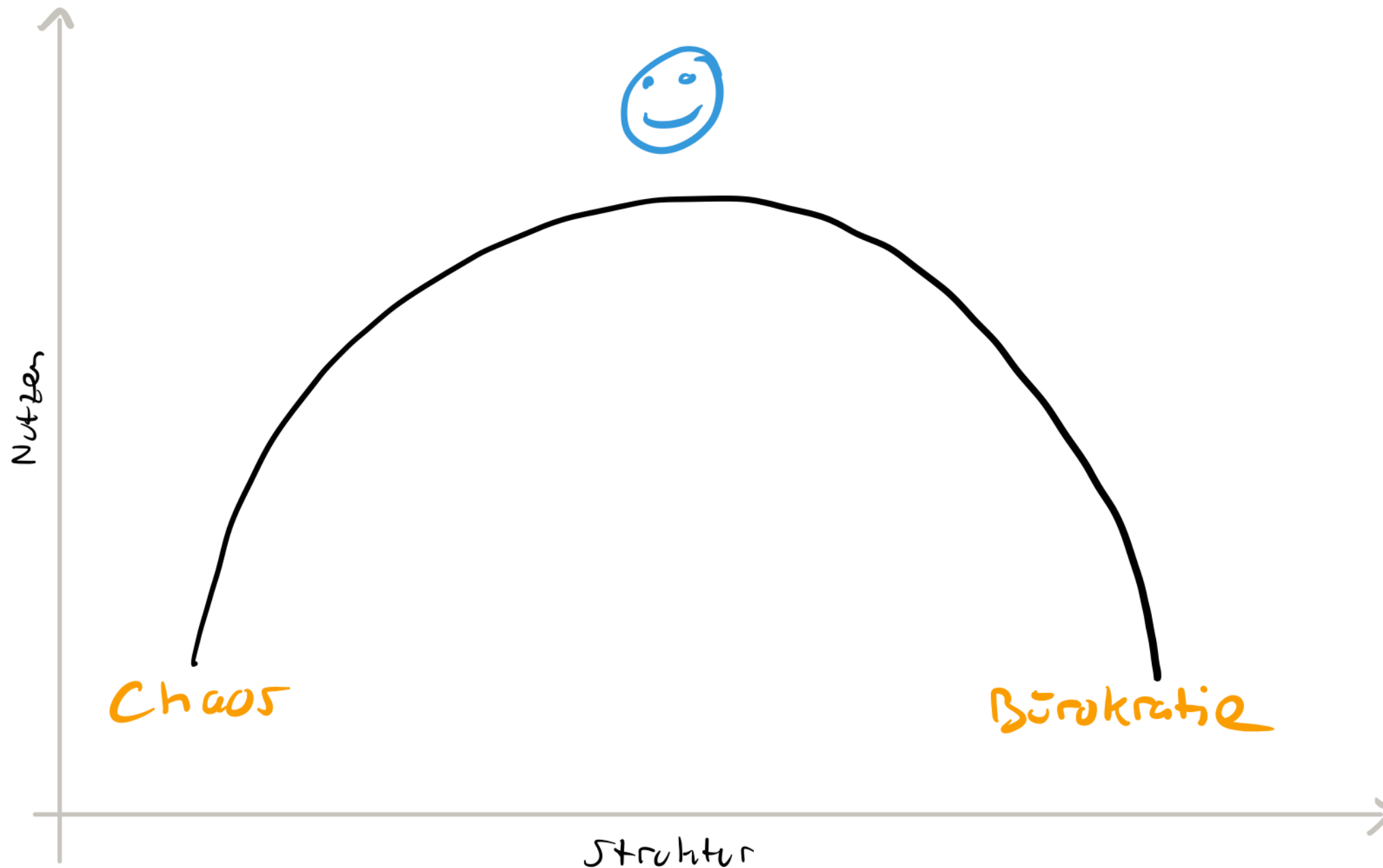
«Zu viele Klicks.»

⚠ Die Anzahl der Interaktionen ist bei häufiger Nutzung durchaus relevant.

Überstrukturierung

Symptome:

1. (Zu) viele Formularfelder
2. (Zu) viele Pflichtfelder
3. (Zu) starre Abläufe
4. (Zu) feingliedrige Kategorien und Hierarchien



Wie beurteilen wir Struktur? Durch Beobachtung der Nutzer.

«zu viel»

«zu starr»

«zu feingliedrig»

Schwierig zu messen.

Deshalb: Beobachten, welche Schwierigkeiten Nutzer haben, z.B.:

- Auffindbarkeit: Schnell, langsam, gar nicht
- Alibi-Eingaben, Workarounds (kosten ebenfalls Zeit & Nerven)

Überstrukturierung kommt von...

1. Wächst historisch
(Deshalb oft zu viel Struktur,
und nicht etwa zu wenig)
2. Wunsch nach fehlerloser,
kompletter Eingabe der Daten in
einer bestimmten Reihenfolge
(→ Validierung!)
3. Wunsch nach Auswertbarkeit
(Reporting)

⚠ Oft unrealistisch. Nutzer arbeiten nicht sequenziell, nicht unterbruchsfrei, und haben nicht immer alle Informationen bereit.

⚠ Häufig aufgrund Bedürfnis nach Optimierung, aber ohne Impact der "Messung" abzuschätzen

1. *Überstrukturierung*
2. **Mangelnde Integration**
3. *Zu wenig Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. *Positives!*
9. *Verbesserungsmöglichkeiten*

«Wir haben ein zweites Tool, das was ähnliches macht, aber da steht nicht immer dasselbe drin, deshalb muss ich rückfragen»

«User neues Tool arbeitet nicht mit dem Barcode-Scanner, deshalb nehmen wir einen Umweg über Excel»

Excel:

- Nicht zwingend unproduktiv
- Weist auf Nutzerbedürfnis hin

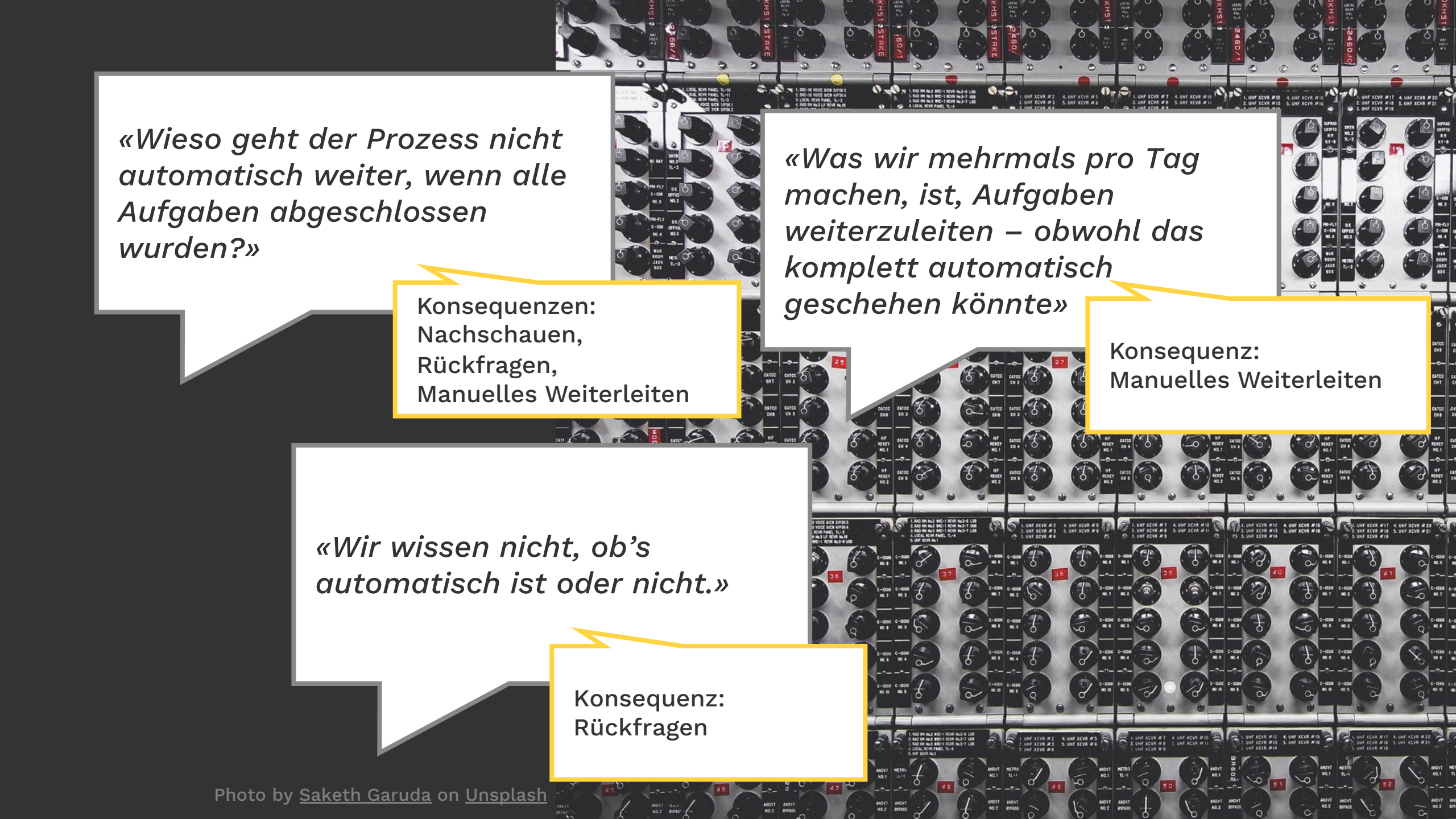
Anzeichen von mangelnder Integration

- Nutzer übertragen Daten von Hand zwischen Systemen
 - Abschreiben
 - Copy-Paste
 - Ausdrucken & Einscannen
- Rückfragen: «Was steht im anderen System drin?»
- → Führen alle zu mehr Arbeit als notwendig

Mangelnde Integration kommt von...

- Fehrender Machbarkeit
 - Technisch
 - Finanziell
- Bewussten Entscheiden
 - «Wenn das jemand mal händisch machen muss, ist das egal»
 - «Wir bauen eine Komplettlösung, da ist dann alles drin»
- Mangelnder Phantasie
 - Es muss nicht immer eine vollautomatische Softwareschnittstelle sein.

1. *Überstrukturierung*
2. *Mangelnde Integration*
- 3. *Mangelnde Automatismen***
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. *Positives!*
9. *Verbesserungsmöglichkeiten*



«Wieso geht der Prozess nicht automatisch weiter, wenn alle Aufgaben abgeschlossen wurden?»

Konsequenzen:
Nachschauen,
Rückfragen,
Manuelles Weiterleiten

«Was wir mehrmals pro Tag machen, ist, Aufgaben weiterzuleiten – obwohl das komplett automatisch geschehen könnte»

Konsequenz:
Manuelles Weiterleiten

«Wir wissen nicht, ob's automatisch ist oder nicht.»

Konsequenz:
Rückfragen

Anzeichen für mangelnde Automatismen

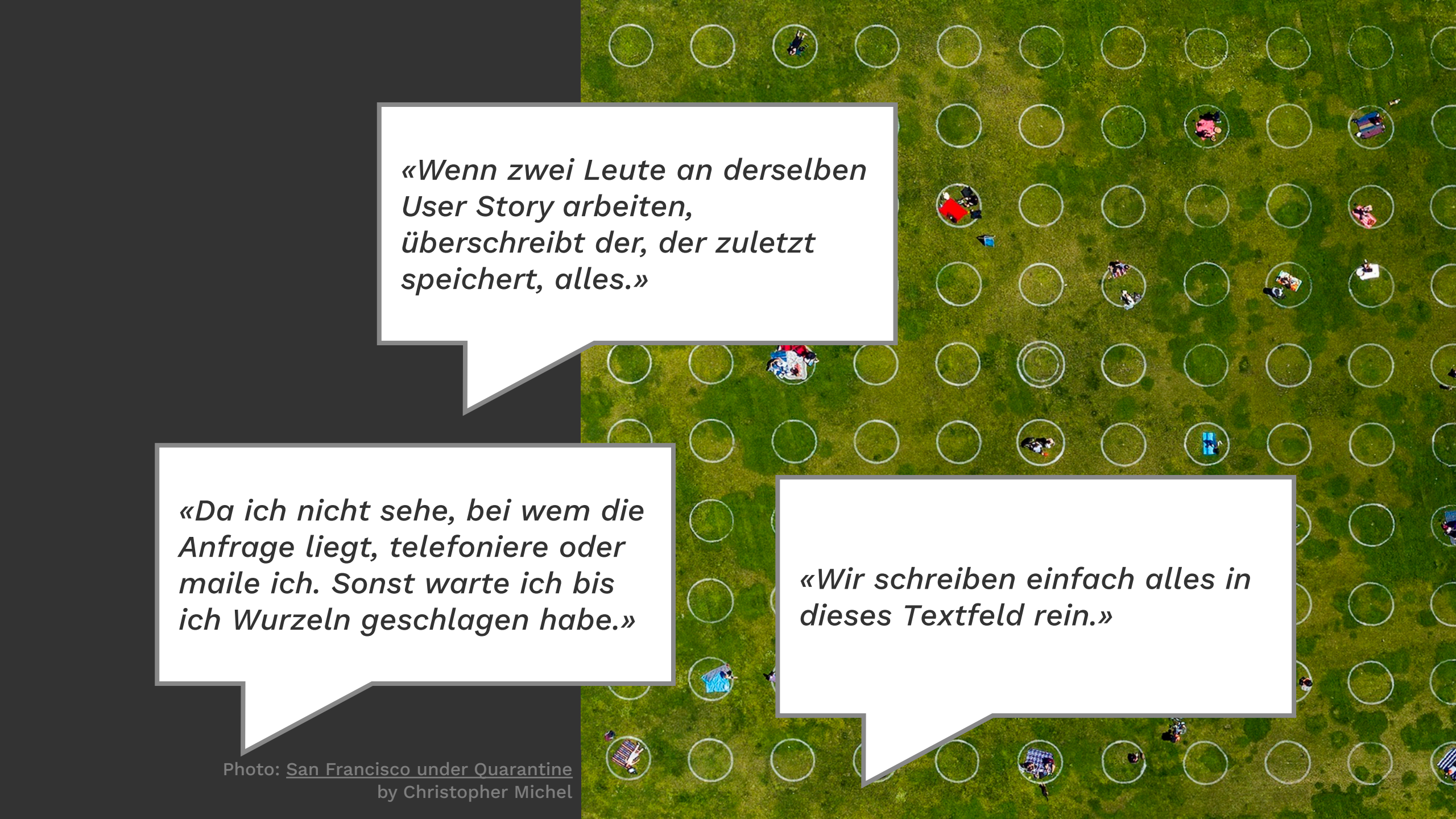
- Aufgaben / Arbeiten bleiben liegen
- Aufgaben benötigen länger als erwartet
- ... aber vor allem die zuvor genannten Aussagen.
- Das alles bekommt man nur, wenn man mit den Nutzern redet und sie beobachtet.

Mangelnde Automatismen kommen von...

Fehlendem Verständnis auf Design-/Entwicklungsseite für:

- Wo es Automatismen braucht und wo sie möglich sind
- Unter welchen Bedingungen ein Automatismus ausgelöst werden soll
- Dass ein Automatismus für den Nutzer nachvollziehbar sein muss

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
- 4. *Ungenügende Kollaborations-
Funktionen***
5. *Berechtigungen, die im Weg stehen*
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. *Positives!*
9. *Verbesserungsmöglichkeiten*



«Wenn zwei Leute an derselben User Story arbeiten, überschreibt der, der zuletzt speichert, alles.»

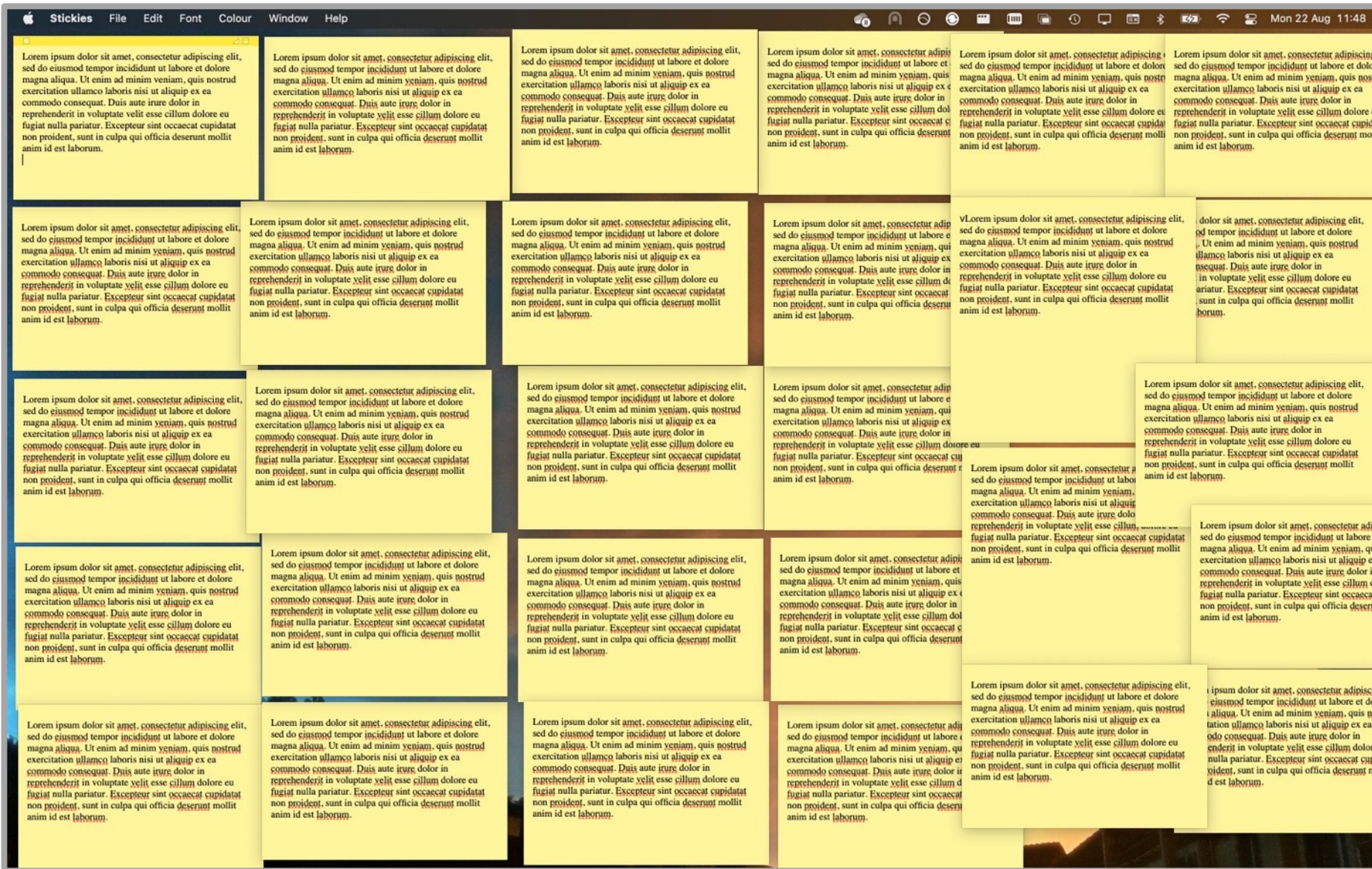
«Da ich nicht sehe, bei wem die Anfrage liegt, telefoniere oder maile ich. Sonst warte ich bis ich Wurzeln geschlagen habe.»

«Wir schreiben einfach alles in dieses Textfeld rein.»

Kollaborationsfunktionen

- Jede Art von arbeitsbezogener Kommunikation
- Sichtbarkeit von Personen und Zuständigkeiten
- Sichtbarkeit des Status
- Gleichzeitige Bearbeitung von Artefakten

Beispiel: Copy-Paste von Task-Beschreibungen als Notizen auf dem Desktop (Symbolbild)



«Details sind für mich nicht mehr sichtbar, nachdem ich den Task abgesendet habe. Ich kann den Task auch nicht zurücknehmen oder umleiten.»

«Das erlaubt mir, zu rekonstruieren, was ich ihnen gesendet habe.»

Anzeichen für ungenügende Kollaborations-Funktionen

- Kommunikation findet auf anderen Kanälen statt
- Kommunikation via Freitextfeld, das zweckentfremdet wird
- Häufige Rückfragen
- Gegenseitiges Warten
- Aufgaben nochmal von vorne beginnen



Kostet alles Zeit und Nerven.

Mangelnde Kollaborationsfunktionen kommen von...

- Fehrender Machbarkeit
- Fehlendem Bewusstsein, wie die Zusammenarbeit abläuft
 - Teilnehmer
 - Aufgaben, Ziele, ggf. Fristen
 - Workarounds und Abkürzungen

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
- 5. *Unpassende Berechtigungen***
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. *Positives!*
9. *Verbesserungsmöglichkeiten*



«Weil ich Admin bin, kann ich selbst nachschauen, aber die anderen sind blind unterwegs.»

«Wir sehen auch Dinge, mit denen wir gar nichts zu tun haben.»

«Wenn ich in den Ferien bin, kann niemand sonst daran weiterarbeiten.»

Anzeichen für unpassende Berechtigungen

- Informations-Flut: Benutzer sieht zu viel
- Fehlbedienungen
 - (scheinen mir allerdings selten)

Zu tolerante Berechtigungen

- Fehlende Übersicht: Benutzer sieht zu wenig
- Fehlende Eingriffsmöglichkeiten

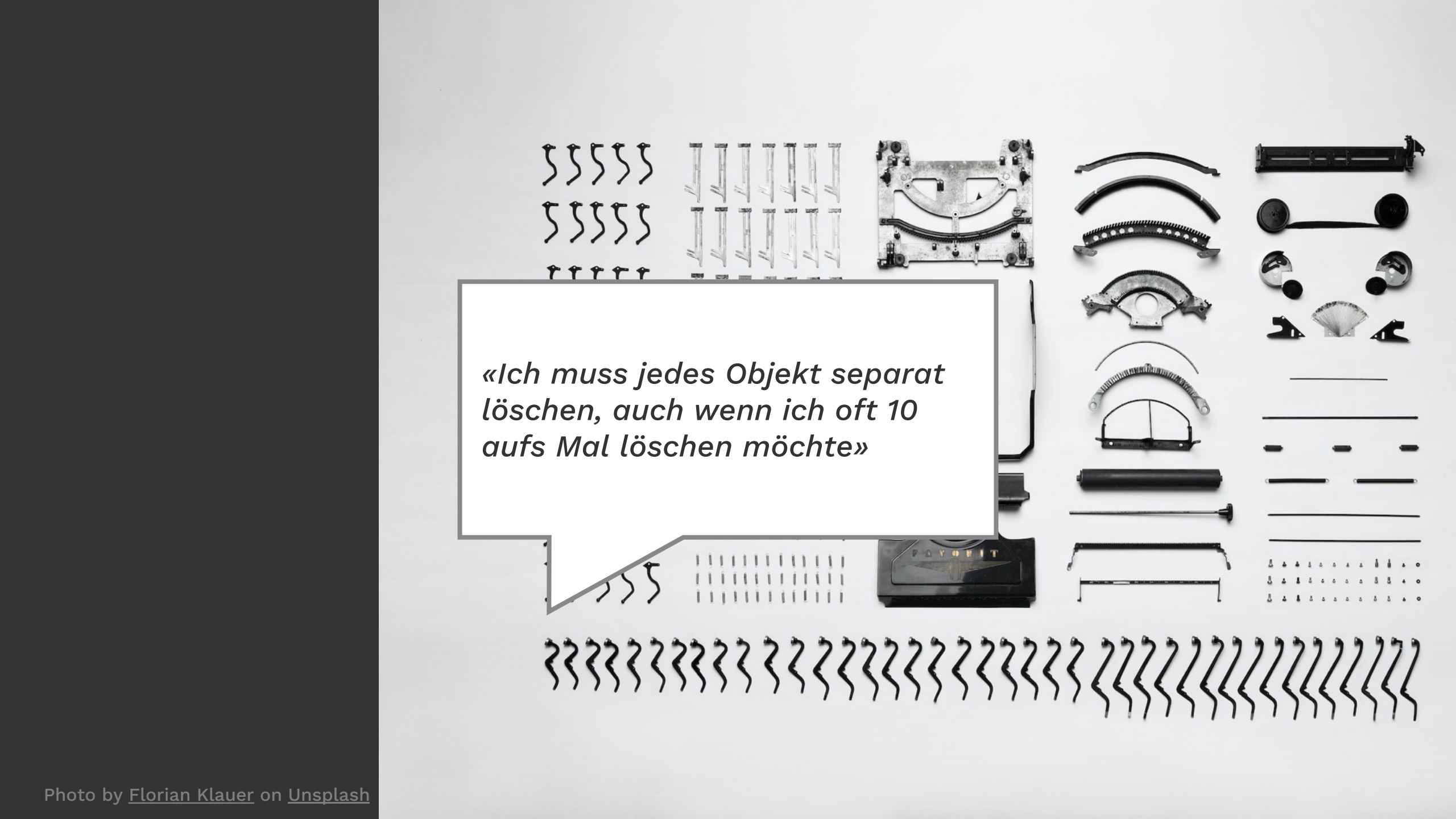
Zu restriktive Berechtigungen

Führen zu Rückfragen und manuellen Interventionen

Unpassende Berechtigungen kommen von...

- Unklaren oder unbekanntem Anforderungen an das Berechtigungssystem
 - Deshalb Tendenz zu restriktiven Berechtigungen
- Angst vor Fehlern, Wunsch nach Kontrolle
 - → Restriktive Berechtigungen
- Allerdings: Wirklich passende Berechtigungen sind sehr schwierig zu erstellen

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
- 6. *Fehlende Massenoperationen***
7. *Ausserdem...*
8. *Positives!*
9. *Verbesserungsmöglichkeiten*



«Ich muss jedes Objekt separat löschen, auch wenn ich oft 10 aufs Mal löschen möchte»

Anzeichen für fehlende Massenoperationen

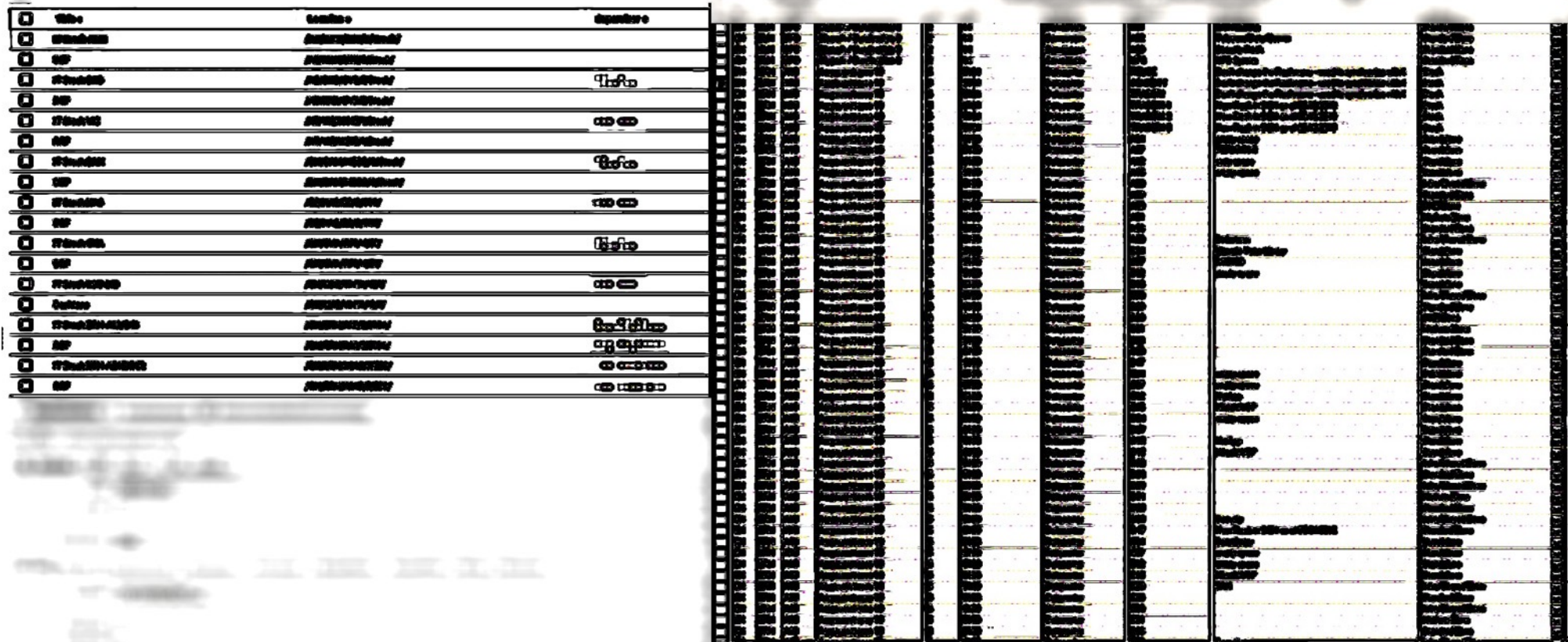
Nutzer müssen Dinge einzeln in die Hände nehmen, und zwar mehrmals.

Fehlende Massenoperationen kommen von...

- Unklarheit, welche Operationen wie häufig und in welcher Kombination benötigt werden
- Unpassendem Interaktionsdesign
 - Beispiel: Sequenzielle Bearbeitung in Wizard statt direkte Bearbeitung in Tabelle
- Bewusst nicht umgesetzt, weil komplex

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
6. *Fehlende Massenoperationen*
- 7. *Ausserdem...***
8. *Positives!*
9. *Verbesserungsmöglichkeiten*





Dieselben Daten,
dieselben Fakten

Nutzerzentriert?

Viele der beobachteten Probleme kommen tatsächlich davon, dass:

- Anforderungen nicht mit Nutzern erhoben wurden
- Systeme einfach hingestellt wurden

«Software wird immer nur noch komplexer, teuer, bürokratischer, aber beim Nutzer kommt nichts an»

Höre ich immer wieder von Stakeholdern & Management.

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. **Positives!**
9. *Verbesserungsmöglichkeiten*



«In den Kommentaren ist alles protokolliert. So geht nichts vergessen.»

(Kommentarfunktion in einem Issue-Tracking-System)

«Wenn der Zuständige in den Ferien ist, kann ich es auch selbst freigeben.»

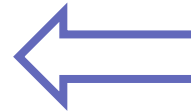
(Offene Berechtigungen in einem Bestellsystem)

«Ich kann eine Offerte während des Gesprächs anpassen und sehe live, was wir daran verdienen.»

(Integration & Automation in einem CRM)

«Detaillierungsgrad ist super gelöst, man weiss immer genau, wo etwas ist.»

(Informationsarchitektur eines Issue-Tracking-Systems)



«Zusammenarbeit ist viel besser geworden. Mit diesem Sounding Board ist viel verbessert worden.»

(Wie es dazu gekommen ist)

1. *Überstrukturierung*
2. *Mangelnde Integration*
3. *Mangelnde Automatismen*
4. *Ungenügende Kollaborations-Funktionen*
5. *Unpassende Berechtigungen*
6. *Fehlende Massenoperationen*
7. *Ausserdem...*
8. *Positives!*
9. **Verbesserungsmöglichkeiten**



Nutzerzentriert sein. Jedes Bisschen hilft.

- Nutzerforschung machen
- Entwickler einbeziehen
- Stakeholder & Management einbeziehen
 - z.B. als Beobachter bei Usability Tests
- Feedbackmöglichkeiten bereitstellen
 - Auch als Feature



**Jedes Bisschen hilft.
(Meine Meinung.)**

Wie wir
Struktur / Informationsarchitektur
verbessern

1. Bewusstsein für obengenannte Probleme schaffen
 1. Insbesondere dafür, dass *zu viel* Struktur auch problematisch ist
2. Minimale Prozesse und Formulare designen
 - Auch als Trockenübung
 - Auch mittendrin im Projekt oder Betrieb
 - Agiles Vorgehen: Was ist der kleinstmögliche nächste Schritt?
3. Daten & Analytics anschauen
 - Wie oft erscheinen gleiche Kombinationen?
 - Wie oft ist etwas leer?
 - Wie oft gibt es Alibi-Eingaben?

Wie wir **Integration** verbessern

Es muss nicht immer eine Softwareschnittstelle sein.

Stufen:

1. Separate Systeme
2. «Integration» via Zwischenablage: Sauberes Copy-Paste ermöglichen
3. «Integration» via Datei: Export und Import
 - Z.B. CSV, Excel, Text
4. Software-Schnittstelle
5. Einzelnes, vollintegriertes System

Diese beiden Möglichkeiten
werden gerne vergessen

Wie wir
Automatismen
verbessern

Siehe Nutzerzentrierung.

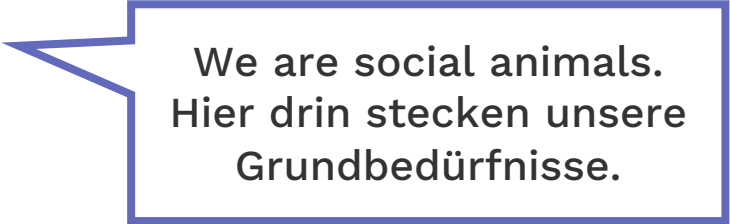
Und: Automatismen müssen für den Nutzer

- vorhersehbar
- nachvollziehbar
- und nach Möglichkeit steuerbar sein.

Wie wir
Kollaborationsfunktionen
verbessern

Wenn möglich:

- Kommentarfunktionen auf Geschäftsfall-/Artefakt-Ebene
- Anzeige der Zuständigkeiten
- Gleichzeitige Bearbeitung



We are social animals.
Hier drin stecken unsere
Grundbedürfnisse.

Allerdings: Hier scheint technische Machbarkeit oft ein Problem zu sein.

Wie wir
Berechtigungen
verbessern

Freiheit
aber
Konsequenzen



Konkret: Lieber etwas tolerantere Berechtigungen,
verbunden mit Nachvollziehbarkeit und Undo

Wie wir
Produktivität*
verbessern

Bewusstsein schaffen dafür, dass Software:

- Uns dabei unterstützen soll, unsere Zeit sinnvoll zu nutzen
- Dafür geschaffen ist, repetitive Tätigkeiten zu minimieren
- Sich lohnen muss, betriebs- und volkswirtschaftlich gesehen.

Danke.

Timo Würsch

UX Architect & Gründer
Interaktionswerk GmbH

[@timo3000](#) [@ix_werk](#)

timo.wuersch@interaktionswerk.ch