Drehbuch Screencast eUmzug

# Begrüssung, Zielsetzung und Aufbau

1. TITELBILD: Herzlich **Willkommen** zum Screencast «ZHAW-Prototyp für den eUmzug».
2. LINKEDIN-PROFIL: Mein **Name** ist Björn Scheppler. Ich bin wissenschaftlicher Mitarbeiter an der ZHAW School of Management and Law. Unter anderem unterrichte ich das Modul Geschäftsprozessintegration für angehende Wirtschaftsinformatiker.
3. GITHUB EUMZUG: Als Teil dieses Moduls erarbeitete ich gemeinsam mit den [ANS ENDE SCROLLEN] hier genannten **Personen** von 2017 bis 2019 die im Folgenden vorgestellte Lösung.
4. AN START SCROLLEN: Das **Ziel** dieses Prototyps ist, anhand eines möglichst realitätsnahen Beispiels praktisch aufzuzeigen, was Geschäftsprozessintegration konkret bedeutet.
5. ZU BILD SCROLLEN: Entsprechend werden, wie aus dem Übersichts-Modell bereits zu erkennen ist, sowohl Menschen über Aufgaben und Formulare in den Prozess eingebunden, als auch zahlreiche Umsysteme über Schnittstellen wie SOAP, REST und mehr.
6. Wer ist das **Zielpublikum** für die folgenden Videos und die recht umfassend dokumentierten Github-Repositories? Einerseits erhalten Studierende die Möglichkeit, in ein realitätsnahes Beispiel detailliert Einblick zu nehmen. Anderseits können auch Junior-Process Engineers profitieren, welche Beispiel-Code für Camunda-Lösungen suchen. Es ist also eine Ergänzung zu den bereits bestehenden Beispielen auf Github: [<https://github.com/camunda/camunda-bpm-examples>] und [<https://github.com/camunda-consulting/code>]
7. Wie ist dieser Screencast **aufgebaut**?
   1. Das nächste Video wird kurz auf die reale Ausgangslage und die gewählte Lösung in der Schweiz eingehen. Dabei wird auch kurz das Frontend der «echten» eUmzug-Lösung gezeigt.
   2. Dann folgt ein Video, wo ich in unserem Prototypen eine neue Prozessmeldung mit Testdaten durchspiele, dabei allerdings nur die Sicht des Meldepflichtigen zeige, um das Video kurz und knackig zu halten. Ich zeige also nur die Benutzerformulare und E-Mails, welche der Benutzer erhält.
   3. Als letztes folgt ein ausführlicheres Video, wo ich nochmals denselben Prozess durchspiele, nun aber parallel dazu die BPMN-Modelle zeige und grob erläutere, was pro Aktivität im Hintergrund geschieht.

# Ausgangslage und eumzug.swiss

## Ausgangslage

1. eCH-0221, S. 11: Die inhaltliche Ausgangslage ist rasch erklärt:
   1. Eine Person oder eine ganze Familie möchte von einer Wohnung in eine andere umziehen.
   2. Zu diesem Zweck müssen die Personen sich bei der Wegszugsgemeinde ab- und bei der Zuzugsgemeinde abmelden.
   3. Bisher mussten die Personen hierzu physisch bei den Gemeindekanzleien vorsprechen.
2. Das Ziel liegt daher auf der Hand: Neu soll dies über einen Online-Schalter möglich sein, damit folgende Vorteile genutzt werden können:
   1. Das Melden des Umzugs ist rund um die Uhr und auch am Wochenende möglich.
   2. Hat man ein Dokument zu Hause vergessen, muss man nicht mehrfach beim Schalter vorsprechen.
   3. Für die Gemeinden sind alle Dokumente direkt digital vorhanden, müssen also nicht eingescannt werden.
   4. Auch die Daten (Personalien, usw.) sind digital vorhanden und fliessen daher ohne Tipparbeit direkt in die Einwohnerkontrollsysteme.

## Geschichte von eumzug.swiss

1. Das Ziel tönt eigentlich reichlich einfach, hat aber mehrere Jahre in Anspruch genommen, bis es über einzelne Gemeinden hinweg und sogar über Kantone hinweg funktioniert.
2. Dies liegt unter anderem an der föderalen Struktur in der Schweiz mit unterschiedlichen Gesetzgebungen, aber auch daran alle Anbieter der Einwohnerkontrollsysteme, Verbände, usw. mit an Boot zu haben.
3. Und dann ist da natürlich auch die Herausforderung der technischen Umsetzung mit allen Schnittstellen, Standards und vielem mehr.
4. Vorreiter war der Kanton Zürich, dessen Lösung inzwischen fast schweizweit übernommen wurde unter dem gemeinsamen Auftritt eumzug.swiss [WEBSEITE ZEIGEN]
5. [DURCH STANDARD BLÄTTERN] Es gibt auch den eCH-Standard 0221 mit einem Referenzmodell, wie eine solche eUmzug-Plattform aussehen muss. Abgesehen von der Lösung des Kantons Zürich und unserem Prototyp ist mir aber keine weitere Implementation dieses Referenzmodells bekannt.
6. [STANDARDS IM ORDNER ZEIGEN] Damit man sich ein Bild davon machen kann, wie anspruchsvoll allein das Verstehen der Standards ist, sei darauf hingewiesen, dass der eCH-Standard 0221 selbst wieder auf weitere der hier gezeigten Standards verweist.

## Echte Lösung

1. [EUMZUG.SWISS öffnen] Bevor ich im nächsten Video den von uns erstellten Prototypen zeige, hier ein kurzer Ausschnitt daraus, wie die tatsächliche Applikation aus Frontend-Sicht funktioniert.
2. Da es um meinen eigenen Umzug ging, habe ich die entsprechenden Angaben unsichtbar gemacht.
3. Bei vielen der Schritte werden im Hintergrund Umsysteme synchron angesprochen, z.B. …

# Prototyp aus Frontend-Sicht

## Einführung

1. [GITHUB-REPO EUMZUG] Im Hintergrund habe ich bereits dieses Maven-Projekt als auch alle darin verlinkten Maven-Projekte für die Umsysteme gestartet.
2. [BROWSER] Entsprechend kann ich mich als Benutzer beim Prozessportal anmelden.
3. Natürlich würde dieses in der Realität ziemlich anders aussehen, aber immerhin sieht man, dass es etwas customized ist, in diesem Fall fiktiv für den Kanton Bern.
4. In der Aufgabenliste sehe ich, dass ich aktuell als Benutzer keine Aufgaben habe.
5. Ich kann nun einen Prozess starten, für den ich berechtigt bin, konkret ist dies aktuell nur der Umzugsmeldeprozess [START PROCESS]. Alle weiteren hier sichtbaren Prozesse wären in der Realität ausgeblendet.
6. Von hier ausgehend könnte ich den Prozess starten, was ich aber nicht tue, da ich sonst jedes Formular von Hand ausfüllen müsste.
7. [SOAPUI] Stattdessen habe ich hier bereits Testdaten vorbereitet und starte daher den Prozess über die REST API von Camunda mit diesen Testdaten.

## Prozess aus Sicht des Benutzers

1. [BROWSER aktualisieren] Nun habe ich eine erste Aufgabe erhalten mit dem ersten Formular.
2. Wie auch in eumzug.swiss muss ich meine Personalien und meinen jetzigen Wohnort eingeben.
3. Mit einem Klick auf «Abschliessen» geht es Weiter zum nächsten Formular. Respektive stimmt das nicht ganz und ist eine Einschränkung des Prototyps, auf die ich in der Dokumentation in Github genauer eingehe. Sie hat zur Folge, dass für jedes Formular eine neue Aufgabe erstellt wird, was natürlich so in der Realität nicht sinnvoll wäre.
4. Im Personenregister wurden die Personalien offenbar gefunden und das System gibt mir daher mögliche mitumziehende Personen bekannt im gleichen Haushalt [ANNA AUSWÄHLEN].
5. Wegzugsadresse erfassen ist selbsterklärend, genauso wie Wegzugsadresse. In beiden Fällen wird das Gebäude- und Wohnungsregister aufgerufen, so dass ich nun auch sehe, welche Wohnungen überhaupt an der neuen Adresse zur Auswahl stehen.
6. Mit Umzugsdaten erfassen und Wohnverhältnis erfassen kommen zwei weitere selbsterklärende Formulare.
7. Im nächsten Formular muss ich die Versicherungskartennummern erfassen, da in der Schweiz die Krankengrundversicherung beim Umzug geprüft wird.
8. Entsprechende Testdaten kann ich von der Dokumentation in Github übernehmen. Die Prüfung erfolgt durch einen synchronen Aufruf eines Umsystems. Bei Hans ist alles bestens, Anna hingegen ist nicht grundversichert. Trotzdem kann man fortfahren.
9. Bei den Kontaktangaben werden die Angaben erfasst für die Benachrichtigung des Meldepflichtigen über das Ergebnis der Prüfung. Nebst E-Mail haben wir just-for-fun auch SMS integriert.
10. Bei Dokumente hochladen sind je nach Person weitere Dokumente erforderlich. Wir verzichten aus Zeitgründen jetzt darauf.
11. Nun werden mir die dynamisch berechneten Gebühren angezeigt sowie alle von mir erfassten Angaben zur Kontrolle.
12. Ich akzeptiere und komme damit zum Zahlungsteil. Dieser nutzt den Stripe-Dienst. Entsprechende Testkreditkarten habe ich in der Github-Dokumentation aufgeführt.
13. Clientseitig hat die «Zahlung» funktioniert.
14. Offenbar auch Server-Seitig. Meinerseits ist nun alles erledigt.
15. Nun wurden im Hintergrund die Daten an die Einwohnerkontrollsysteme gesendet und diese prüfen meine Angaben, teilweise auch durch Einbezug eines Menschen. Wir haben dies allerdings lediglich mit einem Zufallsgenerator simuliert.
16. Nach einiger Zeit kommt die erste Meldung, dass die Umzugsmeldung erfolgreich erfasst wurde.
17. Und nach noch etwas mehr Zeit die zweite Meldung, dass mein Umzug von den Gemeinden akzeptiert wurde.
18. Das war’s.

# Prototyp mit Hintergrund-Informationen

Diesen mache ich spontan. Grundprinzip: Nebenbei läuft das Cockpit, wo ich die Prozessmodelle zeige, damit man sieht, wo man steht als auch, welche Verzweigungen und Ausnahmefälle es gäbe. Zudem wird beim jeweils ersten Typ eines Tasks etwas mehr gezeigt:

1. Erster User Task -> HTML
2. Service Task ‘Person identifizieren’ -> Java Delegate