

CiviCRM Mitgliedschaften

- Flexibel und einfach erweiterbar
- **Aber:** britisch/amerikanisches Konzept: Mitgliedschaft als „Dienstleistung“ statt „Vertrag“
- **Vertrag:** „Du hast gesagt Du willst Mitglied sein, und schuldest uns den Beitrag bis Du kündigst“
- **Dienstleistung:** „Du bist Mitglied solange Du zahlst“. Es gibt stets ein aktuelles Enddatum.

Statusmodell

Typische Status sind z.B.

- neu → laufend → Zahlungsfrist → ausgelaufen
- laufend → Kündigung eingereicht → gekündigt

Statusübergänge können mit Bordmitteln automatisiert werden, aber **nur** basierend auf Start- und Enddatum

Mitgliedsbeiträge

Es gibt eine gute Integration mit Zahlungsprozessoren (Zahlungsdienstleister)

Widerspricht leider:

- Konzept der Mitgliedschaft als Vertrag
- Konzept der selbst erstellten Einzüge (CiviSEPA)

Was tun?

Ansätze zur Automatisierung

- **Mit CiviCRM Bordmitteln** – viel Handarbeit
- **Project60 Membership Extension**
Vielseitig aber komplex
- **Contract Extension** (sehr aufwändig)
Hoher Automatisierungsgrad und
Funktionsumfang (Pausieren, geplante
Beitragsanpassungen, etc.)
- **Eigenkonstruktion** (mit generischen Tools)
Kann beliebige Anforderungen abbilden

P60 Membership Extension

Sinnvolle Zusatzfeatures

- Mehr Kommunikationstokens
- Verwaltung der Mitgliedsnummer
- Aufzeichnung von Kündigungsdatum und Kündigungsgrund
- Verbindung mit Zahlungsvertrag (z.B. SEPA)
- Fremdzahler
- Abgeleitete Felder (Beitrag, Rhythmus, Differenz, Beitragsrückstand)

P60 Membership Extension

Routinen

- Zuordnung von Zuwendungen zu Mitgliedschaften
- Berechnung des Zahlungsrückstands
- Automatische Verlängerung basierend auf „Mitgliedskonto“
- Manuelle Bearbeitung nur für die „restlichen“ Mitgliedschaften (z.B. „Zahlungsfrist“)

CiviBanking und Mitgliedschaften

Optionen

- Membership PostProcessor
 - Zahlungen mit Mitgliedschaft verbinden
 - Mitgliedschaften verlängern
- Recurring Fails PostProcessor
 - Erkennung von Mustern bei den Einzugsfehlschlägen, und entsprechende Aktionen auslösen: Terminierung, beliebige via API Call

Automatisierung Ansätze

Synchron

- CiviRules
- Diverse spezialisierte Extensions

Asynchron

- SQL Tasks
- FormProcessor + geplante Aufgaben
- Externes Datamining

Automatisierung Probleme

Synchron

- Positiv: Ergebnis sofort
- skaliert schlecht (z.B. bei Massенbearbeitung)

Asynchron

- Positiv: Skaliert sehr gut
- Wirkung zeitverzögert
- Ggf. Energieverschwendung