SBB CFF FFS

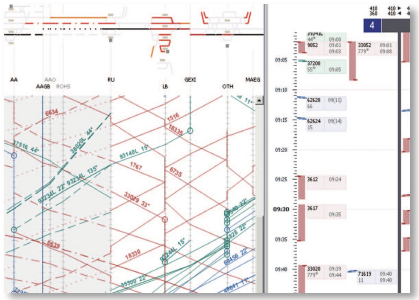
```
26 public class SendungAuswahlListAdapter extends ArrayAdapter<SendungAuswahlDTO> {
27
28     /**
29      * Constructor
30      *
31      * @param context The current context.
32      * @param sendungAuswahlDTOList The list of SendungAuswahlDTO items to display. NonNu
33      */
34     public SendungAuswahlListAdapter(final Context context, final List<SendungAuswahlDTO>
35         super(context, R.layout.sendung_selection_dialog_list_item, sendungAuswahlDTOList
36         if (sendungAuswahlDTOList == null) {
37             throw new IllegalArgumentException("Paramenter sendungAuswahlDTOList is not a
38         }
39     }
40
41     @Override
42     public View getView(final int position, View convertView, final ViewGroup parent) {
43         if (position < getCount()) {
44             final SendungAuswahlListViewHolder viewHolder;
```

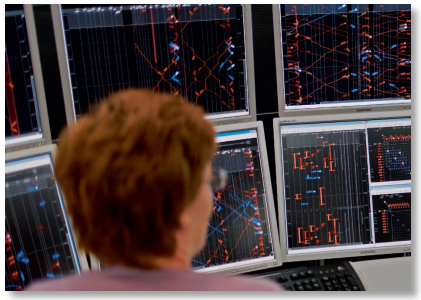
«Lieber Product Owner, dein Business Case fiel leider der technischen Schuld zum Opfer.»

Vladimir Riecicky, ICT-Beschaffungskonferenz 2016

IT in der Bahnproduktion.

Unterstützte Prozesse.





Planung › **Disposition** › **Operative Lenkung** › **Leittechnik**

3’600’000 geplante Züge pro Jahr

166’000’000 Trassenkilometer geplant, vereinbart und verrechnet

76’000 in CERES verarbeitete Transaktionen täglich

800 Touren täglich bearbeitet

25’000 Zugfahrten pro Tag mit gleisgenauen Fahrwegen generieren

1’000’000 Konfliktlösungen pro Tag realisieren

4000 Zugsprognosen alle **3** Sekunden neu berechnet

1’500 Lenkbefehle um Strom von 22’000 Haushalten zu sparen

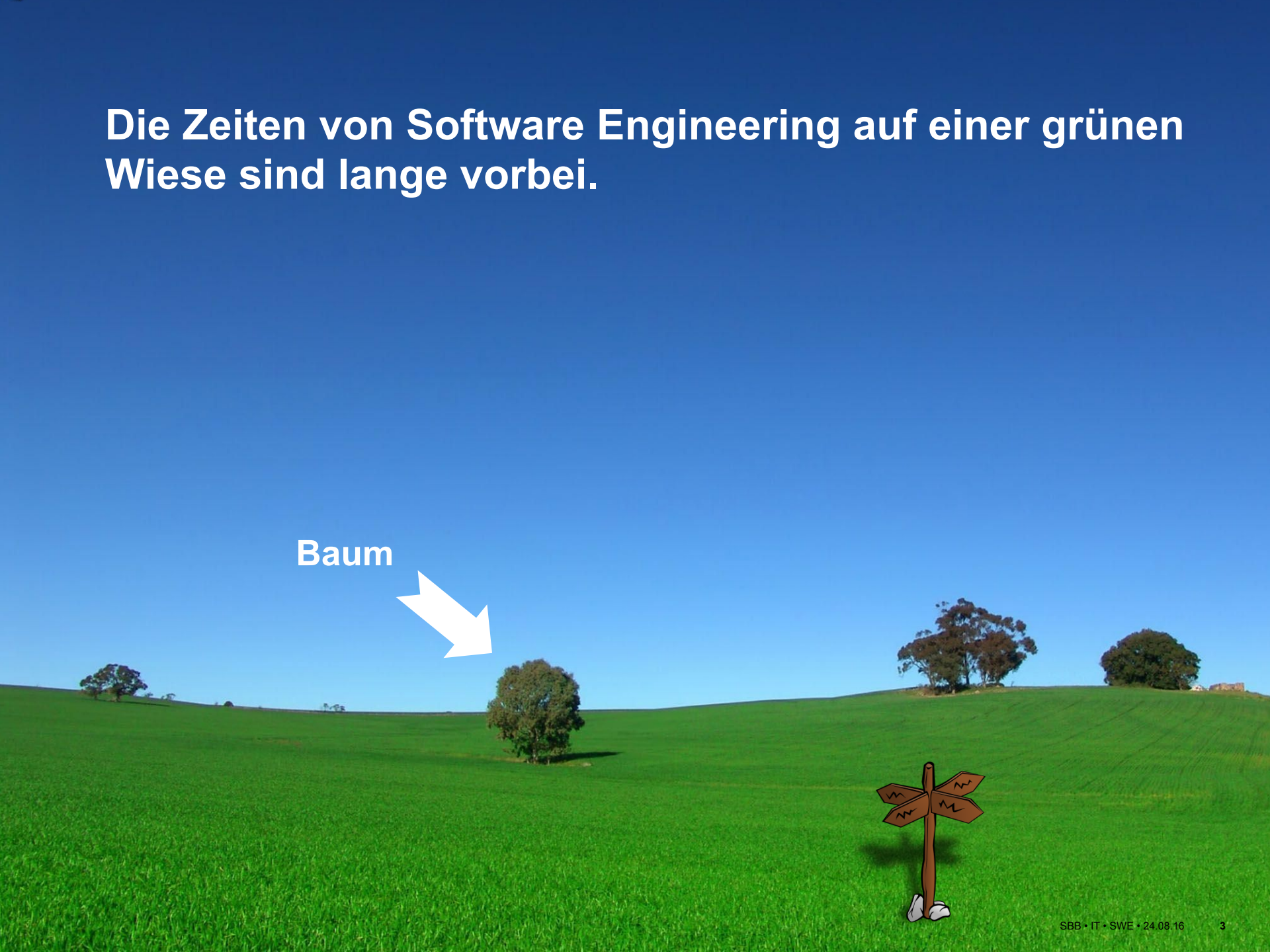
10’000 Optimierungs-varianten alle 6 Sek. an Einfädelungspunkten

1’200’000 Echtzeitmeldungen in alle Kundeninfo-Kanäle

25’000 Rangierfahrstrassen

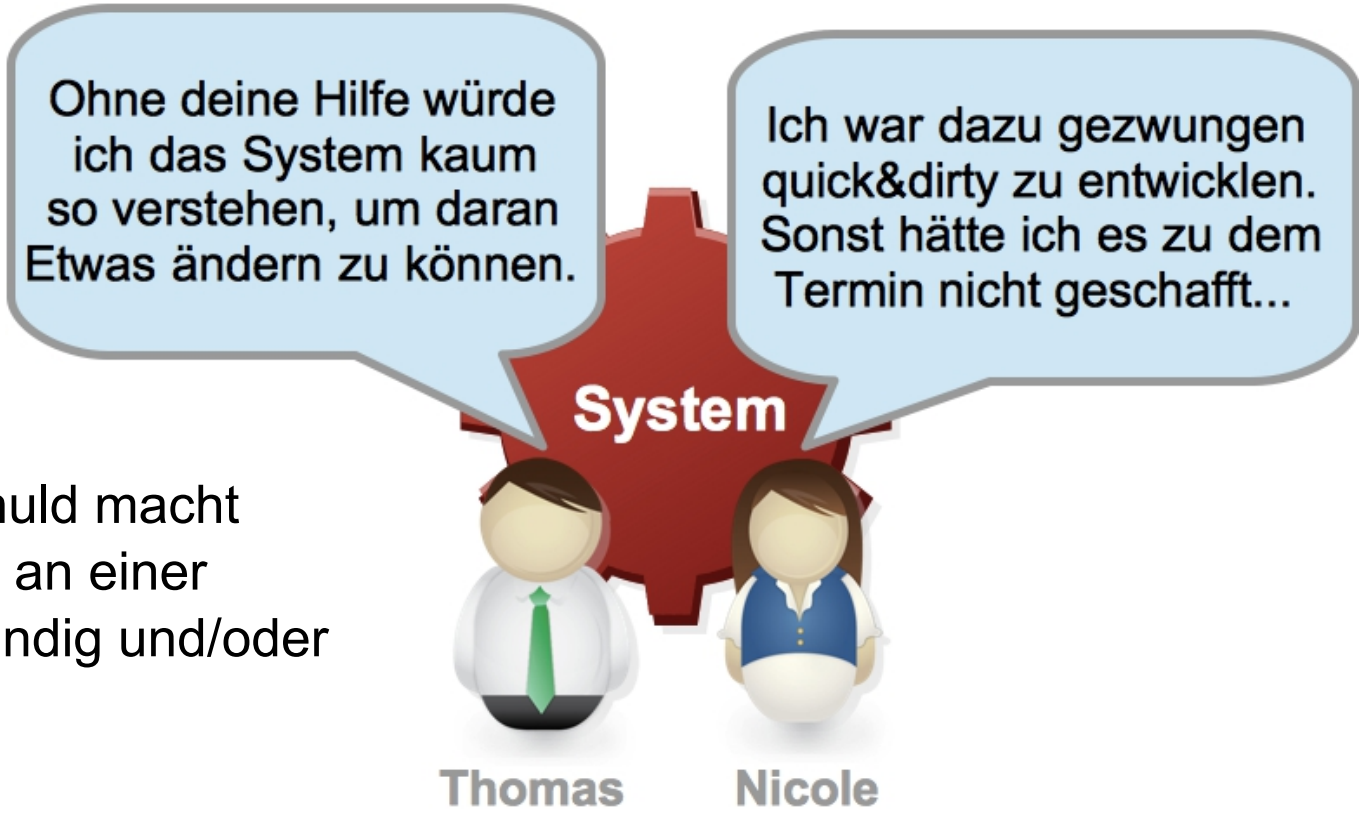
>50’000 Weichenstellbefehle

Die Zeiten von Software Engineering auf einer grünen Wiese sind lange vorbei.



Die grüne Wiese ist in vielen Fällen zu einem Urwald geworden.
Bach?
Baum?

Die «Gärtner» haben es oft nicht einfach den Urwald zu pflegen – Grund dafür ist die «technische Schuld».



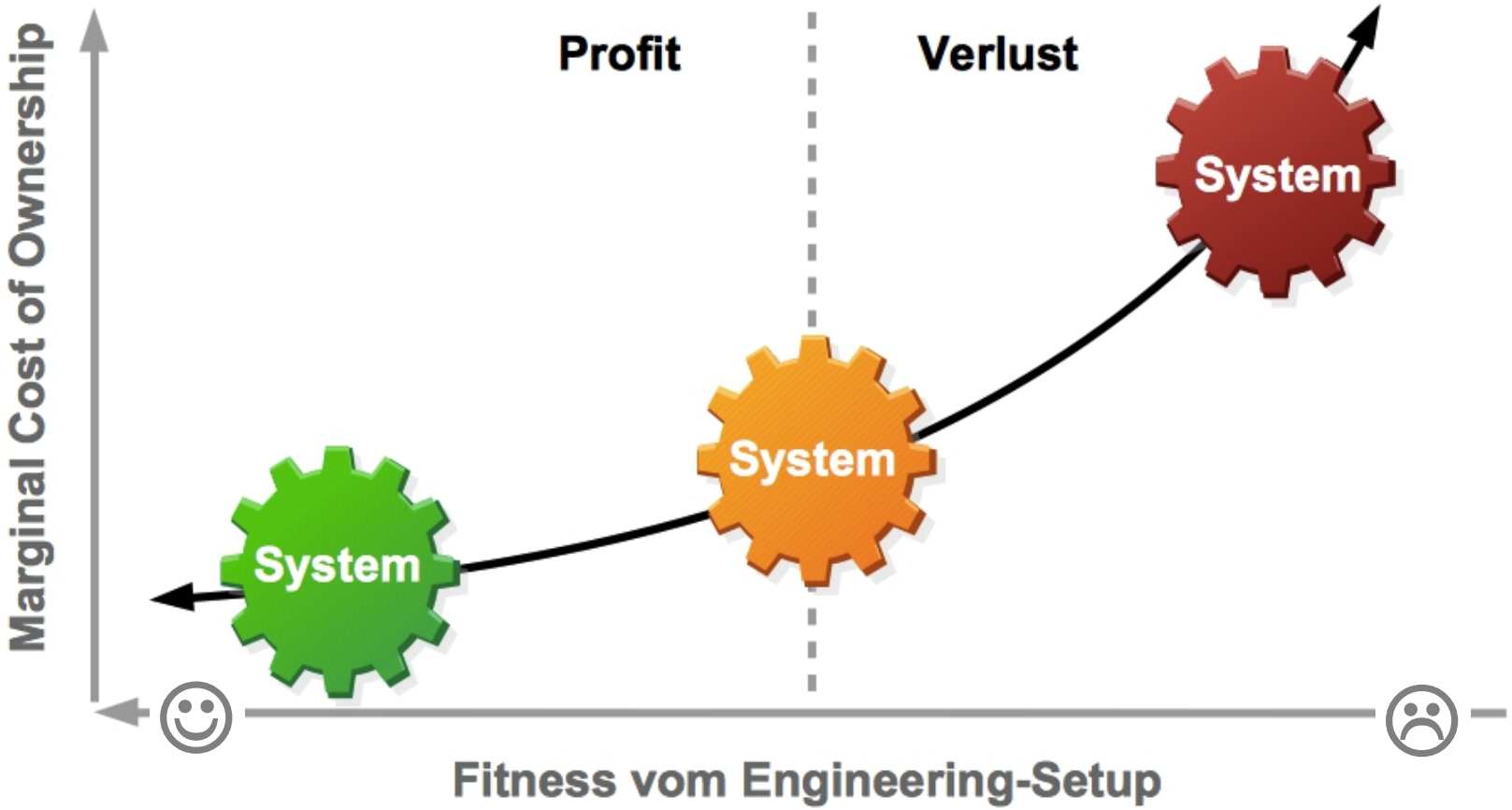
Technische Schuld macht Veränderungen an einer Software aufwändig und/oder riskant.

Es kommt soweit, dass sich Software Engineers vor der eigenen Software fürchten - Software schlägt auf die Stimmung.

Inhalt, Performance und vor allem Flexibilität unserer Geschäftsprozesse werden durch Software diktiert.

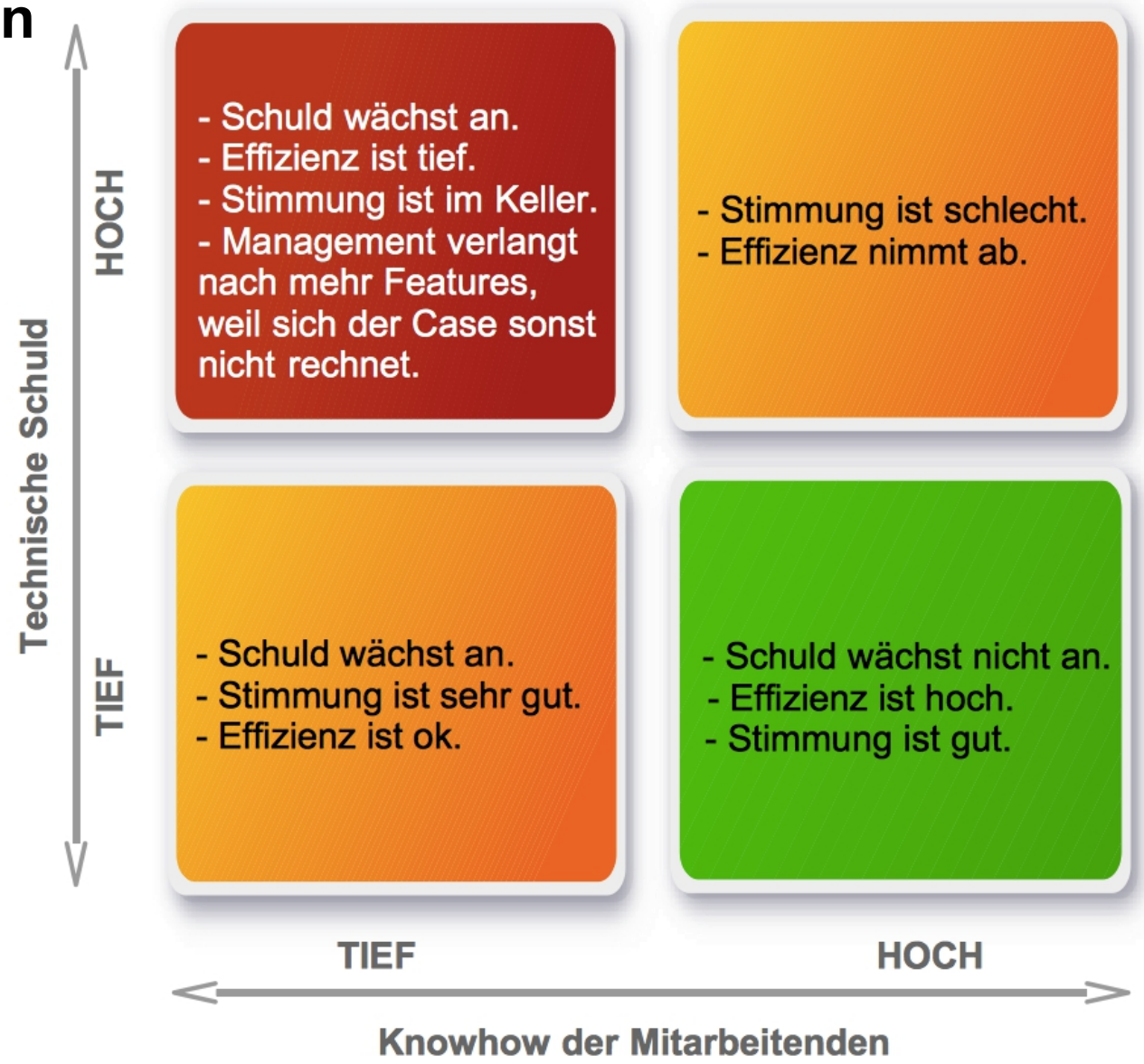
Im Zeitalter der Agilität und Digitalisierung ist Flexibilität (Reaktions-fähigkeit) entscheidend. Was bedeutet das?

Agilität und Digitalisierung verlangen nach einer neuen Steuerungsgrösse: Marginal Cost of Ownership.



Ziel: Umsetzungskosten von **gleich** (fachlich) komplexen Features bleiben über die Zeit **gleich**, oder nehmen ab!

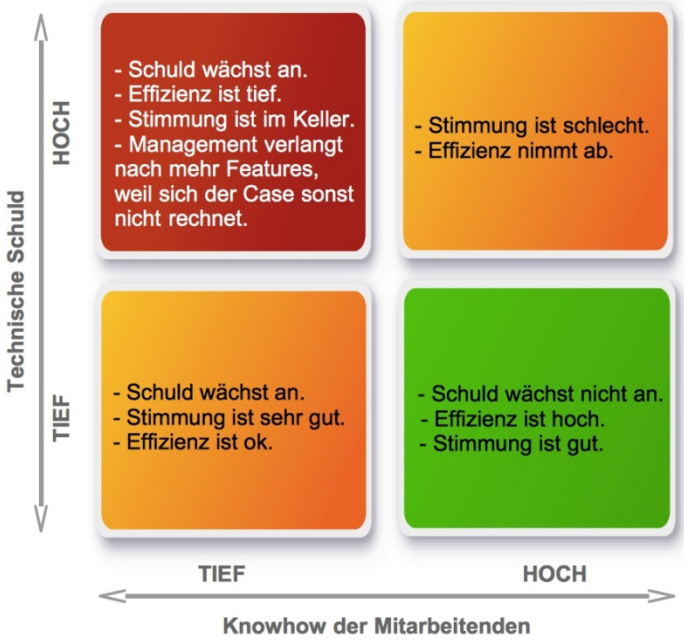
Alles hat seinen Grund…



- → Delivery-Speed
- → Sourcing
- → Mensch

Die Dynamik hinter Marginal Cost of Ownership (MCO) hat einige Implikationen für die IT-Beschaffung.

- Lieferanten sollen ihren Teil der Verantwortung für die Zunahme an technischer Schuld übernehmen. Der Risk-Share soll der beschafften Seniorität folgen.
- Technische Schuld einer Software soll als die Hauptursache für die Entstehung einer Abhängigkeit von den Lieferanten anerkannt werden.



- IT-Beschaffungscontrolling soll MCO als neue und wesentliche Komponente vom Einkaufserfolg integrieren.
- **Der Dialog zwischen dem IT-Einkauf und der IT-Entwicklung braucht im Zeitalter der Digitalisierung eine neue Qualität.**