Argumentativ zur besten Designentscheidung

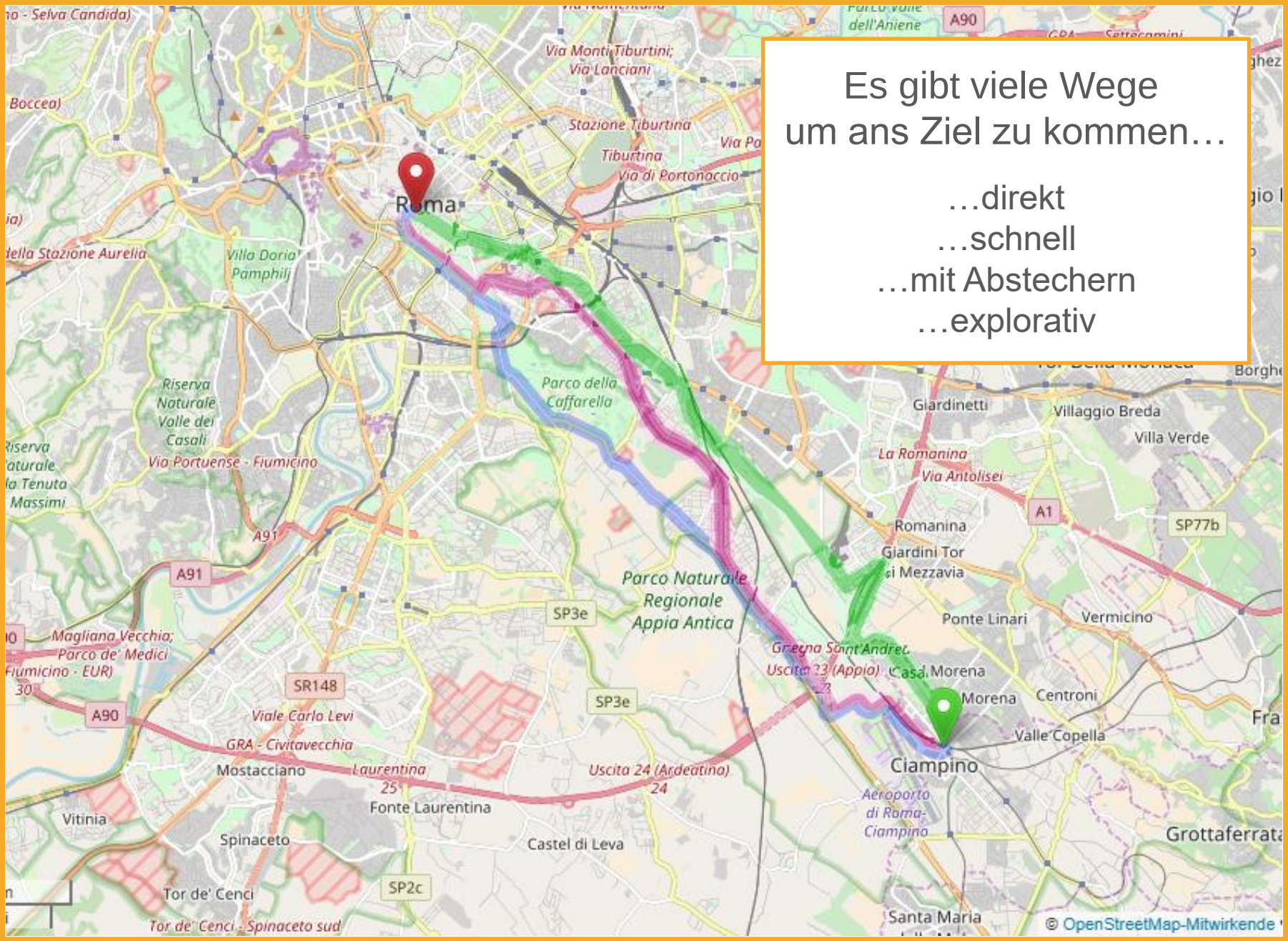
<for developers only…maybe>

Christian Rehn:

- Software Engineer
- www.principles-wiki.net
- www.design-types.net

Matthias Wittum:

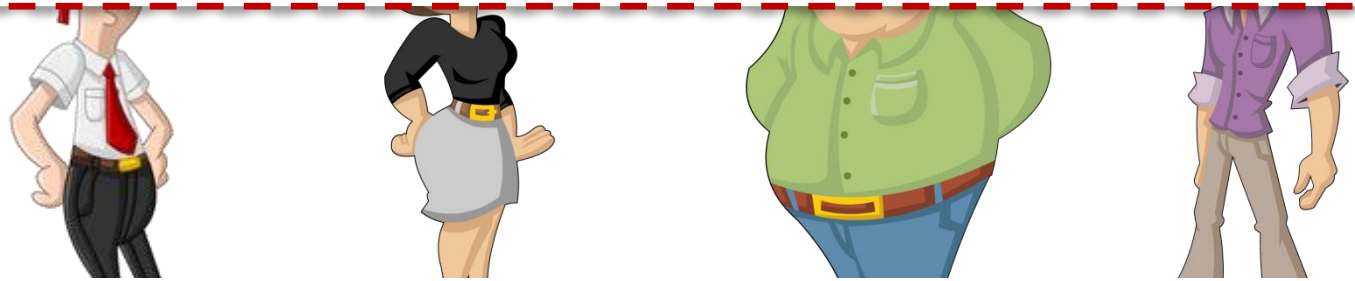
- Head of 1&1 Source Center
- www.design-types.net


Es gibt viele Wege
um ans Ziel zu kommen…
…direkt
…schnell
…mit Abstechern
…explorativ
Roma
Ciampino
Parco Naturale Regionale Appia Antica
Parco della Caffarella
Riserva Naturale Valle dei Casali
Villa Doria Pamphilj
Via Portuense - Fiumicino
Via Monti Tiburtini; Via Lanciani
Stazione Tiburtina
Tiburtina
Via di Portonaccio
Via Antolisei
La Romanina
Giardinetti
Villaggio Breda
Villa Verde
Romanina
Giardini Tor
Mezzavia
Ponte Linari
Vermicino
Casa Morena
Morena
Centroni
Valle Copella
Grottaferrata
Aeroporto di Roma-Ciampino
Uscita 24 (Ardeatina)
Laurentina
Fonte Laurentina
Castel di Leva
Mostacciano
Viale Carlo Levi
GRA - Civitavecchia
Vitinia
Spinaceto
Tor de' Cenci
Tor de' Cenci - Spinaceto sud
Santa Maria
A90
A91
A1
SP77b
SP3e
SP2c
SR148
Magliana Vecchia; Parco de' Medici
Fiumicino - EUR)
© OpenStreetMap-Mitwirkende

Szenario

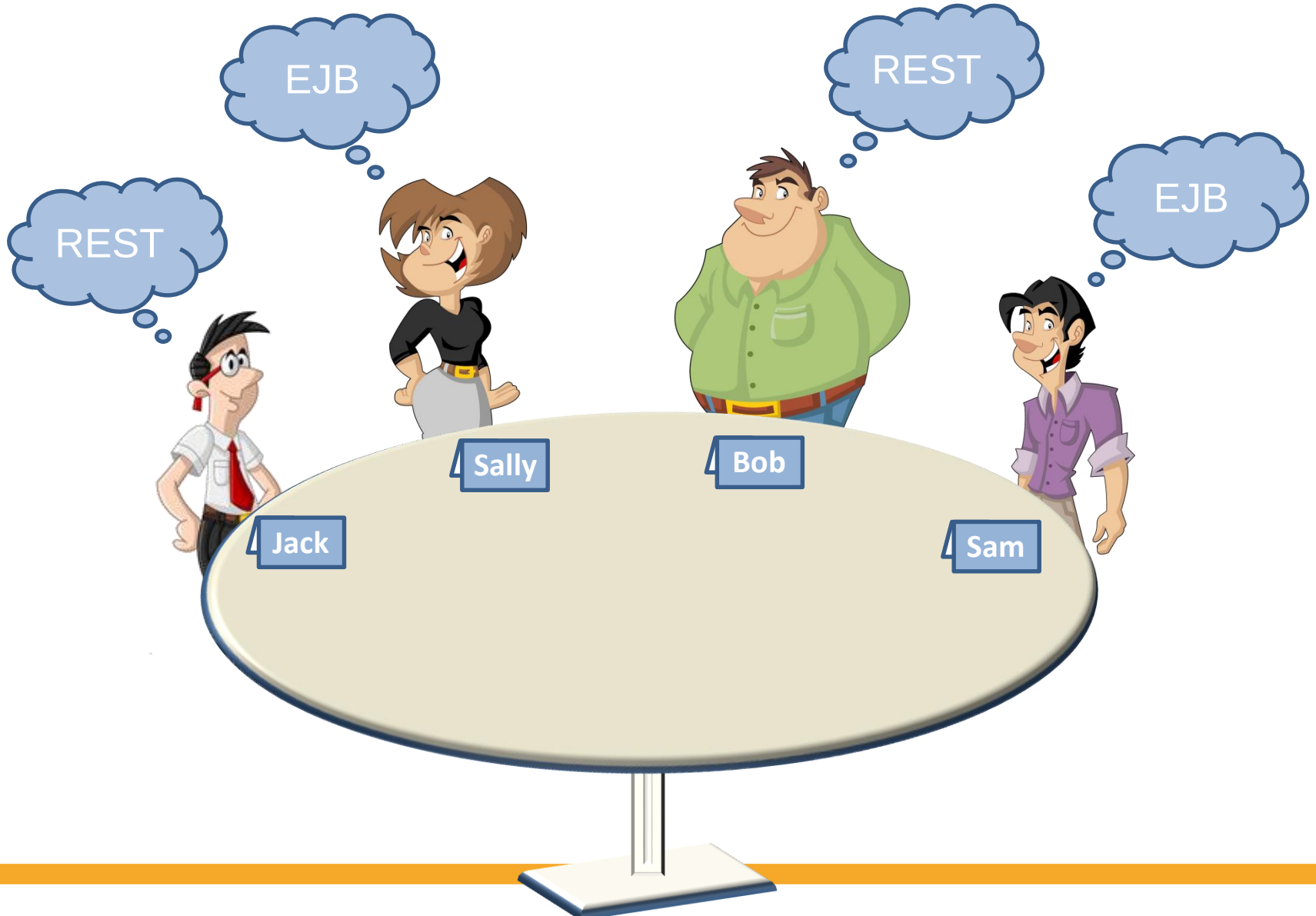
vs. { REST }

Ein völlig fiktives Szenario

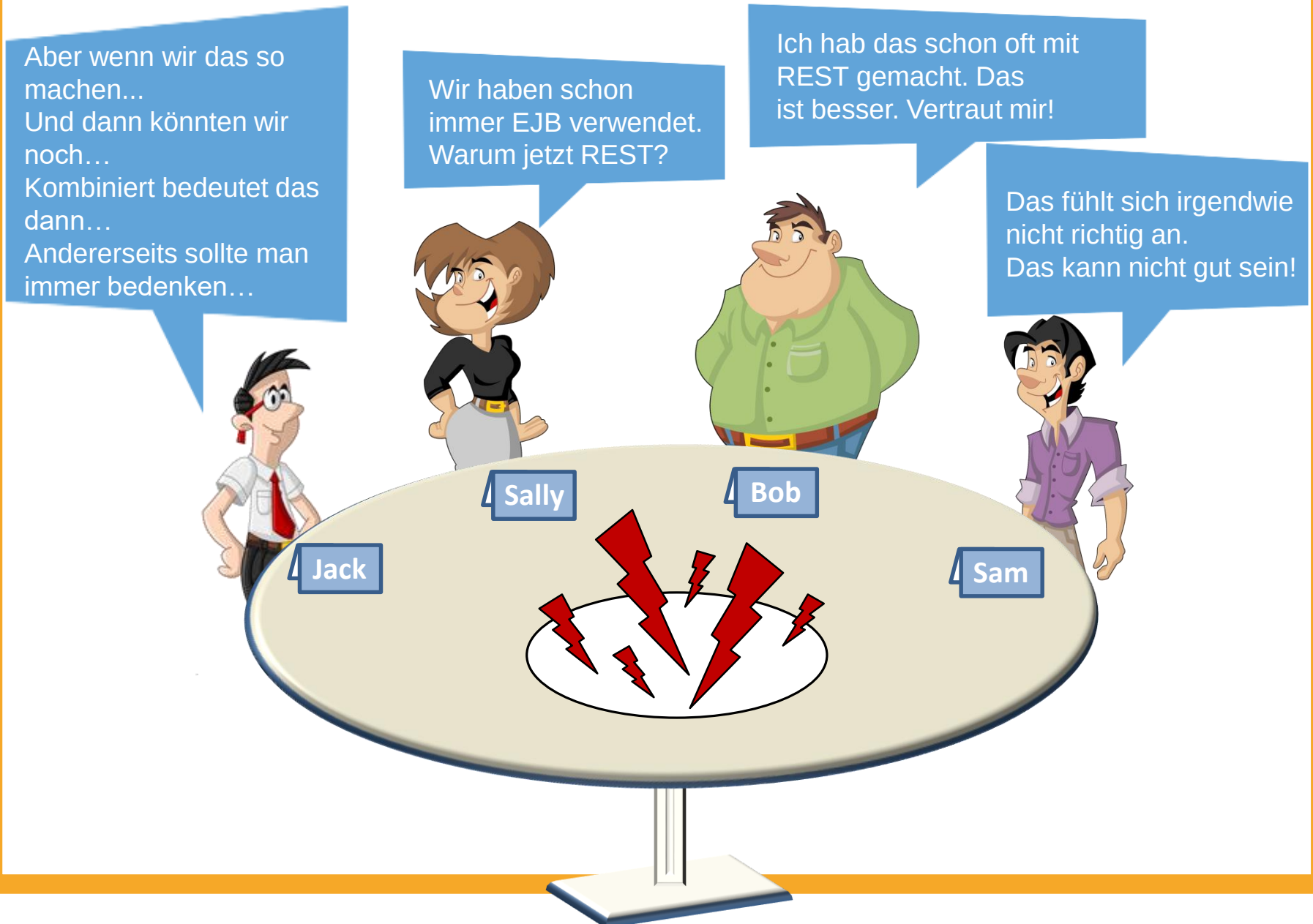


Eine Gruppe Entwickler diskutiert, ob eine neue Schnittstelle per EJB oder REST angeboten werden soll.

Technische Diskussion: „die historische Variante“



Technische Diskussion: „die historische Variante“



Überreden oder überzeugen?

Wir sind nicht die ersten...

IH/E

SDP

SoC

ML

KISS

ISP

SCP

DRY

SRP

LoD

OCP

RoP

DIP

FF

ECV

ZOI

MIMC

LC

UP

PSU

LLA

HC

MP

EUHM

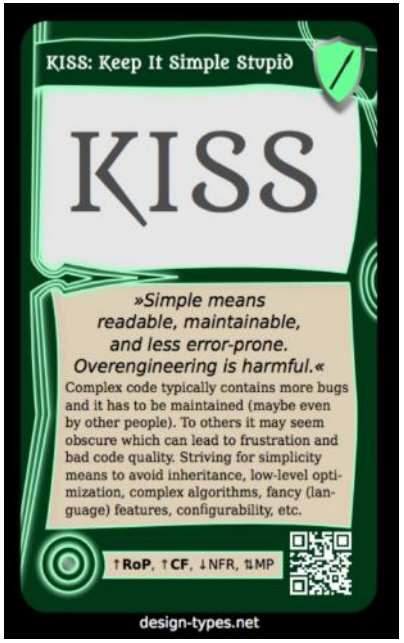
TdA/IE

PLS

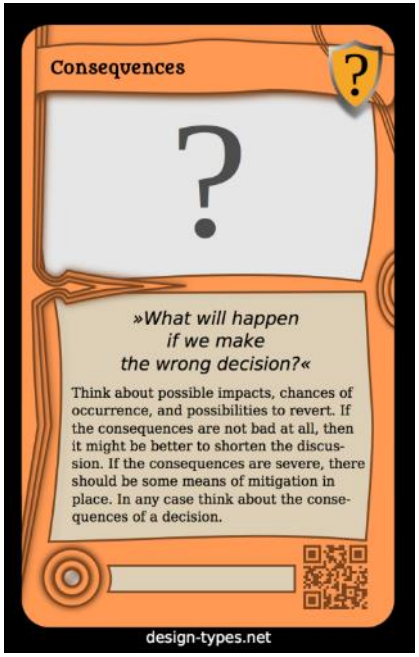
ADP

IAP

Design Cards – Argumentkarten

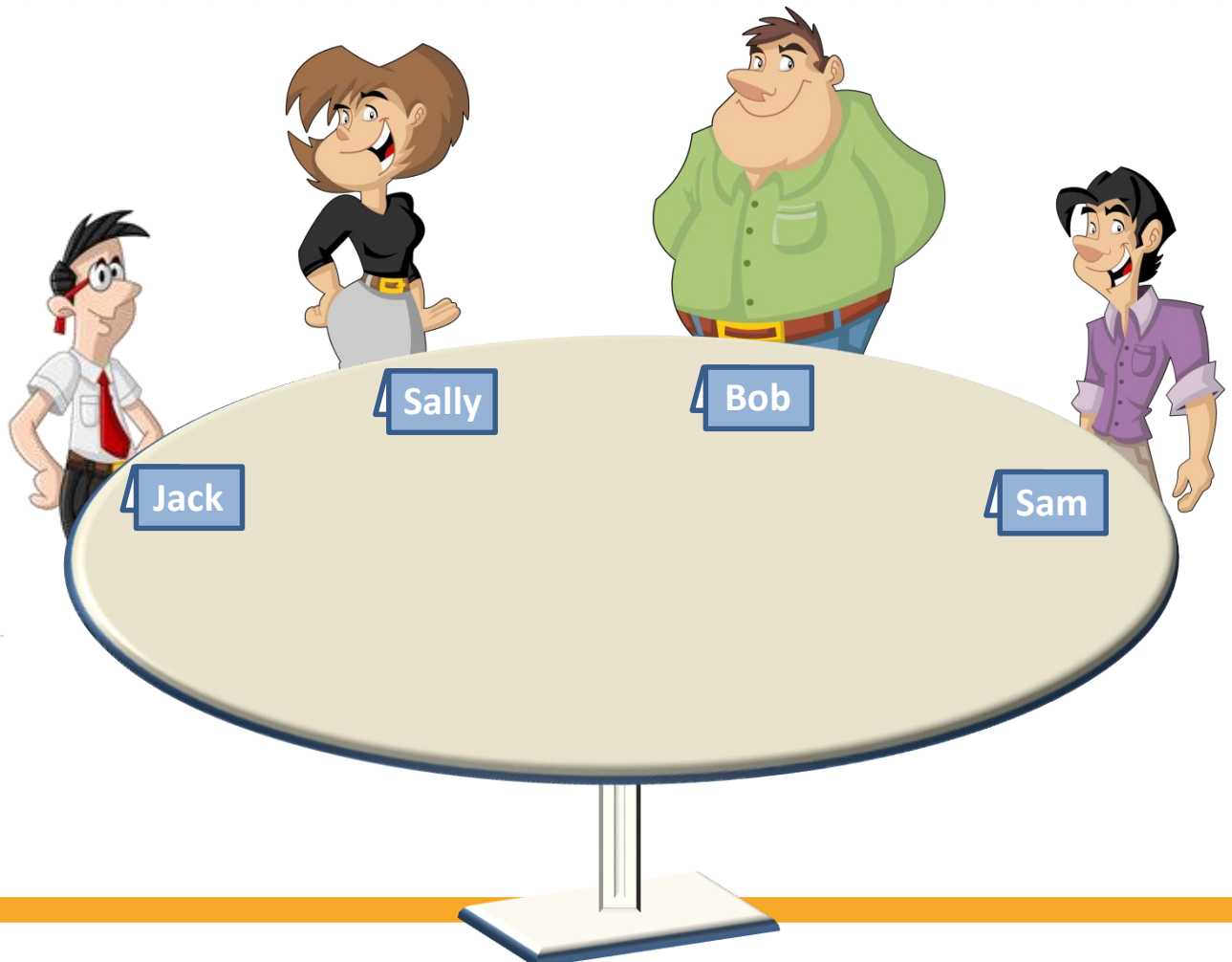


Design Cards – Moderationskarten

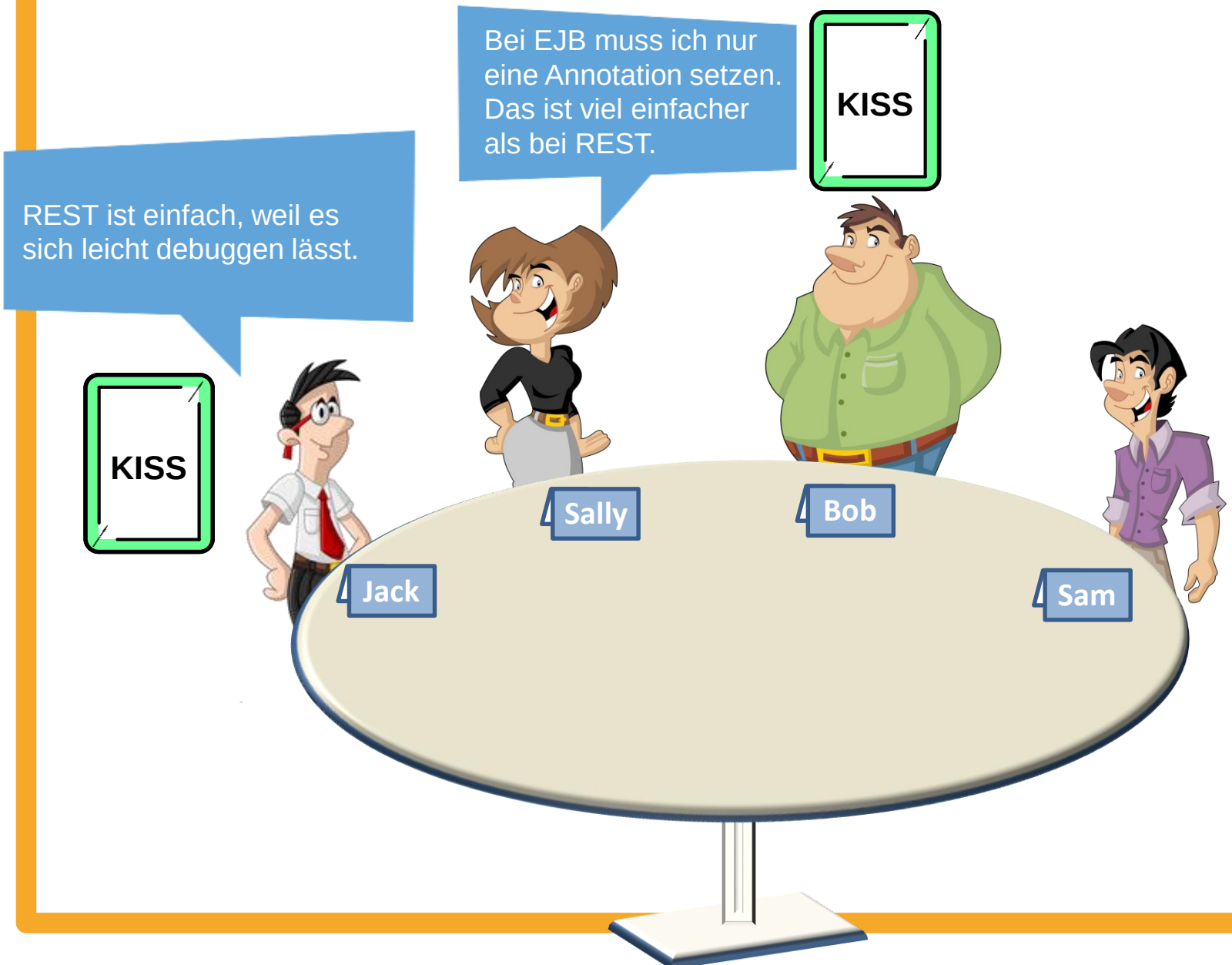


Technische Diskussion: „kartengestützte Argumentation“

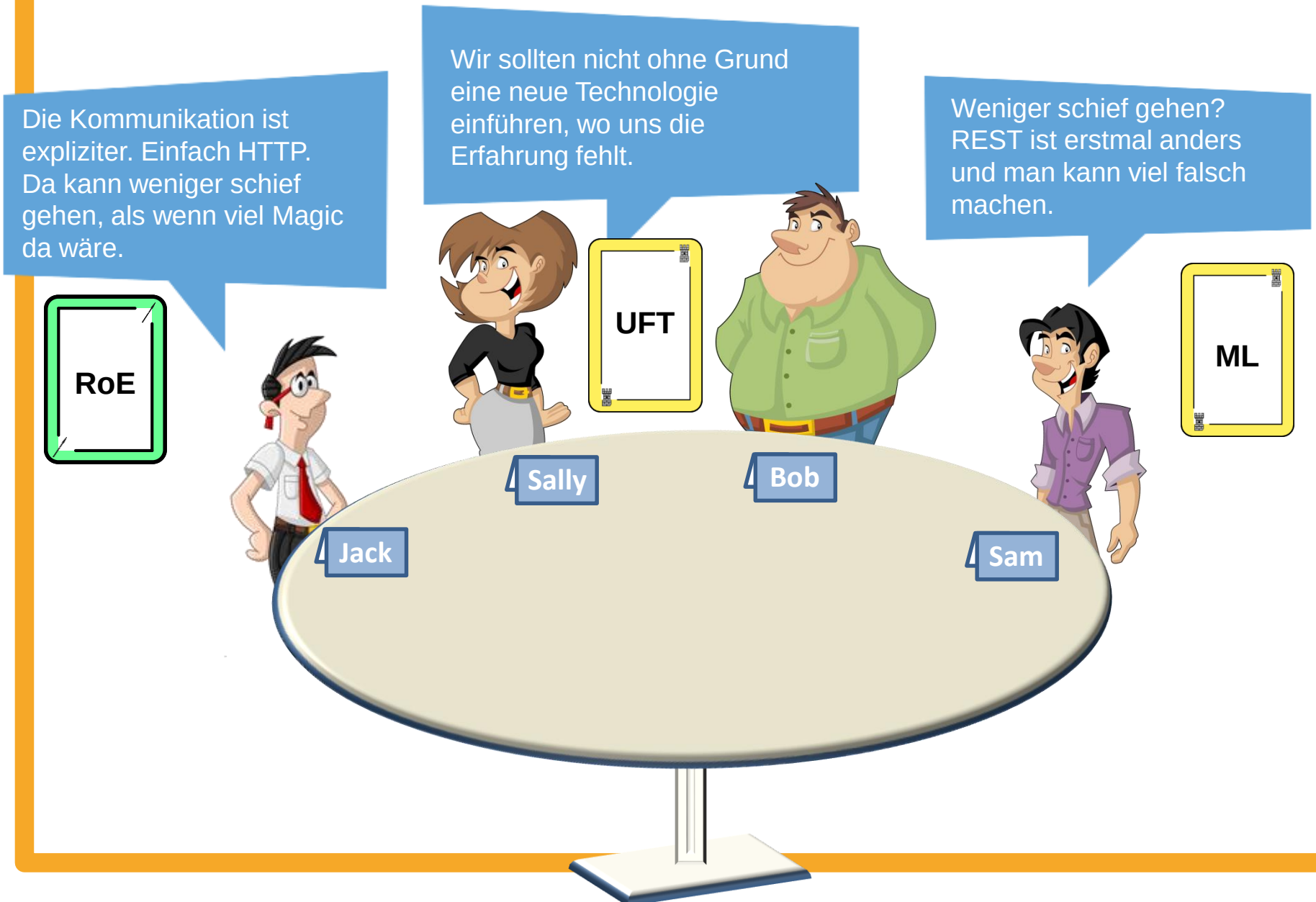
Ziel: Einsatz von nachvollziehbaren Argumenten



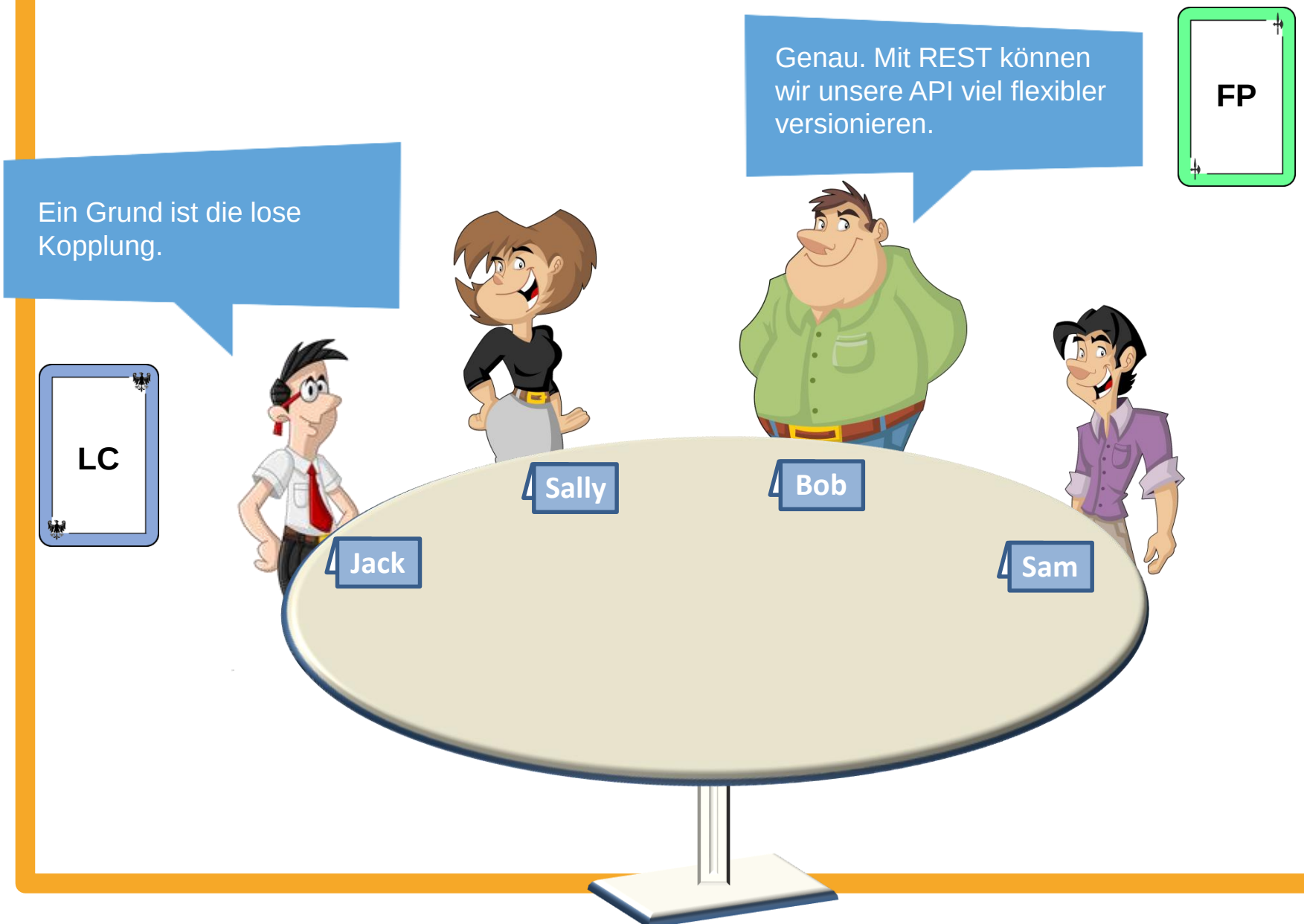
Technische Diskussion: „kartengestützte Argumentation“



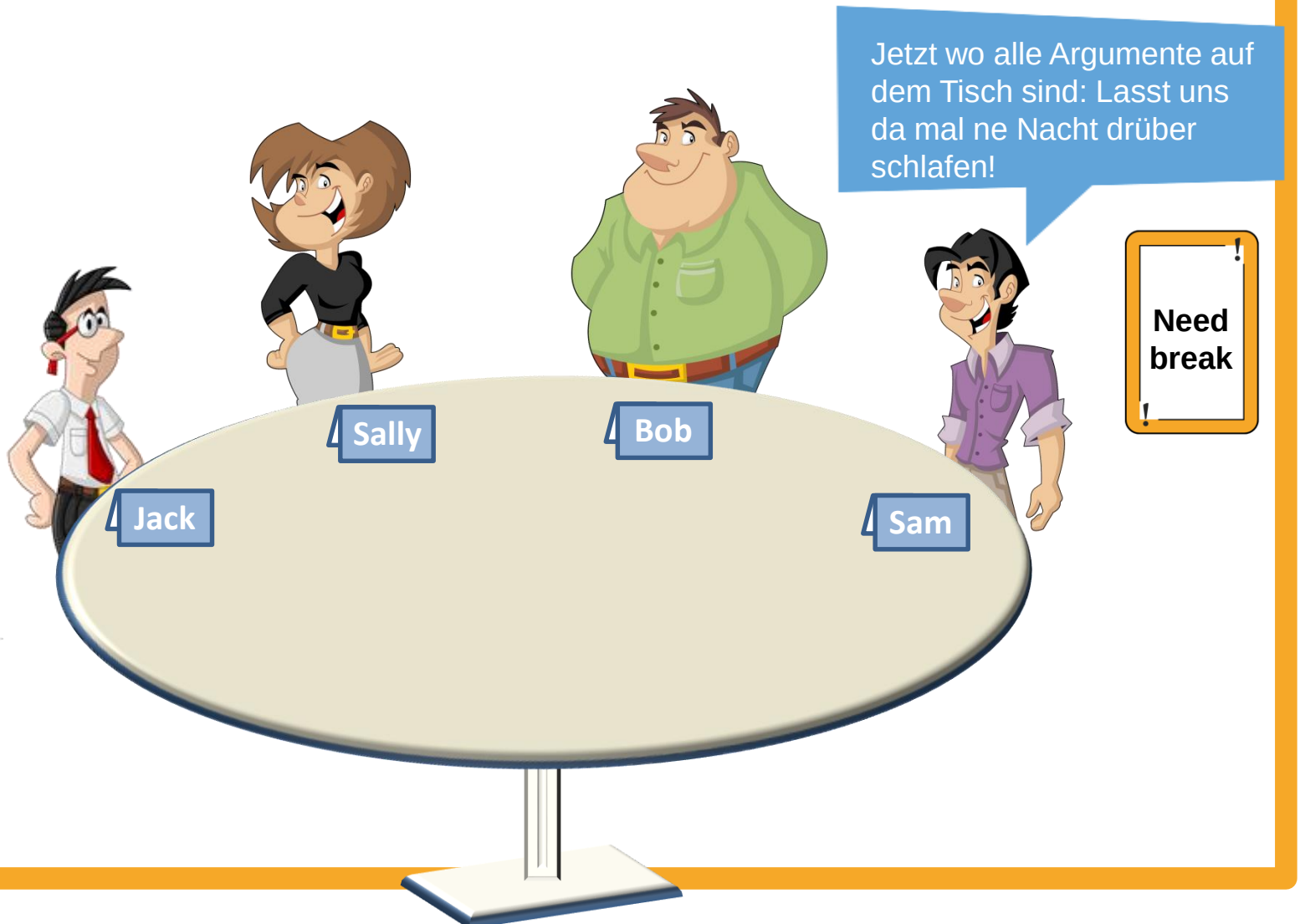
Technische Diskussion: „kartengestützte Argumentation“



Technische Diskussion: „kartengestützte Argumentation“

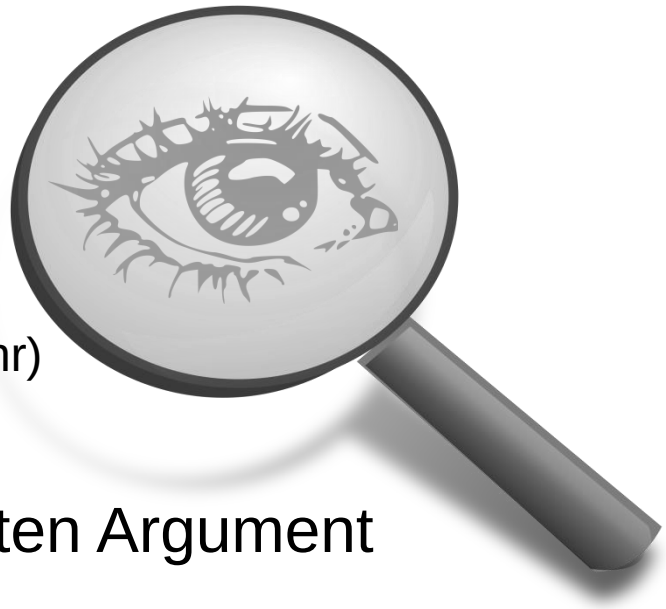


Technische Diskussion: „kartengestützte Argumentation“



Erste Beobachtung

- Die Entwickler argumentieren klarer und nachvollziehbarer
  (Jack überrennt seine Kollegen nicht mehr)



- Ein Argument leitet zum nächsten Argument oder Gegenargument über

Die Karten im Detail

Wir haben noch gar keinen Konsens gefunden!

Die Dimensionen unserer Entwickler-Typologie

Simple	vs.	**Powerful**
Abstract	vs.	**Concrete**
Pragmatic	vs.	**Idealistic**
Robust	vs.	**Technologic**

Die konkreten Ausprägungen

Simple means:
- to keep code simple for better understandability
- to omit unnecessary things (lower risk; fewer bugs)
- to reduce complexity
- to prefer explicit solutions instead of implicit ones
- etc.

Powerful means:
- to build powerful and generalized solutions
- to have flexibility/extensibility by foresighted design
- to have configurable solutions
- to master complexity
- etc.

S

Abstract means:
- to think in concepts and abstractions
- to focus on the big picture and component interactions
- to know all potential consequences of a change
- to build models of the real world
- etc.

Concrete means:
- to think in code or simultaneously transfer ideas into code immediately
- to optimize algortihms for better performance
- to understand systems by reading the code
- etc.

C

Pragmatic means:
- to fulfill requirements asap
- to focus on customer needs to guarantee a value
- to omit unnecessary things
- to bring others down to earth
- etc.

Idealistic means:
- to make things right – not only 80%
- to consider all aspects not only functional ones
- to know that everything has its right place
- to do not misuse existing concepts, APIs, etc.
- etc.

I

Robust means:
- to protect applications against risks and potential bugs
- to define and adhere to standards
- to avoid too much magic and complexity to reduce risks
- to use proven solutions which stood the test of time
- etc.

Technologic means:
- to use new, modern and more productive technologies and to get rid of legacy
- to evolve with technology to be more competitive
- to broaden your personal horizon
- etc.

R

Selbst ausprobieren: www.design-types.net

Ein Beispiel-Ergebnis

SAPR: The Construction Manager

- Simple : This means you prefer s straight-forward solutions
- Abstract : This means you alway the big picture in mind
- Pragmatic : This means you like things done fast
- Robust : This means you strive f and robust software

Description
The Construction Manager loves to work like on a construction site. There is a plan and everybody works hand in hand to reach the aimed goal. He focuses on working solutions that are built on proven technologies. This ensures that the result will stand the test of time. The most matching motto is: Getting things done. He rather implements by himself than choosing the wrong and maybe unstable framework. He knows very well about his abilities and has reservations about foreign technologies that did not proof their maturity over a certain period of time. He also focuses more on the interaction of particular modules instead of having too many sophisticated and complex constructs in his design. He prefers simple craftsmanship which tells him not to finish before a certain level of robustness has been shown by manual or automated tests.

Your designs are
Stable and reasonably planned without unnecessary complexity

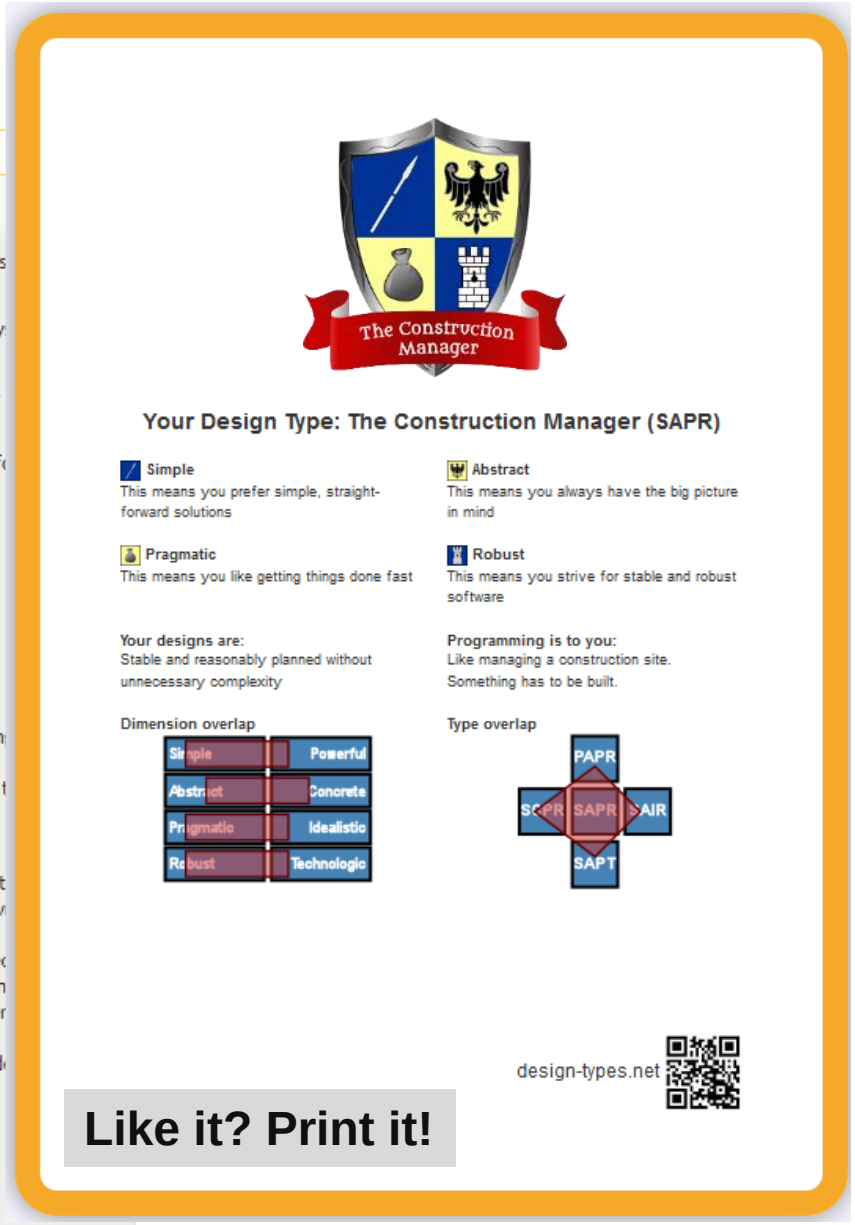
Programming is
Like managing a construction site. Something has to be built.

Principles you probably like
KISS, MIMC, RoE, LC, HC

Principles you rather disregard
GP, PSU, TdA/IE

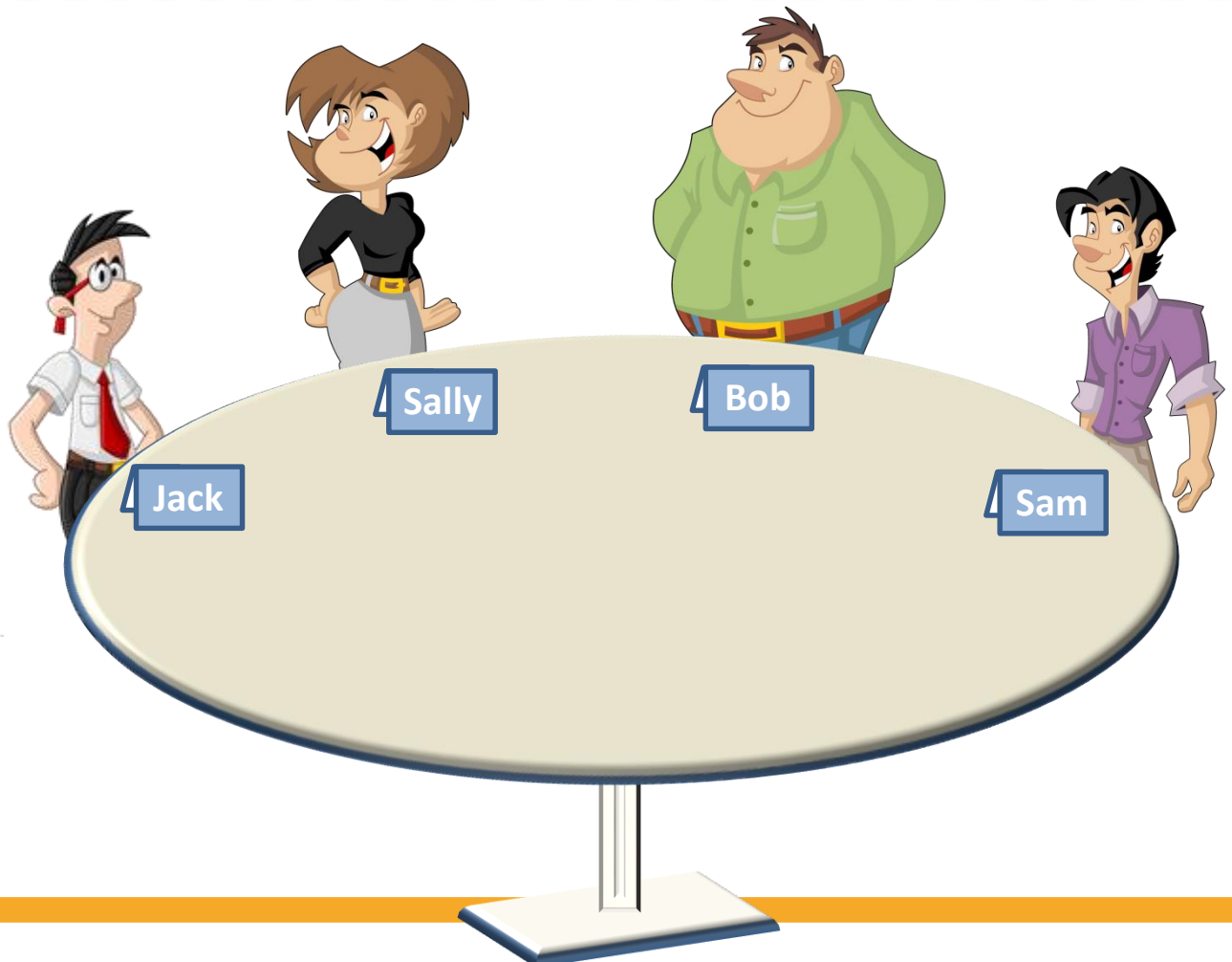
Strengths
- Fast in delivering stable and workin solutions.
- Code and design are normally easy t understand.

Suggestions
- Keep your technical knowledge up t avoid building too much on your ow existing library could do.
- Don't get left behind by evolving te
- Keep your design flexible and exten be prepared for continuous requirer changes.
- Don't forget the trade-offs you mad increasing development speed.

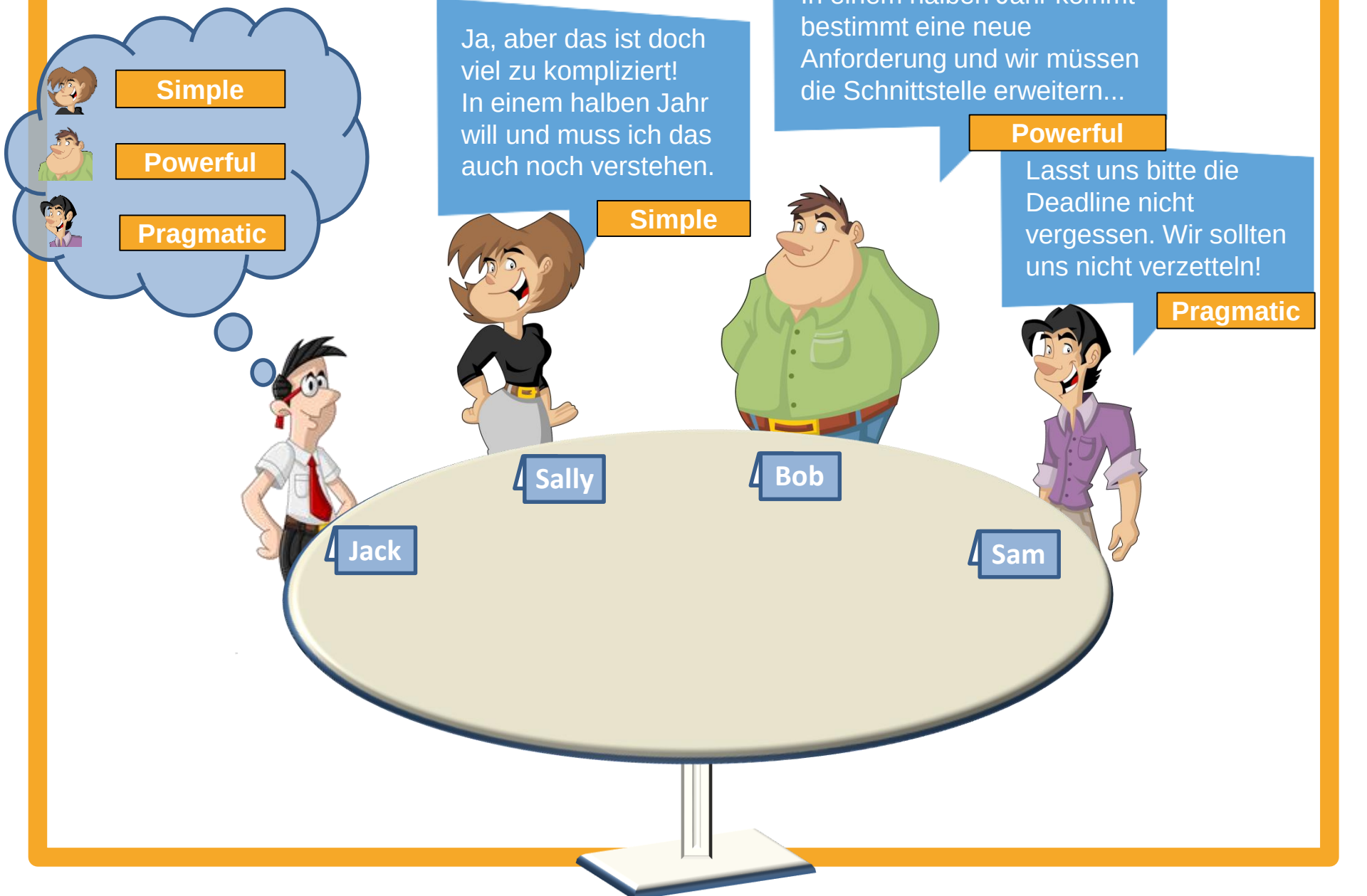


Technische Diskussion: „Diskutieren mit Kontext“

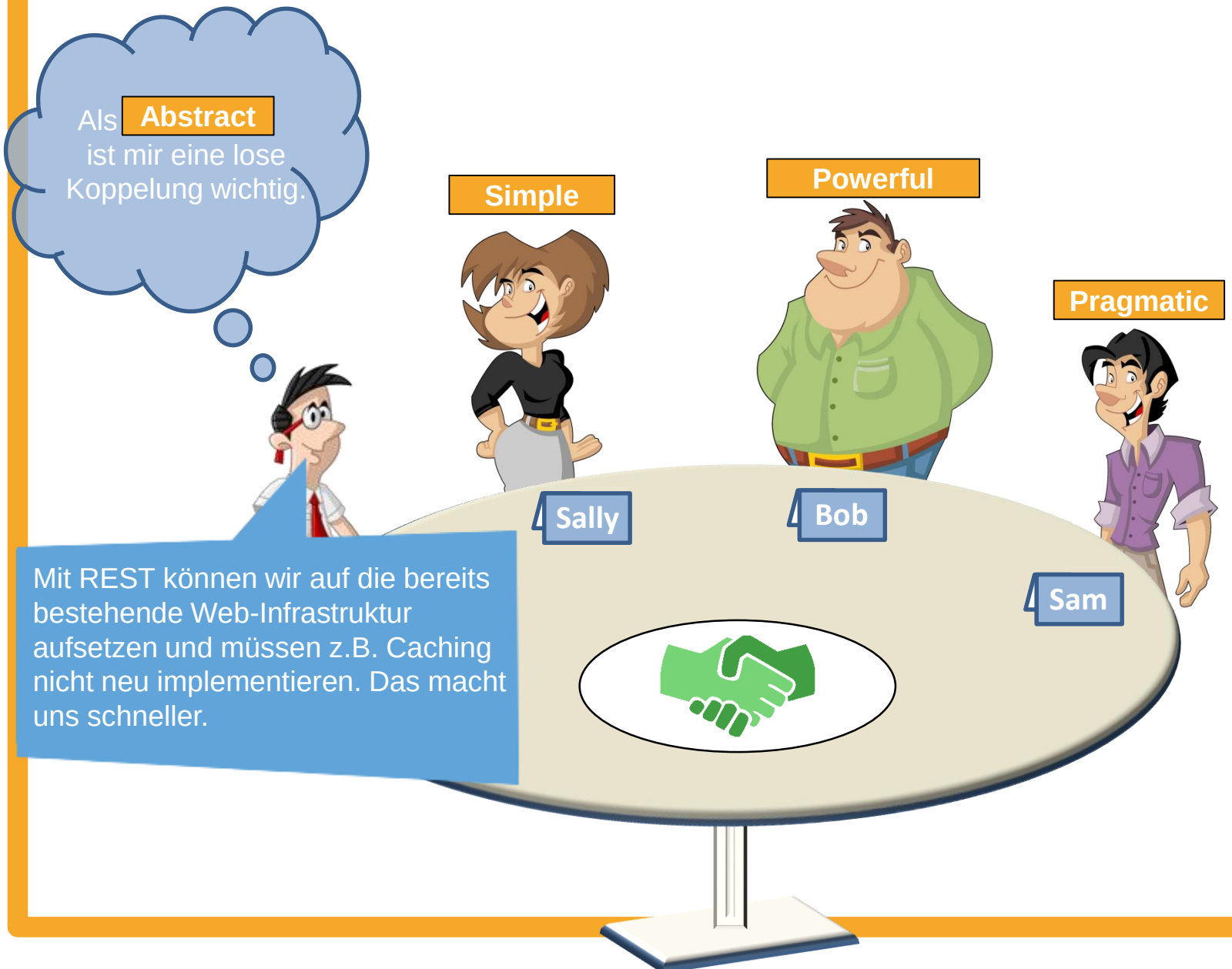
Der nächste Tag…



Technische Diskussion: „Diskutier



Technische Diskussion: „Diskutieren mit Kontext“



Was nehmen wir mit?

- Gute und nachvollziehbare Argumente
  - Design Cards

- Gegenseitiges Verständnis für unterschiedliche Positionen
  - Design Types

- Um Blockaden oder Patt-Situation auflösen zu können, benötigt man Exitstrategien
  - Moderationskarten

Gibt es noch mehr?

Design Matrix

Description of design challenge

Name:

Topic overview & Solution details:

If useful, link relevant documents

Result of design decision

Date: Status: ☐ Approved ☐ Rejected

Decided by:

Stakeholder:

Summary:

Simple

- ☐ Is the solution easy to understand (even in the future)? Is there a solution that is easier?
- ☐ Does it avoid „clever“ magic and overly generic approaches?
- ☐
- ☐

notes

notes

- ☐ Is the solution foresighted enough?
- ☐ Does it take non-functional requirements into account?
- ☐ Is the solution generic and reusable?

Powerful

Ziel: Vorbereitete Design-Entscheidungen aus verschiedenen Perspektiven beleuchten.

Abstract

- ☐
- ☐
- ☐
- ☐ on its own?
- ☐ Are modules cohesive and is coupling low?

notes

notes

- the same breath?
- ☐ Can the solution grow naturally over time? (e.g. allow further changes/refactorings)

Concrete

Pragmatic

- ☐ Does the solution provide value early on?
- ☐ Does the solution really address the customer‘s goals/use cases?
- ☐ Does the solution really fit to the timeline?
- ☐ Can we use already existing Code (snippets, libraries, services)?

notes

notes

- ☐ Is this the right solution?
- ☐ Is it consistent with the rest of the system?
- ☐ Is ensured that there are no workarounds or bad decisions that will produce serious problems later?

Idealistic

Robust

- ☐ Is the solution hard to misuse?
- ☐ Are the chances for something to go wrong minimized?
- ☐ Are standards used and adhered to?
- ☐ Are used technologies/libraries stable?
- ☐ Do all involved people have the necessary knowledge?

notes

notes

- ☐ Is there already an existing technology or library that helps us?
- ☐ Is the solution state-of-the-art?
- ☐ Is the solution a technologic progress?
- ☐ Can we get rid of legacy code?

Technologic

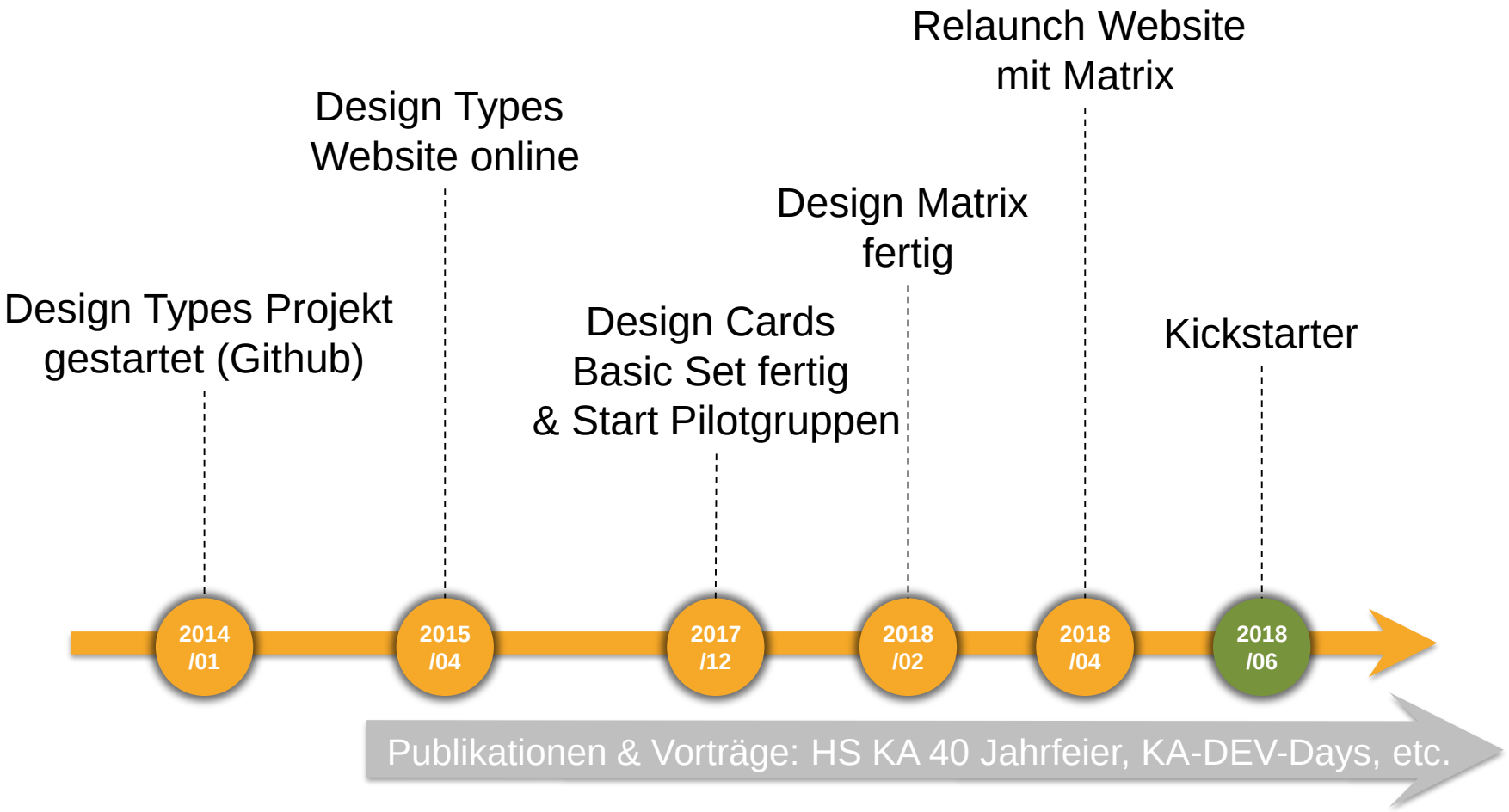
Stand der Dinge

- Design Types
  - ☐ Fertig
  - ☐ > 2600 Teilnehmer bisher

- Design Matrix
  - ☐ Fertig
  - ☐ Aktuell Feedback durch Pilotgruppen
  - ☐ Verfügbar als Download

- Design Cards
  - ☐ 26/54 Karten fertig (Basic Set)
  - ☐ Online-Karten gerade im Entstehen
  - ☐ Aktuell Feedback durch Pilotgruppen

Timeline – was passiert(e)?



Thank you for your interest...
...the „Software Design Knights“

Any Questions?

Contact

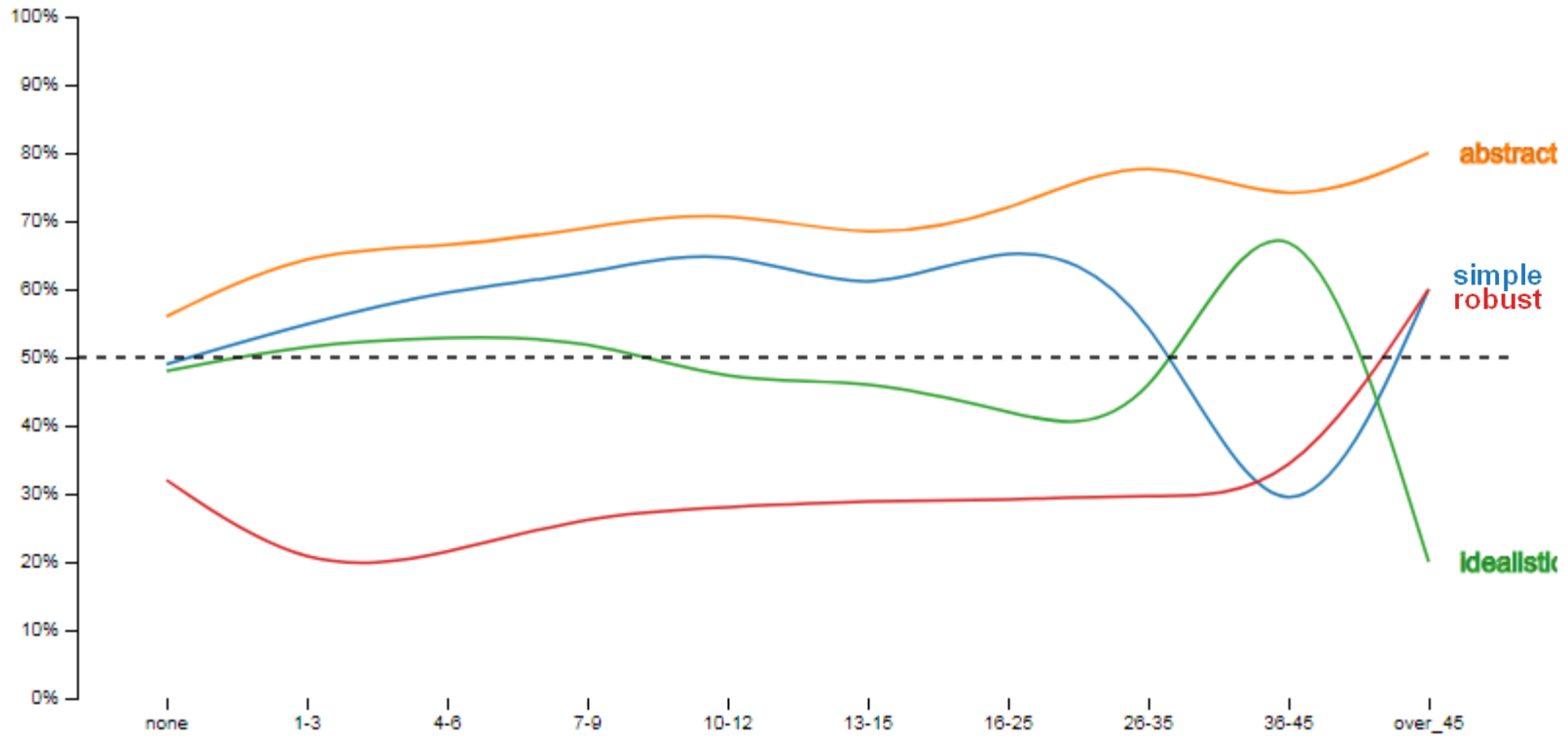
Web: www.design-types.net

Mail: email@design-types.net

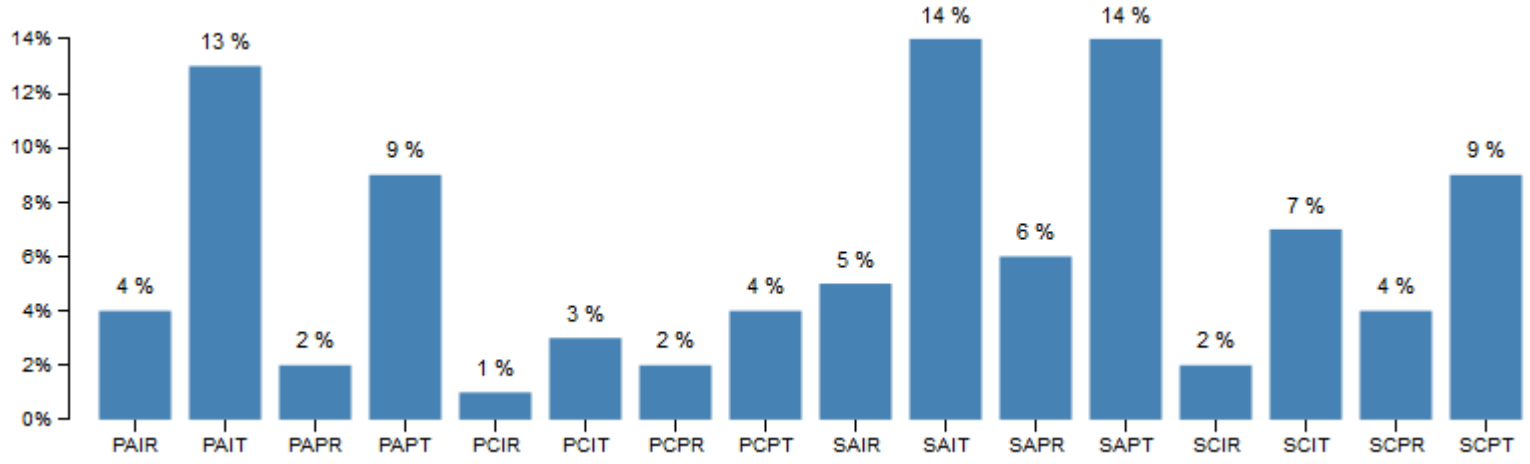
Twitter: @SWDesignKnights

Anhang

Statistiken – Veränderung durch Berufserfahrung



Statistiken – Verteilung der Typen



Statistiken – Häufigkeit der Antworten