Bessere Präsentationen

für Entwickler(innen) und Architekt(innen)

Michael Plöd

Fellow bei INNOQ

150+ Konferenzvorträge

@bitboss

https://speakerdeck.com/mploed

Liebe Software Entwickler(innen), ich will Euch nicht zur dunklen Seite der Macht ziehen!

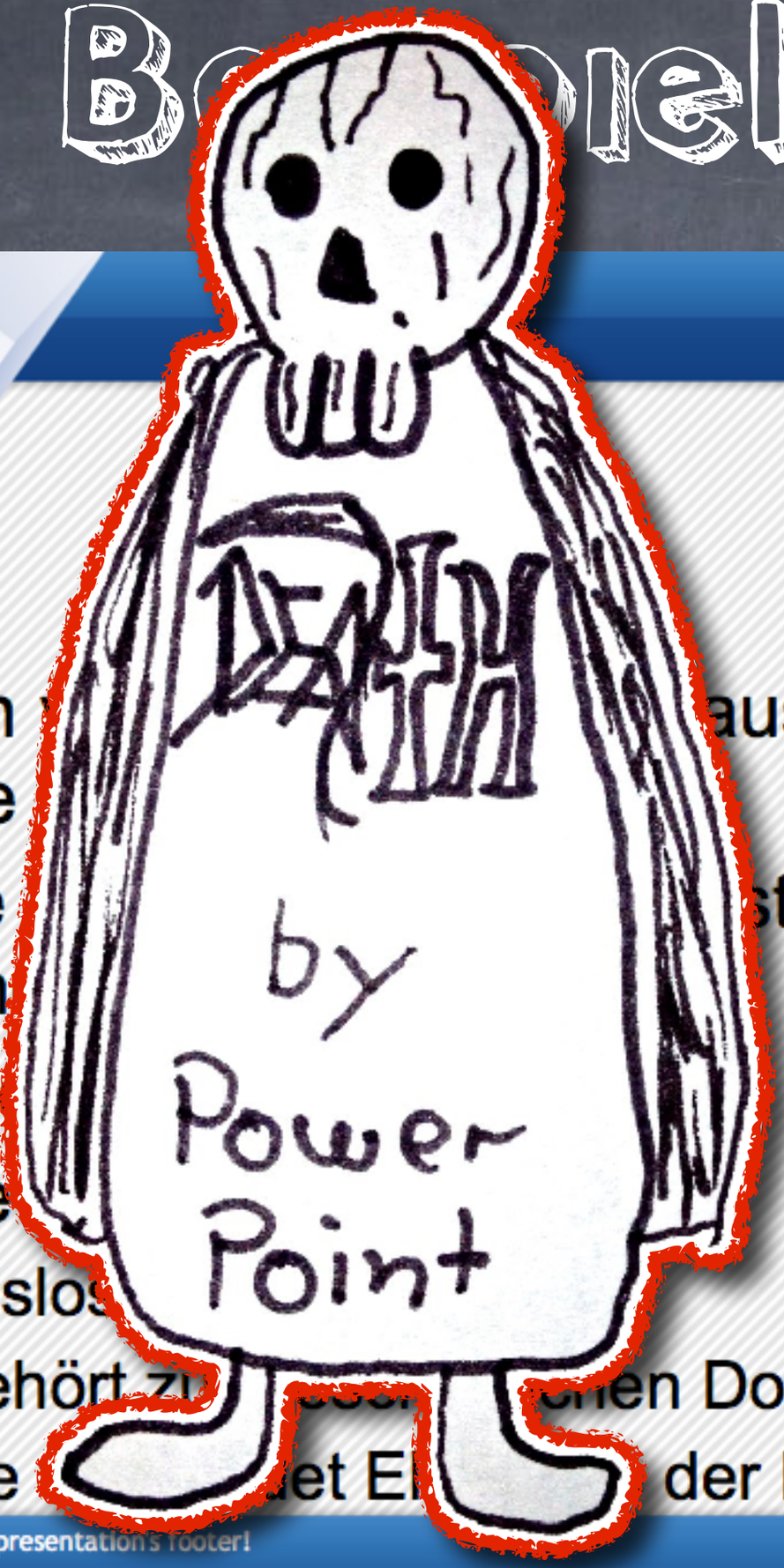
Es gibt zu viele schlechte und langweilige Präsentationen

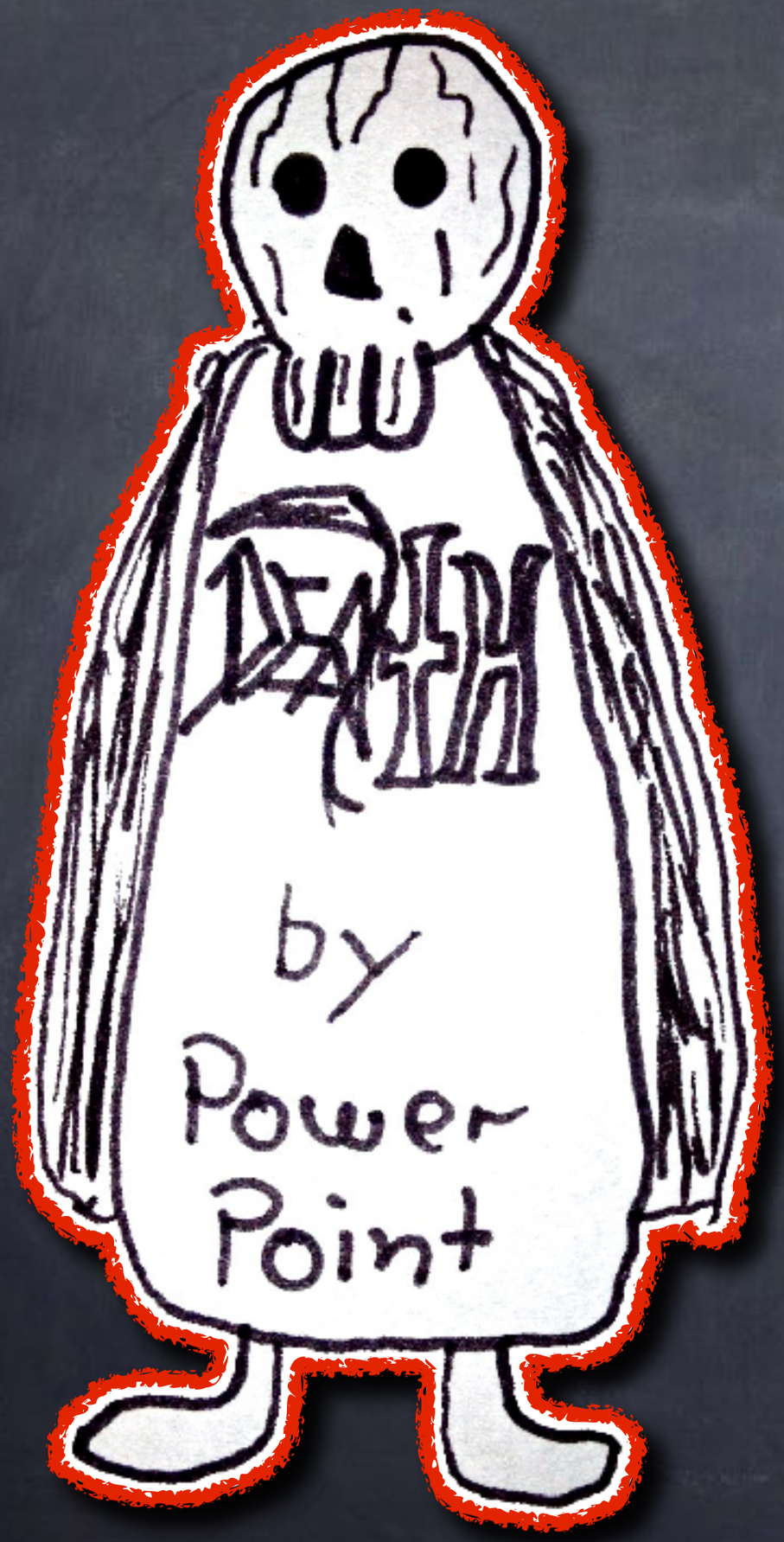
Auch ich habe mit schlechten Präsentationen angefangen

Bagatell

Services

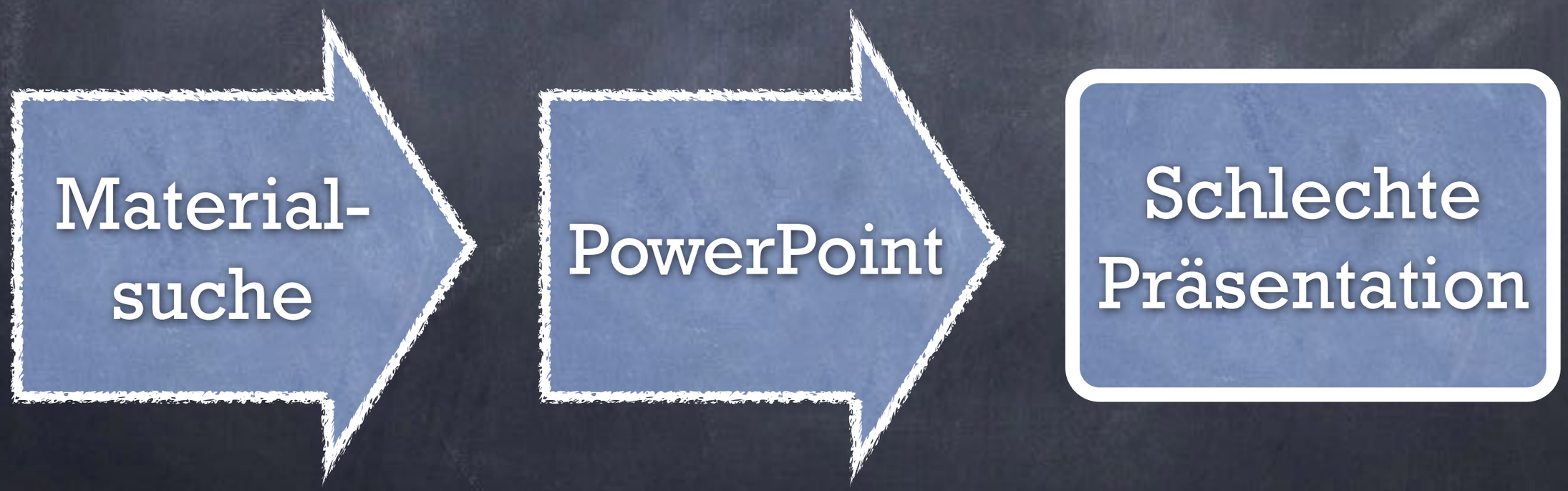
- Extraktion … aus Entitäten und Value …
- Fachliche … standteil eins Domänen …
- Vorteil: M…
- Merkmale …
  - Zustandslos…
  - Logik gehört zu … chen Domäne
  - Interface … et El… der Projektsprache





PowerPoint Skills sind unwichtig

Der übliche Prozess besteht aus 2 Schritten



Gute Präsentation

Design

Argumente

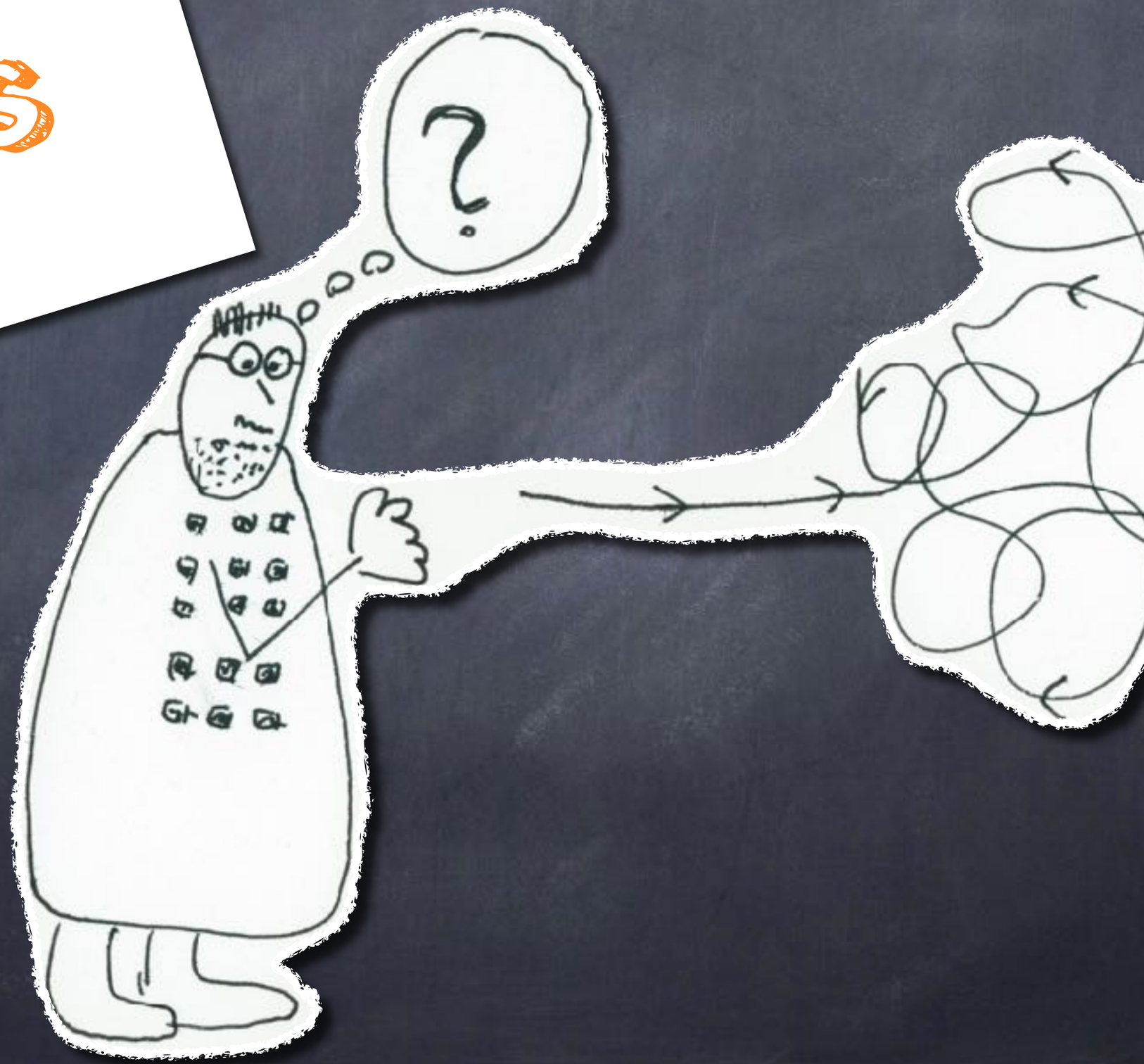
Vortrag

Jede(r) kann gute Präsentationen erstellen, ...

...wenn man einem

einfachen Prozess

folgt...



Fünf

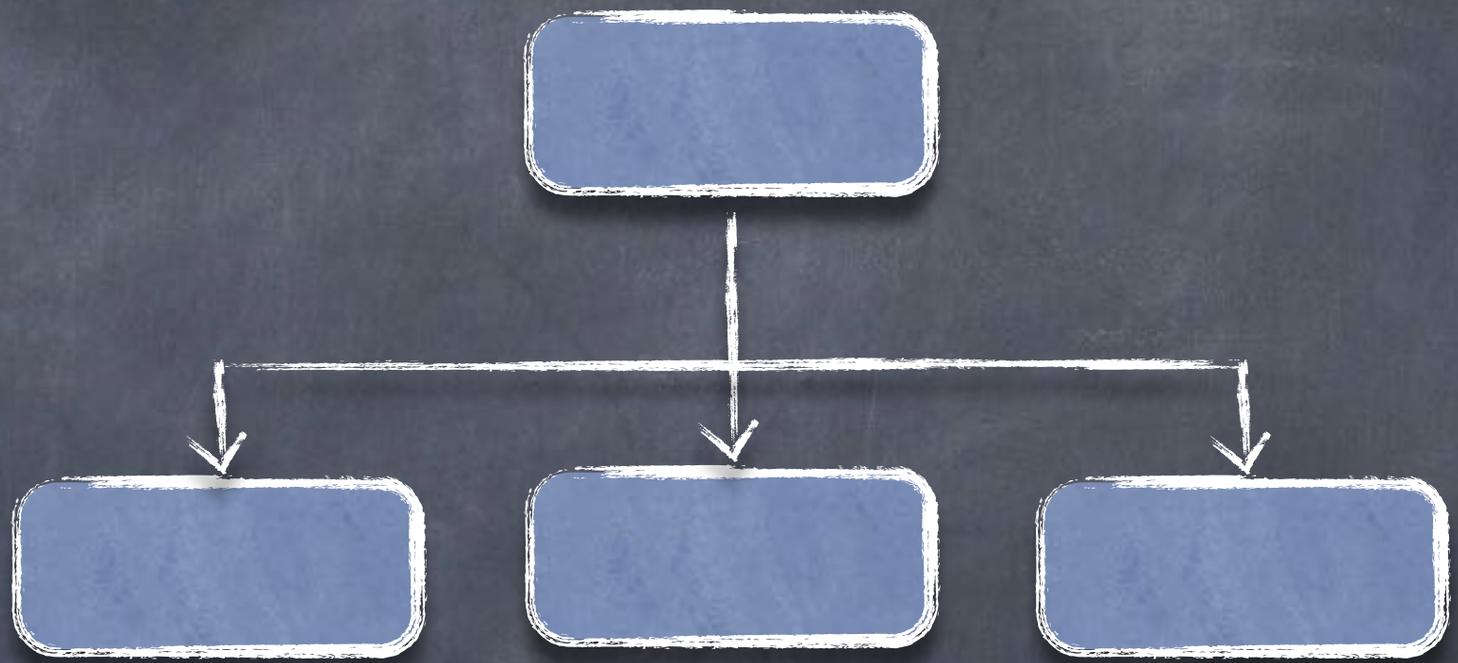
einfache Schritte.....

Schritt 1:

Zielgruppenanalyse

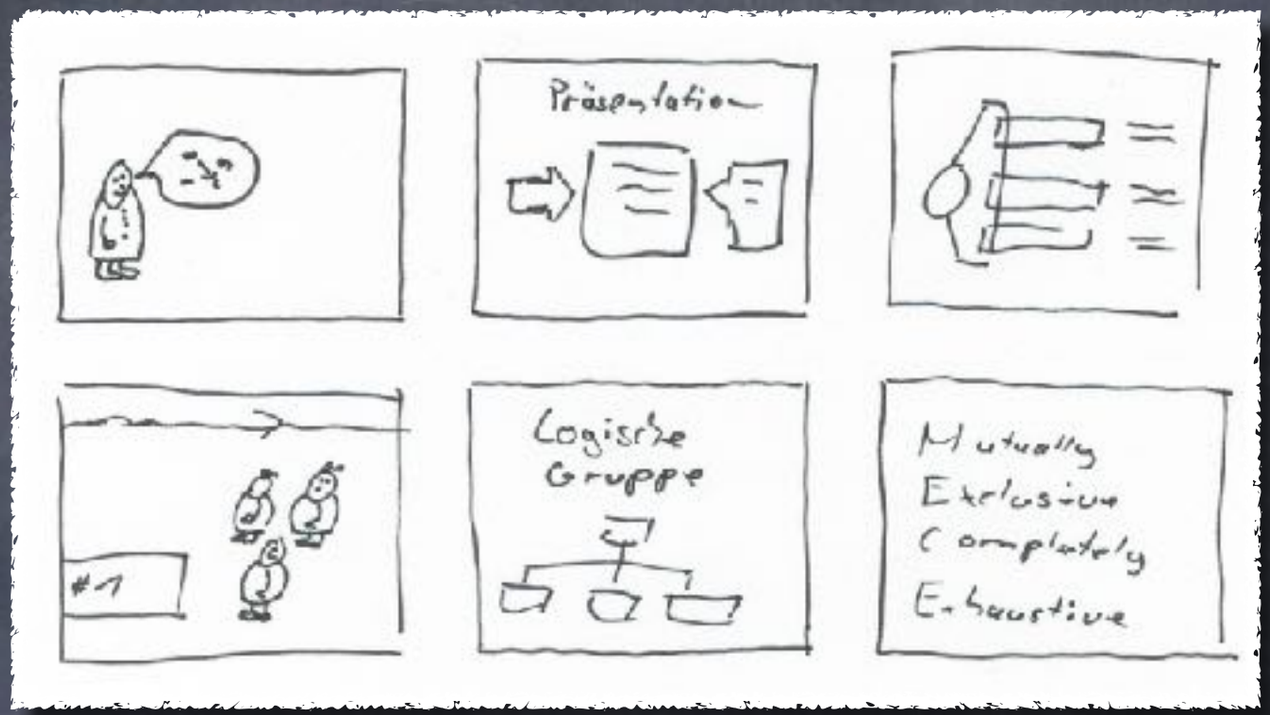
Schritt 2:

Stoff- und Materialsammlung



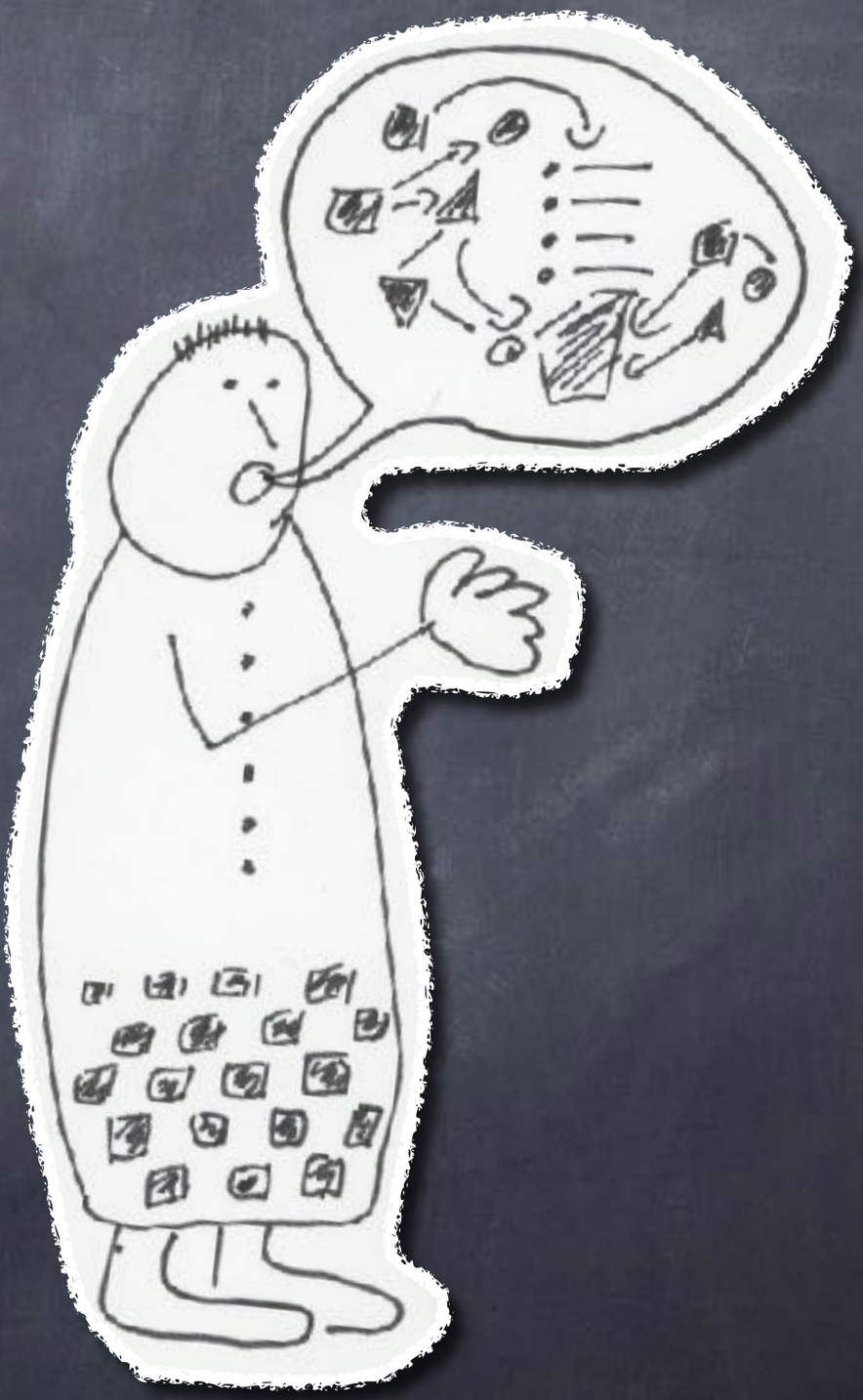
Schritt 3:

Argumentation



Schritt 4:

Layout



Schritt 5:

Vortrag

Wir arbeiten am Anfang

Analog

Schritt 1:

Zielgruppenanalyse

Ihr seid nicht die Helden

Das Publikum ist der Held

einfache Fragen zur Analyse der Zielgruppe

Quelle: Nancy Duarte - Resonate

Was beschäftigt meine Zielgruppe?

Warum sind sie hier?

Wie kann ich ihr Problem lösen?

Welchen Widerstand wird es geben?

Wie ist das Publikum?

Wie kann ich sie am besten erreichen?

Was soll meine Zielgruppe tun?

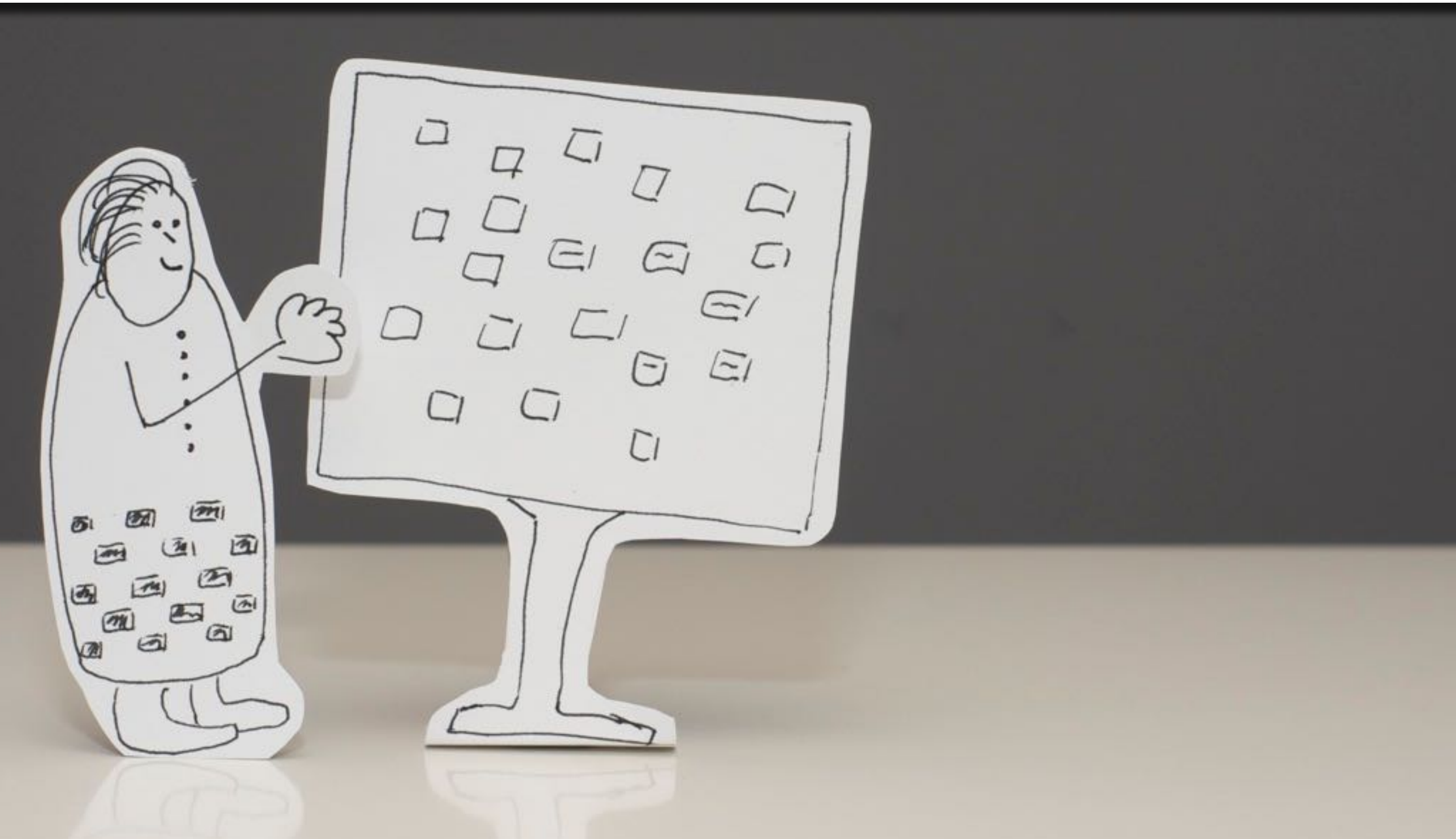
Auch eine

Analyse des Umfelds

ist nötig

Schritt 2:

Materialsammlung



Sammlung von Ideen mit Stift und Papier

OFFLINE

Ideen

MECE

Logische Gruppe

Logische Kette

Storyboarding

(Unterschied Rot)

Big Idea

Farben Lehre

Goldene Schnitt

Layout

Zielgruppe Analyse

Type Graphic

Animierter Start / Schluss in Report einbauen (Quarto)

Go Analog!

Intro: Was Interessiert Manager bei Technologie

Wie verkaufe ich eine gute Technologie an meinen Chef?

Aufgaben bei Vorbereitung meiner

Prozess

Themen

Layout

Grafische Gestaltung

Grund Elemente der Layout

Ebenen der Kommunikation

Unterschied Report / Präsentation / Story

Abgrenzung von Slideology zu Presentation Zen

Equipment
-> Presenter
-> Moderation-Karten
-> Flip Chart

Übersichtliche Folie

"Bar-Chart and Bullet-Point Bonanza"
an Book of the myths

Thesis-Vorlage

Spanning-Poster

Storyboarding (Film / Präsentation Zen)

Stand Idee: Poster

SQVID

6 W: who / where / ...

Grafische Mittel an Book of Myths

SQVID = Zielgruppen

Research: Lesen, Sprache, Gestaltungsmöglichkeit

Folien templates

Präsentieren: Rede

Präsentieren: Präsentieren

Handout vs Präsentation Applikation

Beispiel:

Ideensammlung

Sammeln, bewerten, auswählen

Subselect Fetching

Log Files bzgl. Queries

N+1 Problem

Session und 1st Level Cache

Lazy Loading

Query Cache

Batch Fetching

Bind Variablen Ignorieren

2nd Level Cache

Eager Fetching

Zuerst:

Unstrukturiert und offen sammeln

Hibernate Statistics

Karthesi-sches Produkt

Fetching Strategien

- Subselect Fetching
- Eager Fetching
- Batch Fetching
- Lazy Loading

Probleme

- N+1 Problem
- Bind Variablen Ignorieren
- Karthesi-sches Produkt

Analyse

- Log Files bzgl. Queries
- Hibernate Statistics

Danach:

Gruppieren

Caching

- Session und 1st Level Cache
- 2nd Level Cache
- Query Cache

Zuletzt:

Botschaften

Es gibt 2 Haupt-Probleme

N+1 Problem

Karthesi-sches Produkt

Es gibt 2 Ansatz-punkte für Tuning

Fetching Strategien

Subselect Fetching

Eager Fetching

Batch Fetching

Lazy Loading

Caching

Session und 1st Level Cache

2nd Level Cache

Query Cache

Analyse Tools

Log Files bzgl. Queries

Hibernate Statistics

Schritt 3:

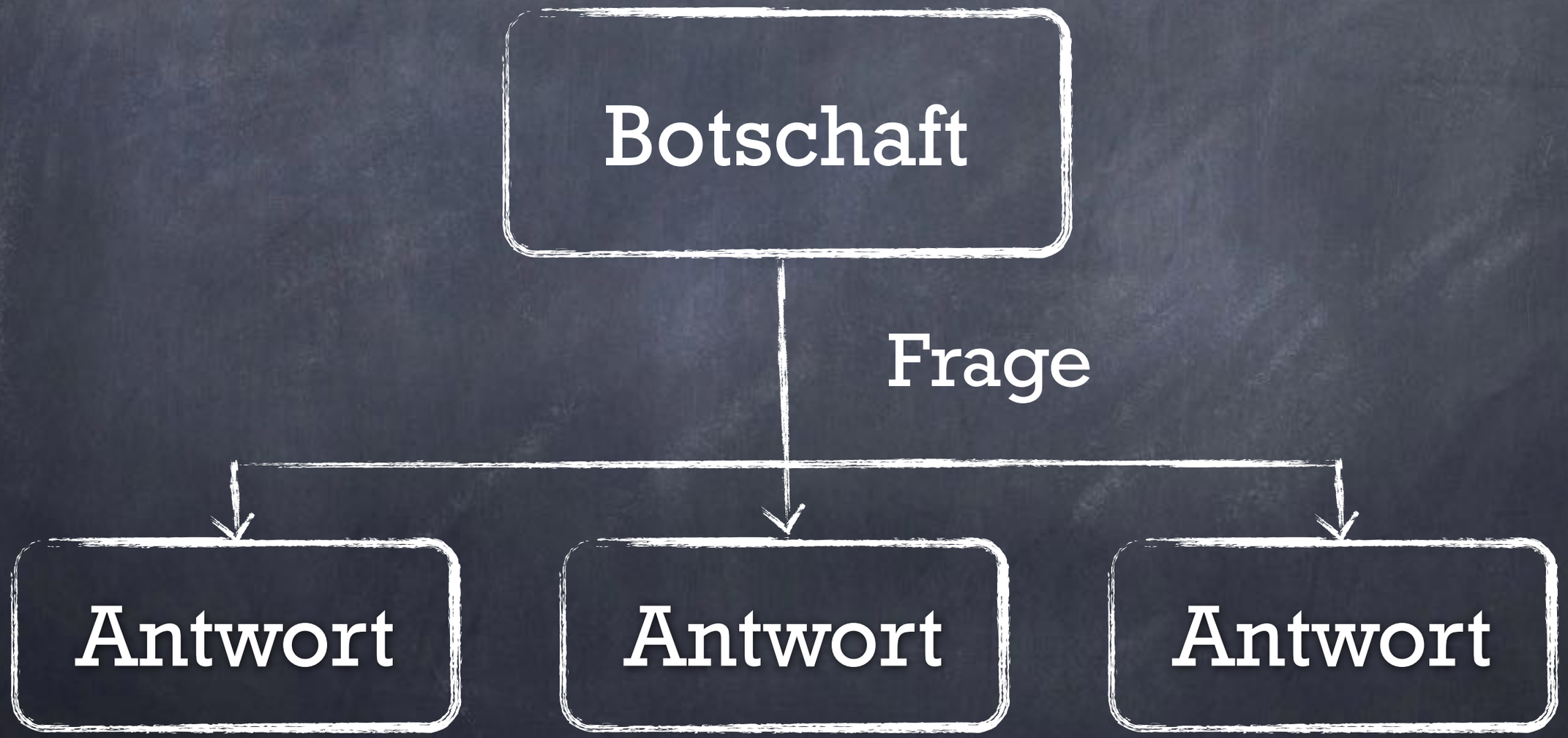
Argumentation + Story

Jede Präsentation hat eine

Hauptbotschaft

Sie ist das Minimum, was die Zuhörer mitnehmen sollten

Logische Gruppe



Logische Gruppe

Jeder kann gute Präsentationen erstellen

warum?

- Prozess
- Argumentations-techniken
- Design Grundlagen

Mutually

Exclusive

Completely

Exhaustive

ME

Aussagen überschneiden sich nicht hinsichtlich ihrer Inhalte

CE

Antworten reichen aus, um Botschaft unangreifbar zu machen

Logische Kette

Botschaft

warum?

Neutrale Aussage (Situation)

+

Kommentierende Aussage (Complication)

→

Folgerung aus Situation und Complication

Logische Kette

Der Test von Projekt X soll zwei Wochen ausgesetzt werden

warum?

Es gibt im aktuellen Test Release zahlreiche Fehler

+

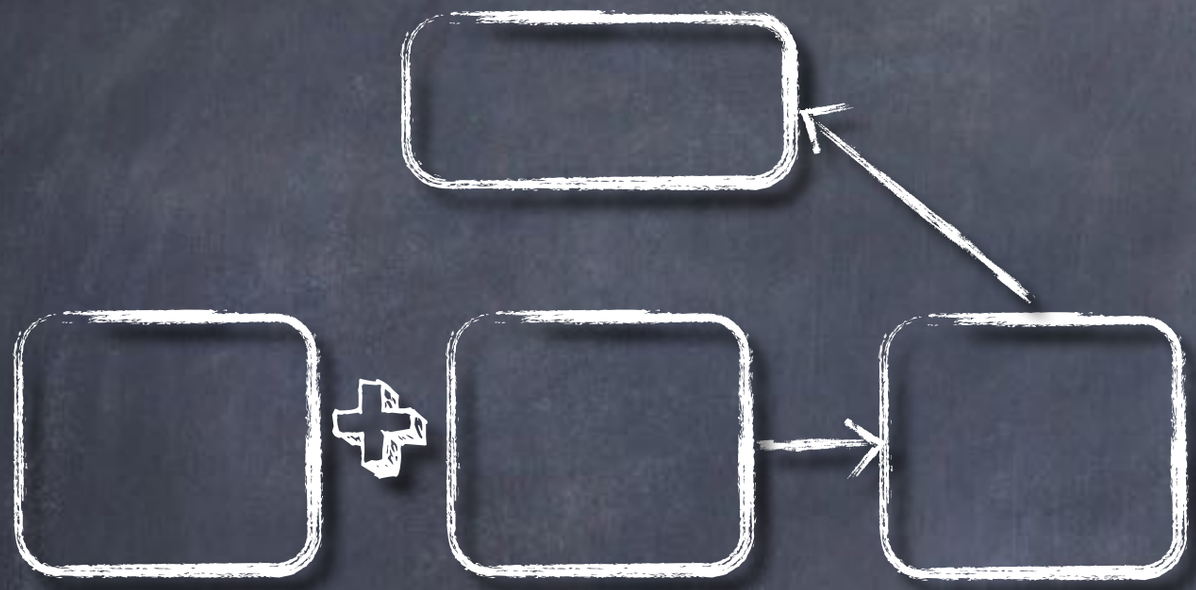
Die Fehler verhindern eine reibungslose Test-Durchführung

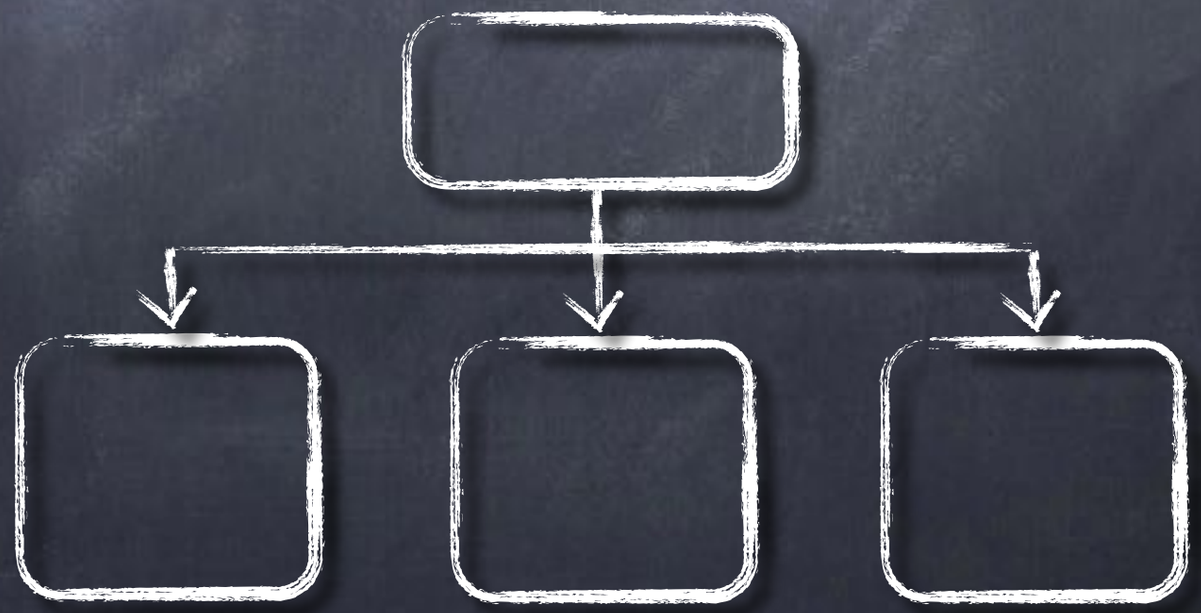
→

Deshalb sollte Test ausgesetzt werden um Entwicklern die Möglichkeit zu geben die Fehler auszubessern

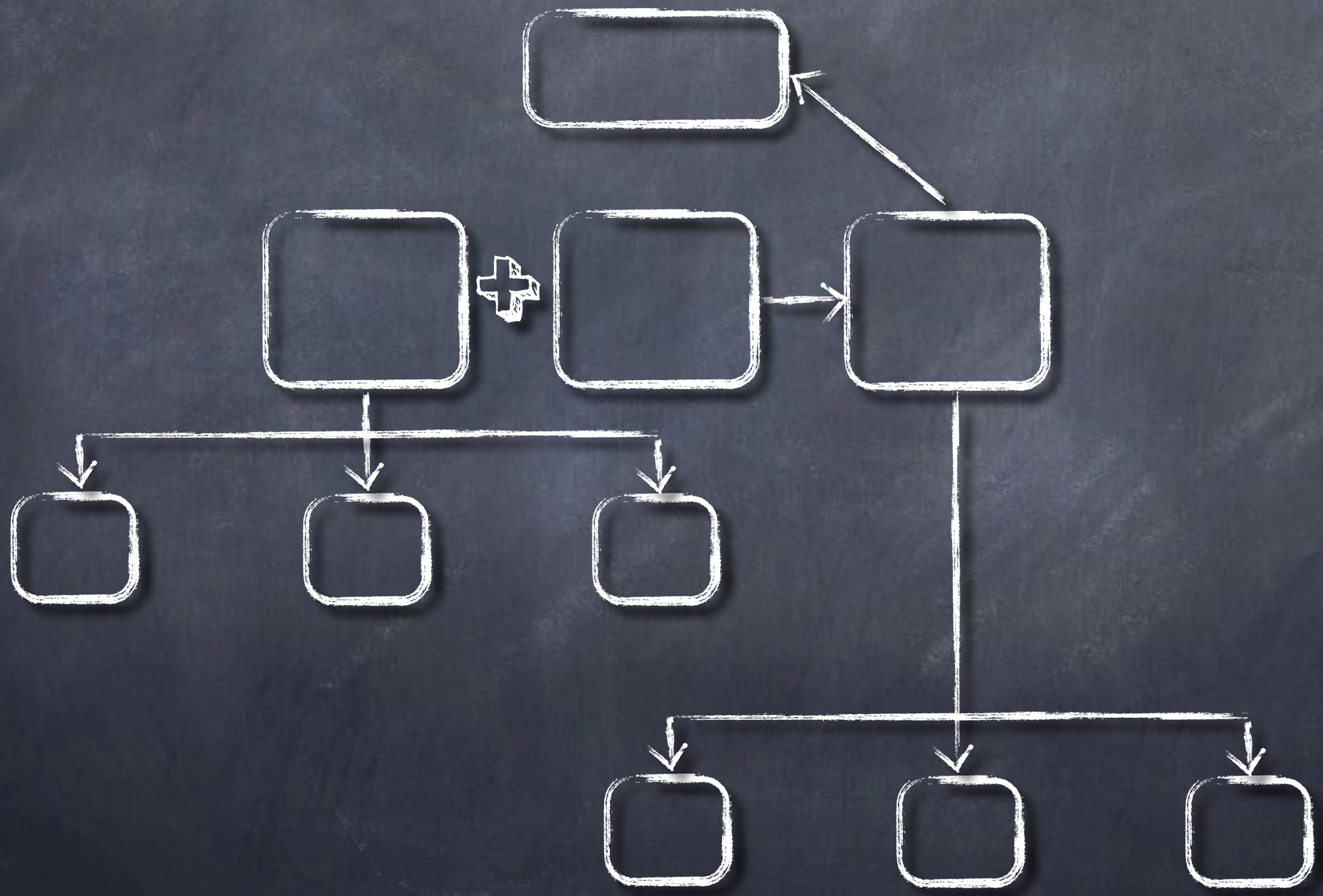
Logische Kette und logische Gruppe können

kombiniert werden

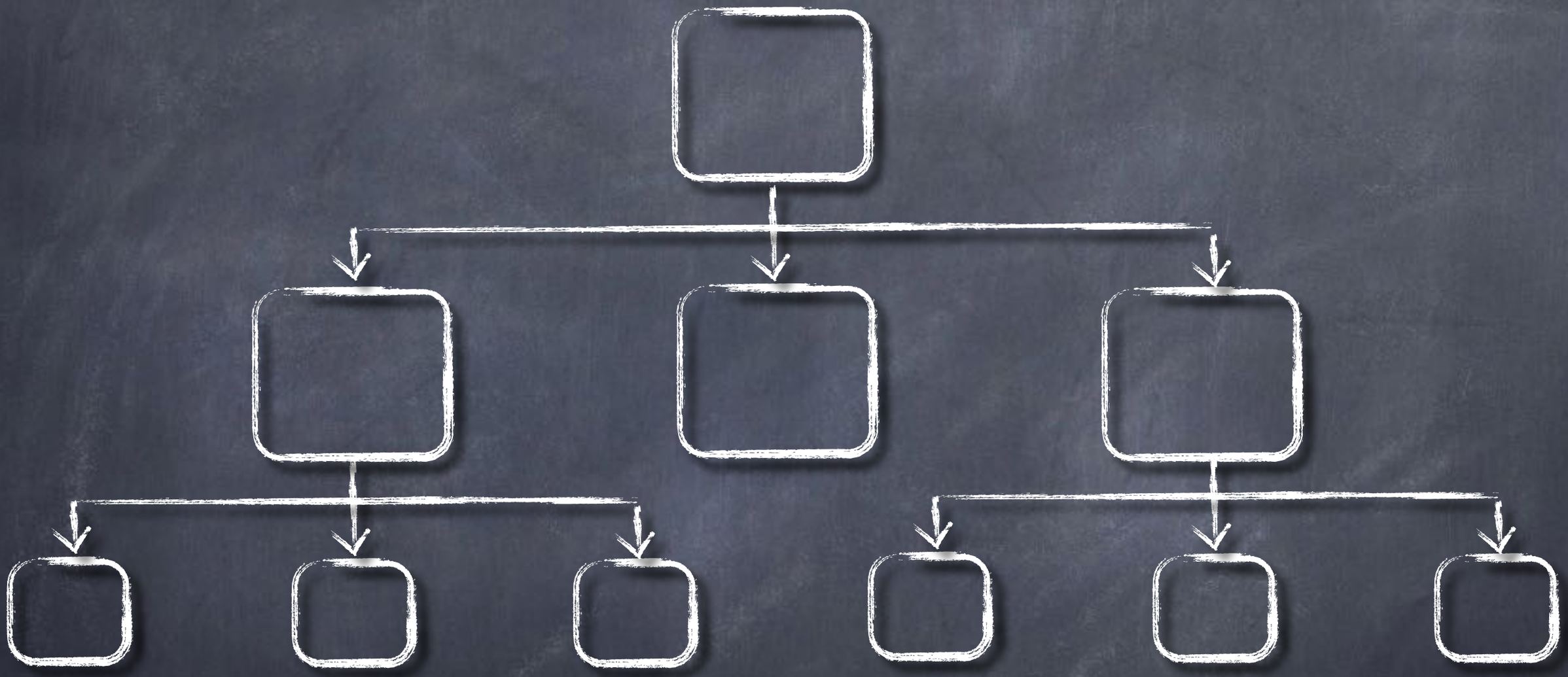




Kette untermauert mit Gruppen



Gruppe untermauert mit Gruppen



Was haben sie gemeinsam?

Eine Story

die unterschiedliche Zielgruppen anspricht

Ethos

Pathos

Logos

Auf Basis unserer Argumentationsketten
erstellen wir ein

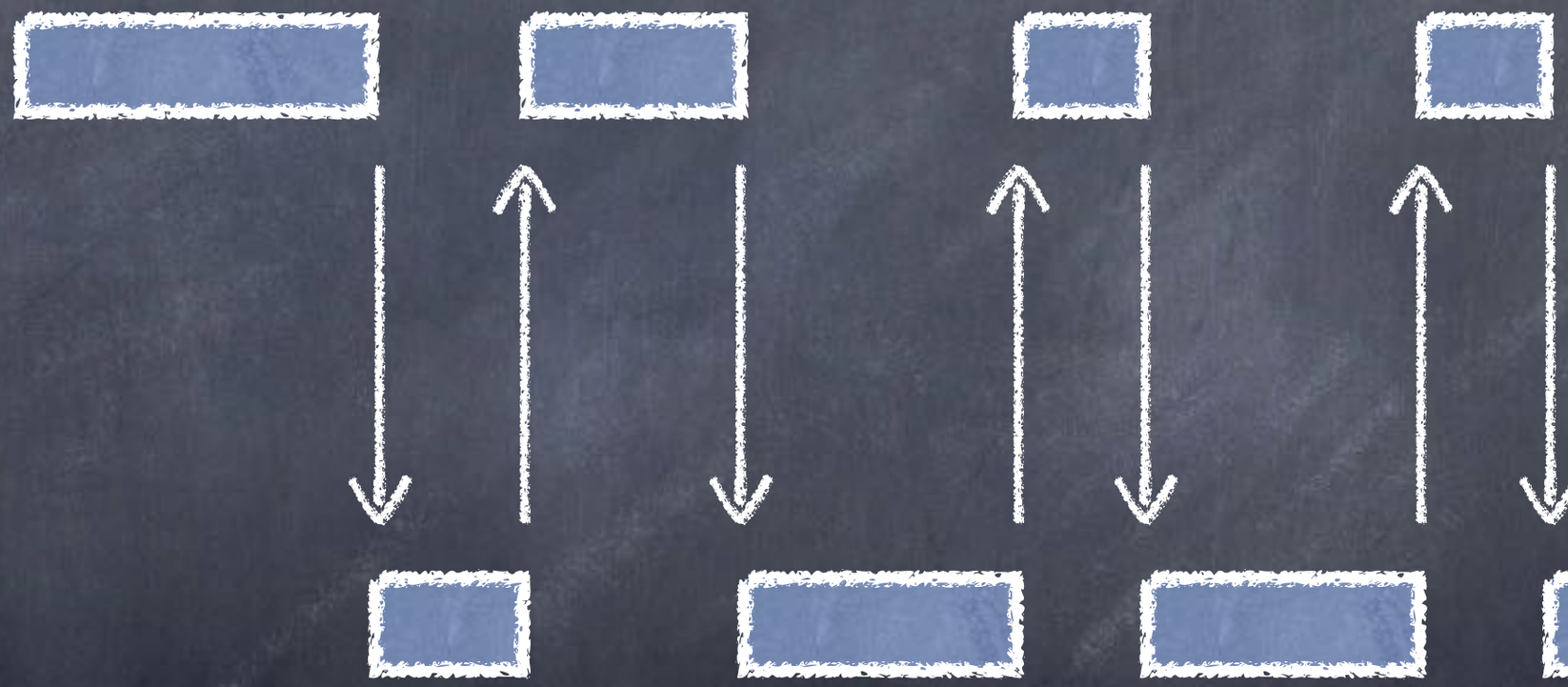
Storyboard

Die

Sparkline

hilft uns bei der Strukturierung der Story

Was ist?



Was kann sein?

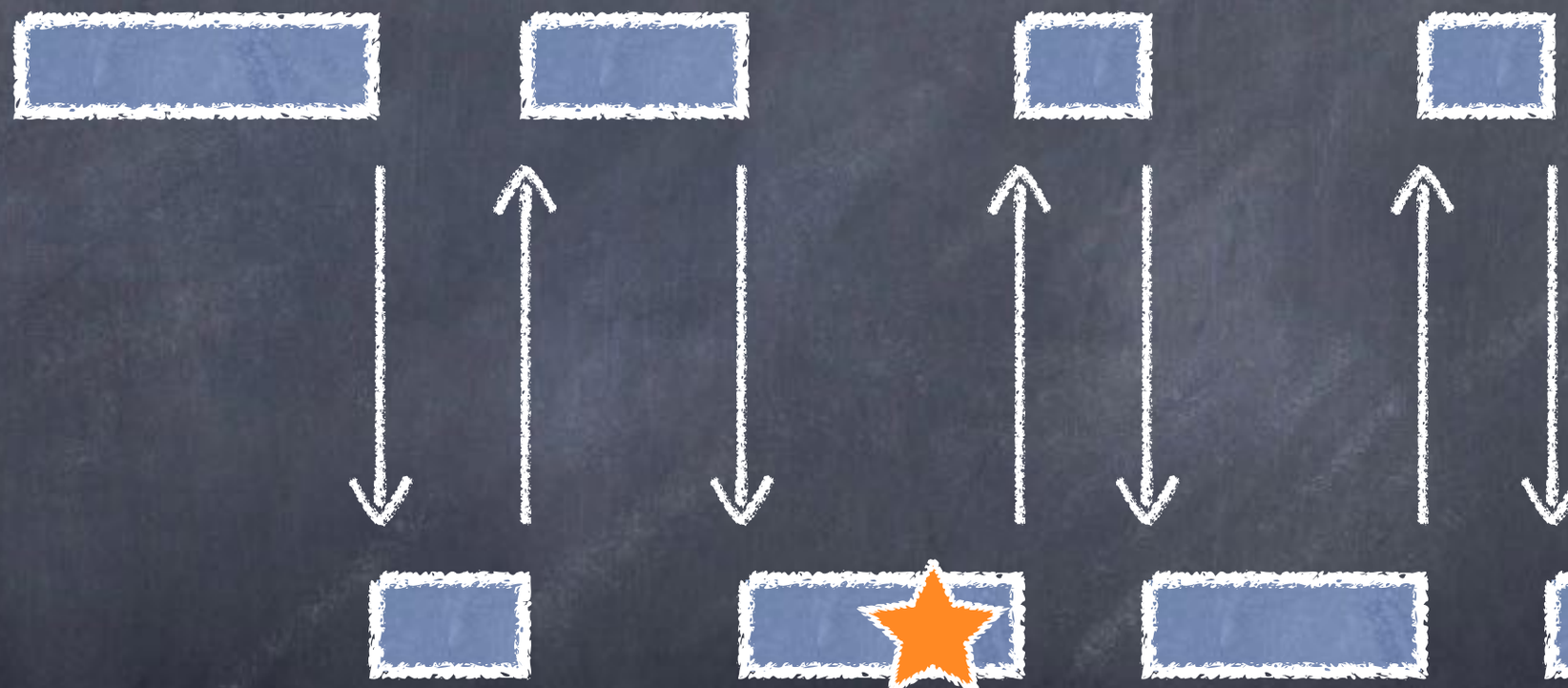
Quelle: Nancy Duarte - Resonate

Der

STAR-MOMENT

ist das Highlight der Präsentation

Was ist?

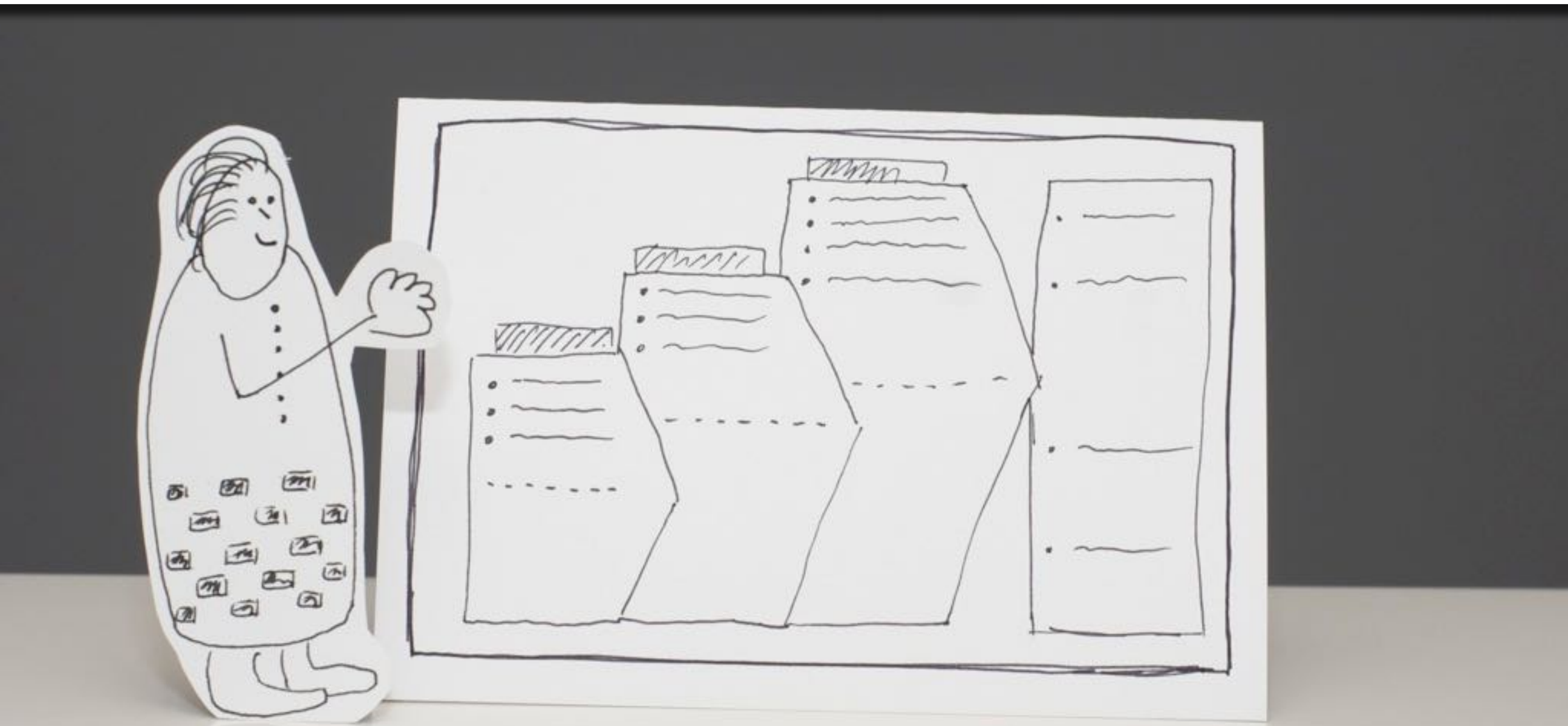


Was kann sein?

Quelle: Nancy Duarte - Resonate

Schritt 4:

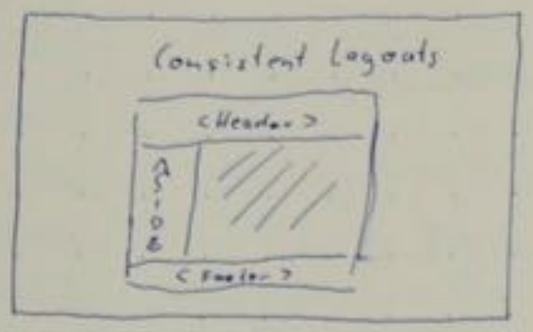
Layout + Design



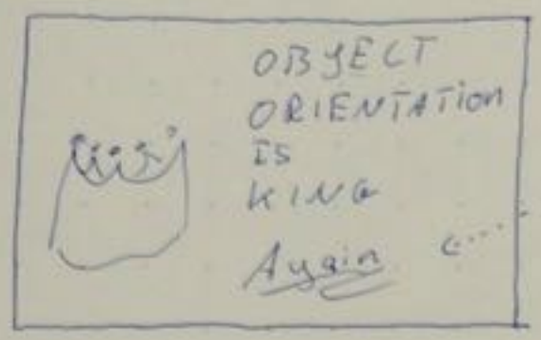
Ursprungslayout

von Ideen mit Stift und Papier

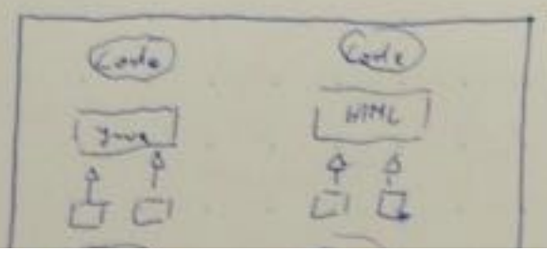
OFFLINE



- Screenshot von Seite
- Überführung in Vorlage
- Wicket - Full OO



- Bild Krone
- Animation für
- Again



- Code Beispiele
- Oben: Base classes
- Unten: Child classes

Beispiel:

Layout

einfache Regeln für die Gestaltung von Folien

Eine Botschaft pro Folie

IMMER

BIND VARIABLEN

verwenden

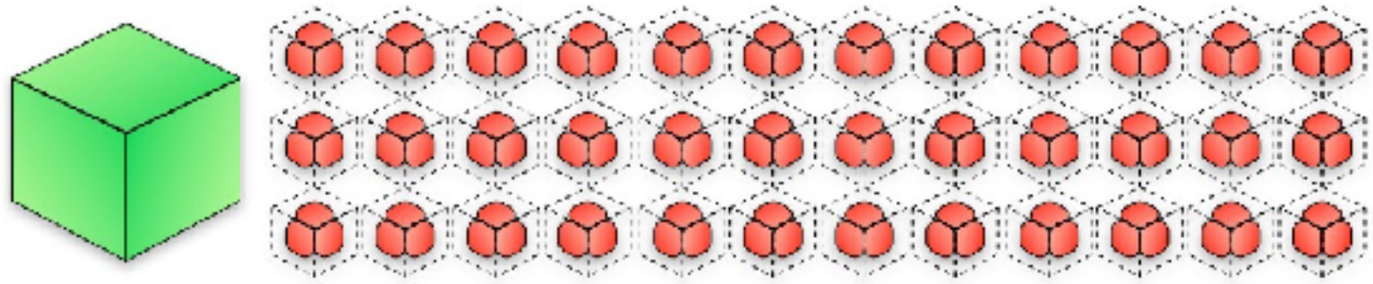
```
Query query =
session.createQuery("from User u where u.name= :name");
q.setString("name", "michael");
```

Einfache Folien

IST ORM **LANGSAM**

Schau-bilder anstelle von Worten

N+1 Selects Problem

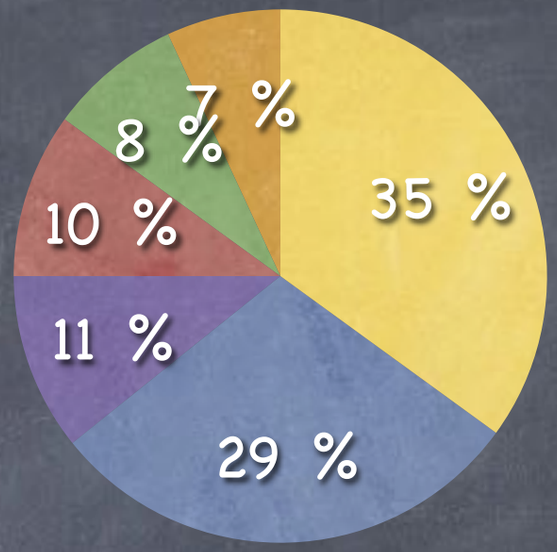


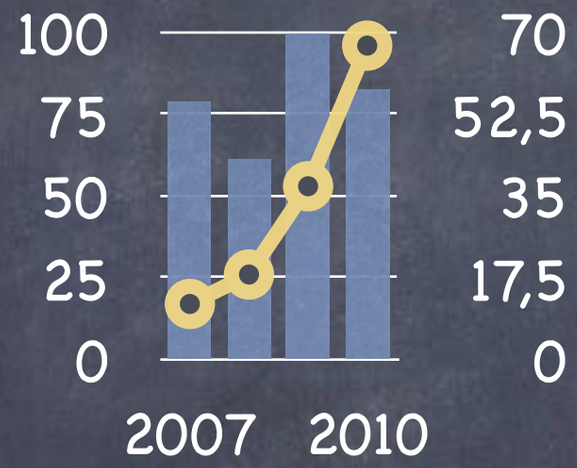
```
List list = s.createCriteria(Konto.class).list();
for (Iterator it = list.iterator(); it.hasNext();) {
  Konto kto = (Konto) it.next();
  kto.getKunde().getName();
}
```

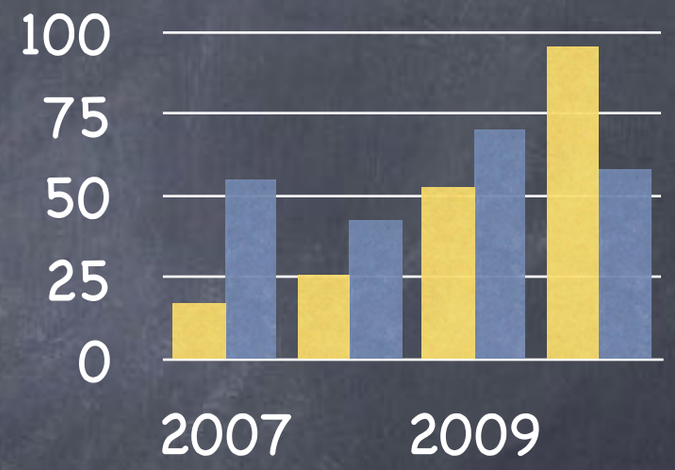
```
SELECT * FROM KONTEN                          +1
SELECT * FROM PERSONEN WHERE PERSON_ID = ?
SELECT * FROM PERSONEN WHERE PERSON_ID = ?    N
SELECT * FROM PERSONEN WHERE PERSON_ID = ?
...
```

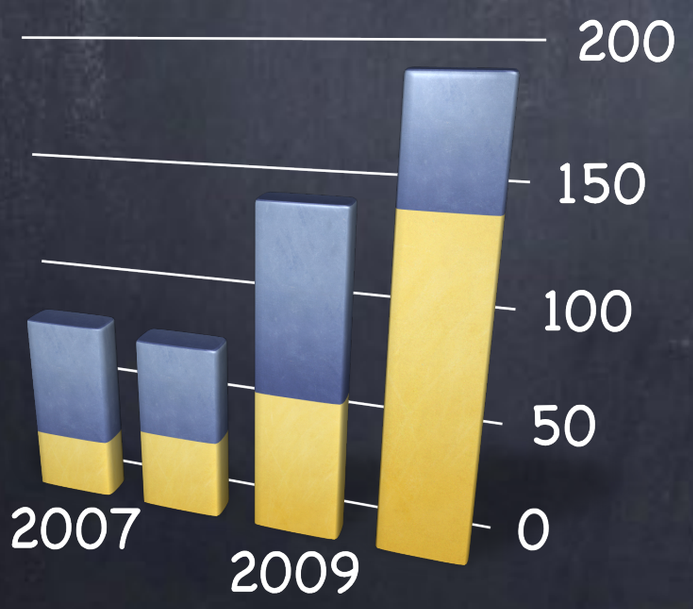
Visualisierung von Daten

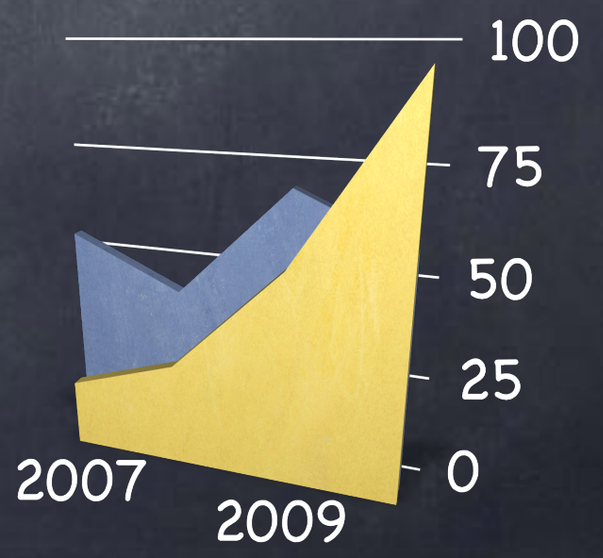
Präsentationsprogramme bieten zu viele Diagrammarten



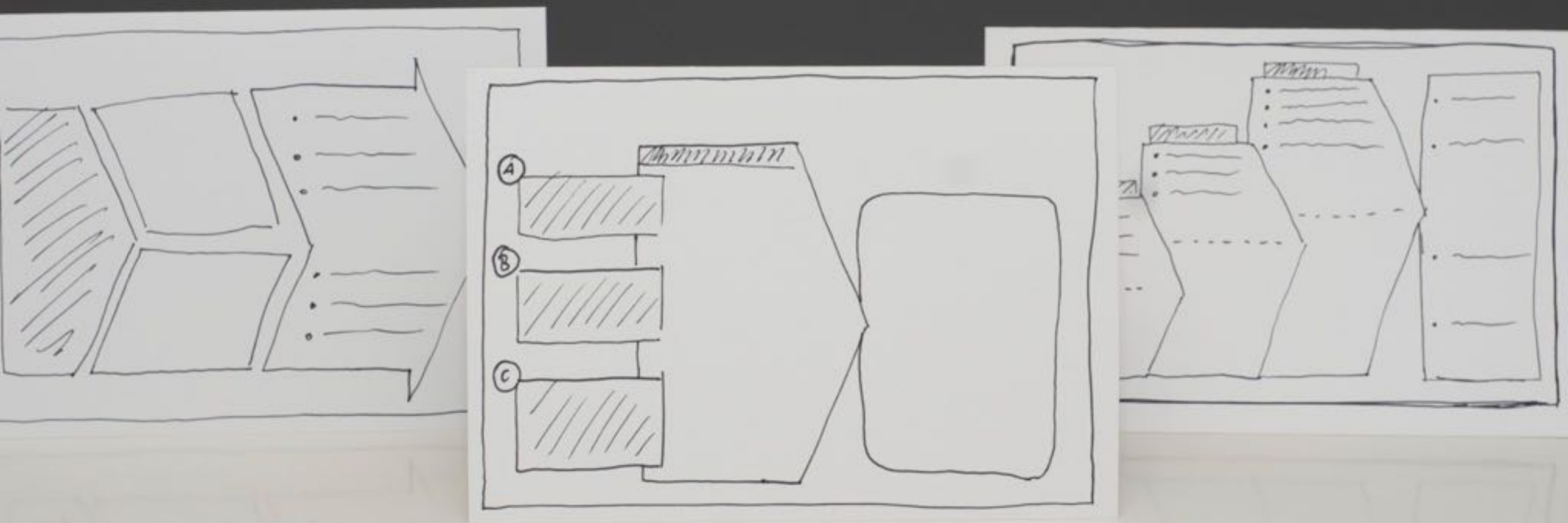








Visualisierung von textuellen Sachverhalten mit Schaubildern



Nicht manipulieren

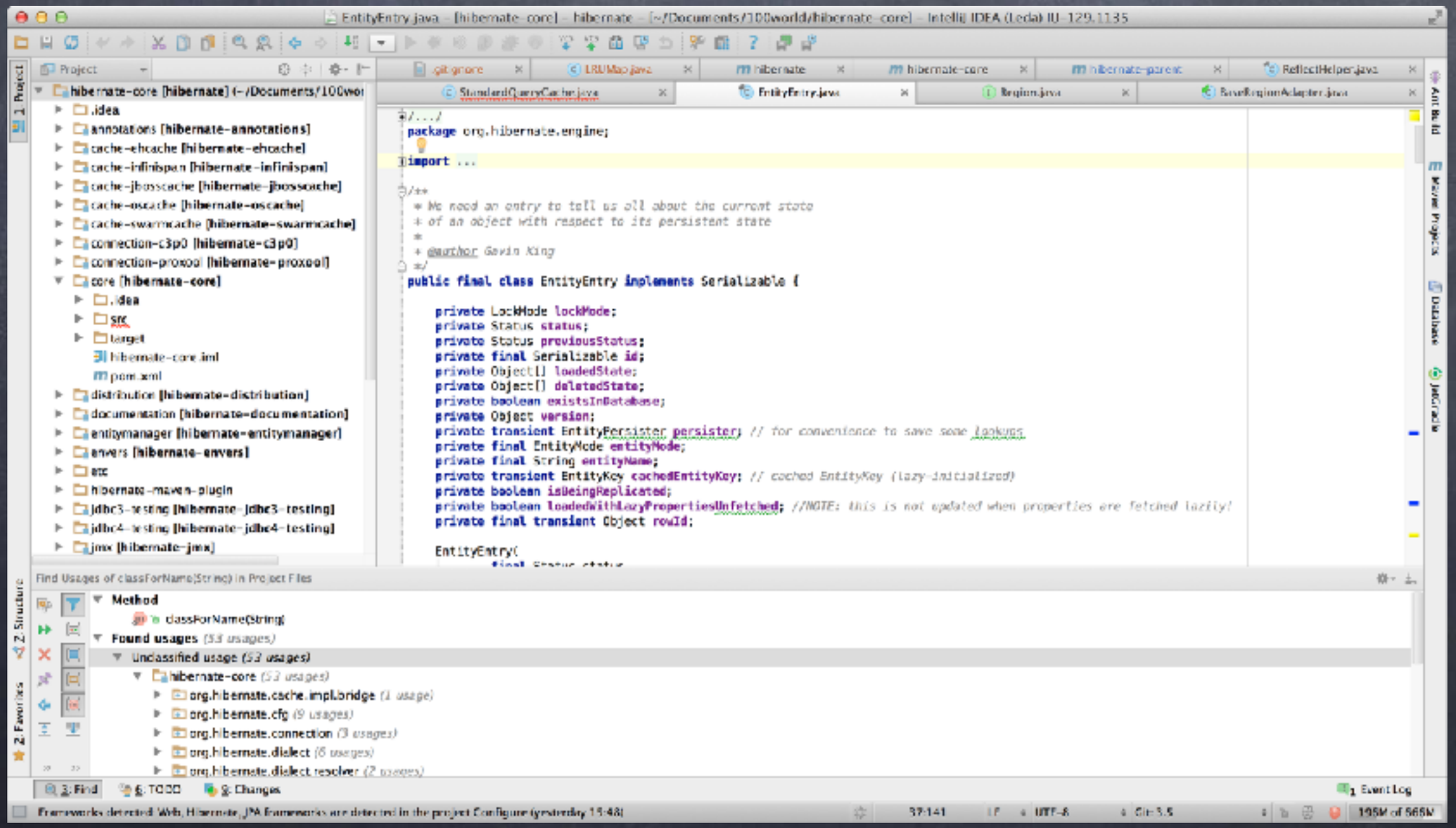
Visualisierung von Code

```
public List get(
                QueryKey key,
                Type[] returnTypes,
                boolean isNaturalKeyLookup,
                Set spaces,
                SessionImplementor session) throws HibernateException {
        if ( log.isDebugEnabled() ) {
                log.debug( "checking cached query results in region: " + cacheRegion.getName() );
        }

        List cacheable = ( List ) cacheRegion.get( key );
        if ( cacheable == null ) {
                log.debug( "query results were not found in cache" );
                return null;
        }

        Long timestamp = ( Long ) cacheable.get( 0 );
        if ( !isNaturalKeyLookup && !isUpToDate( spaces, timestamp ) ) {
                log.debug( "cached query results were not up to date" );
                return null;
        }

        log.debug( "returning cached query results" );
        for ( int i = 1; i < cacheable.size(); i++ ) {
                if ( returnTypes.length == 1 ) {
                        returnTypes[0].beforeAssemble( ( Serializable ) cacheable.get( i ), session );
                }
                else {
                        TypeFactory.beforeAssemble( ( Serializable[] ) cacheable.get( i ), returnTypes, session );
                }
        }
        List result = new ArrayList( cacheable.size() - 1 );
        for ( int i = 1; i < cacheable.size(); i++ ) {
                try {
                        if ( returnTypes.length == 1 ) {
                                result.add( returnTypes[0].assemble( ( Serializable ) cacheable.get( i ), session, null ) );
                        }
                        else {
                                result.add(
                                                TypeFactory.assemble(
                                                                ( Serializable[] ) cacheable.get( i ), returnTypes, session, null
                                                )
                                );
                        }
                }
                catch ( RuntimeException ex ) {
                        if ( isNaturalKeyLookup &&
                                        ( UnresolvableObjectException.class.isInstance( ex ) ||
                                        session.getFactory().getEntityNotFoundDelegate().isEntityNotFoundException( ex ) ) ) {
                                //TODO: not really completely correct, since
                                //      the uoe could occur while resolving
                                //      associations, leaving the PC in an
                                //      inconsistent state
                                log.debug( "could not reassemble cached result set" );
                                cacheRegion.evict( key );
                                return null;
                        }
                        else {
                                throw ex;
                        }
                }
        }
        return result;
}
```

Slide

```
public class TestHomePage extends TestCase {
    private WicketTester tester;
    @Override
    public void setUp() {
        tester = new WicketTester(new WicketApplication());
    }

    public void testRenderMyPage() throws Exception {
      tester.startPage(HomePage.class);
      tester.assertRenderedPage(HomePage.class);
      final SimpleDateFormat df = new SimpleDateFormat("hh:mm:ss");
      Date before = new Date();
      tester.assertLabel("static_clock", df.format(before));
      tester.assertLabel("dynamic_clock", df.format(before));
      Thread.sleep(2000);
      tester.clickLink("refresh", true);
      Date after = new Date();
      tester.assertLabel("static_clock", df.format(before));
      tester.assertLabel("dynamic_clock", df.format(after));
    }
}
```

Live Coding


EntityEntry.java - [hibernate-core] - hibernate - [~/Documents/100world/hibernate-core] - IntelliJ IDEA (Leda) IU-129.1135
Project
hibernate-core [hibernate] (~/Documents/100wor
.idea
annotations [hibernate-annotations]
cache-ehcache [hibernate-ehcache]
cache-infinispan [hibernate-infinispan]
cache-jbosscache [hibernate-jbosscache]
cache-oscache [hibernate-oscache]
cache-swarmcache [hibernate-swarmcache]
connection-c3p0 [hibernate-c3p0]
connection-proxool [hibernate-proxool]
core [hibernate-core]
.idea
src
target
hibernate-core.iml
pom.xml
distribution [hibernate-distribution]
documentation [hibernate-documentation]
entitymanager [hibernate-entitymanager]
envers [hibernate-envers]
etc
hibernate-maven-plugin
jdbc3-testing [hibernate-jdbc3-testing]
jdbc4-testing [hibernate-jdbc4-testing]
jmx [hibernate-jmx]
.gitignore
LRUMap.java
hibernate
hibernate-core
hibernate-parent
ReflectHelper.java
StandardQueryCache.java
EntityEntry.java
Region.java
BaseRegionAdapter.java
/.../
package org.hibernate.engine;
import ...
/**
* We need an entry to tell us all about the current state
* of an object with respect to its persistent state
*
* @author Gavin King
*/
public final class EntityEntry implements Serializable {
private LockMode lockMode;
private Status status;
private Status previousStatus;
private final Serializable id;
private Object[] loadedState;
private Object[] deletedState;
private boolean existsInDatabase;
private Object version;
private transient EntityPersister persister; // for convenience to save some lookups
private final EntityMode entityMode;
private final String entityName;
private transient EntityKey cachedEntityKey; // cached EntityKey (lazy-initialized)
private boolean isBeingReplicated;
private boolean loadedWithLazyPropertiesUnfetched; //NOTE: this is not updated when properties are fetched lazily!
private final transient Object rowId;
EntityEntry(
Find Usages of classForName(String) in Project Files
Method
classForName(String)
Found usages (53 usages)
Unclassified usage (53 usages)
hibernate-core (53 usages)
org.hibernate.cache.impl.bridge (1 usage)
org.hibernate.cfg (9 usages)
org.hibernate.connection (3 usages)
org.hibernate.dialect (6 usages)
org.hibernate.dialect.resolver (2 usages)
3: Find
6: TODO
9: Changes
1 Event Log
Frameworks detected: Web, Hibernate, JPA frameworks are detected in the project. Configure (yesterday 15:48)
37:141
UTF-8
Git: 3.5
195M of 565M
1: Project
2: Favorites
Z: Structure
Ant Build
Maven Projects
Database
JRebel

TinyMCEDemoPage.java × TinyMCEDemoPage.html ×

```java
import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.markup.html.form.TextArea;
import wicket.contrib.tinymce.InPlaceEditComponent;
import wicket.contrib.tinymce.TinyMceBehavior;

public class TinyMCEDemoPage extends WebPage{
    public TinyMCEDemoPage() {
        TextArea txt = new TextArea("txt");
        add(txt);
    }
}
```

Screencast

Plane alles

Offline

Schritt 5:

Vortrag

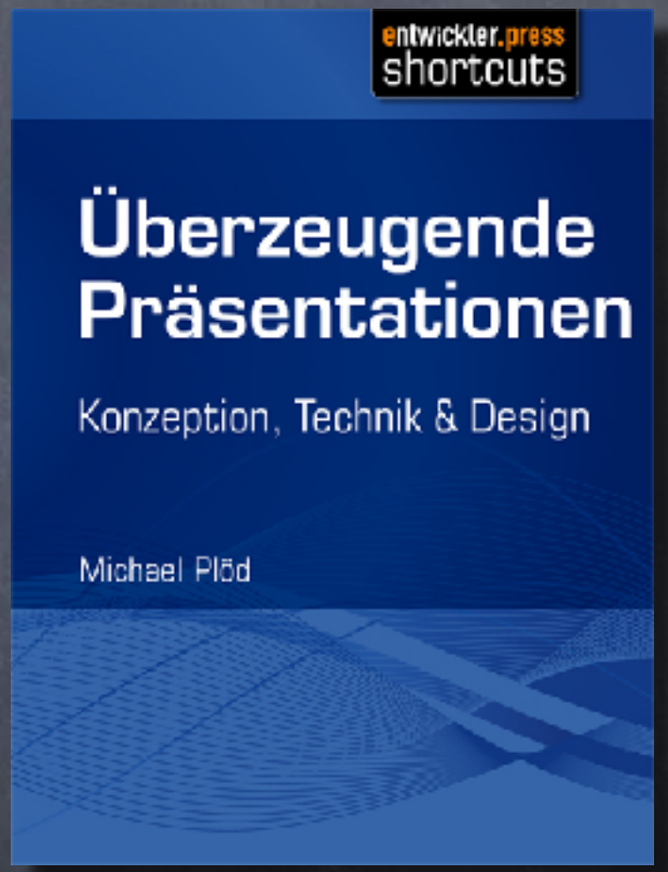
Anti-Chaos

Checkliste

- [ ] Üben
- [ ] Frühe Ankunft
- [ ] Infrastruktur Check
- [ ] Wasser
- [ ] Durchatmen.....

Showtime!

E-Book


entwickler.press
shortcuts
Überzeugende Präsentationen
Konzeption, Technik & Design
Michael Plöd

Vielen Dank!!!

michael.ploed@innoq.com

@bitboss

http://speakerdeck.com/mploed