

Die wahren Probleme finden

Wider die Mikroskop-Falle

Dr. Gernot Starke

innoQ



Dr. Gernot Starke

innoQ Fellow



Schwerpunkte:

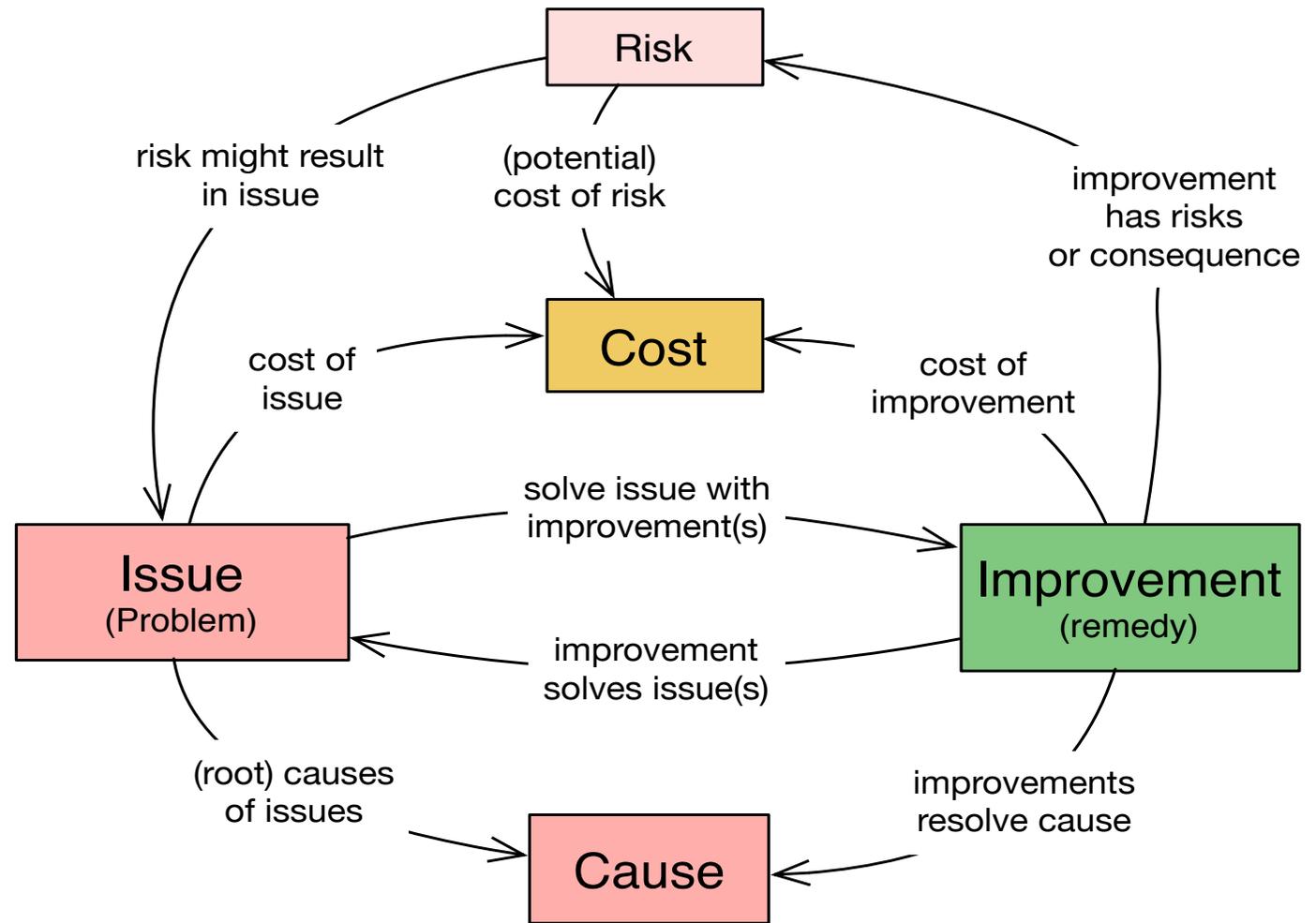
- ▶ Softwarearchitekturen
Entwurf, Entwicklung, Evolution
Modernisierung,
Dokumentation
- ▶ Mentoring und Coaching
- ▶ Analyse und Optimierung
von Entwicklungsprozessen
- ▶ Reviews, Audits,
Retrospektiven



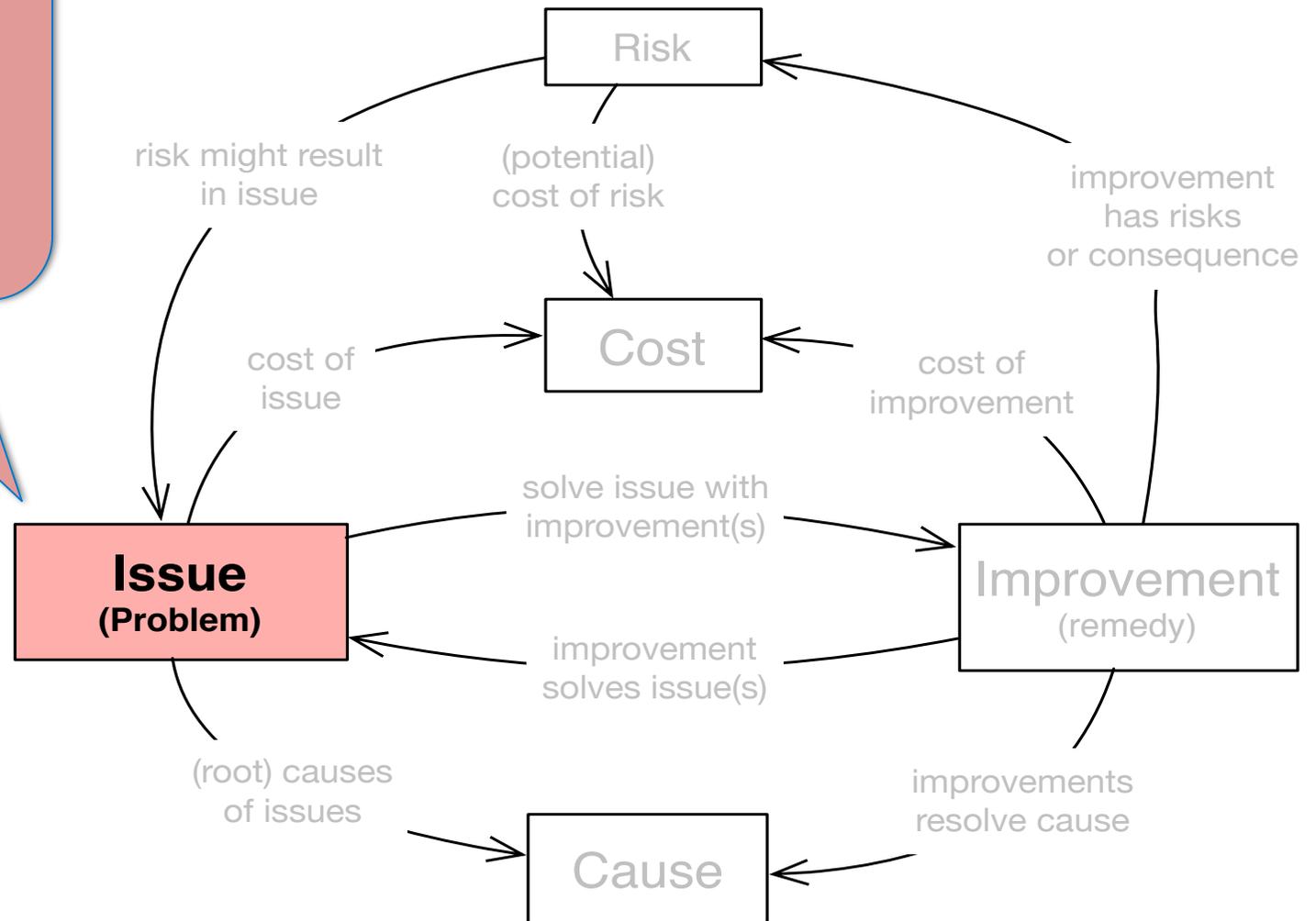
+49 177 – 728 2570
gs@gernotstarke.de

 Follow @gernotstarke

www.arc42.de



Probleme ==
Auslöser für
Verbesserung



Komplizierte Systeme

- › Viele Bestandteile mit Abhängigkeiten
(**Strukturen**)
- › Übergreifende Regeln
(**Konzepte**)
- › Mehrere Beteiligte...

Kompliziertes System



<https://flic.kr/p/7XAjXu>

Probleme in kompliziertem System suchen (o)



Geschmack,
Preis,
Aussehen,
Lage (des Restaurants)



Probleme in kompliziertem System suchen (1)

Geschmack,
Temperatur,
Aussehen,
Konsistenz,
Beliebtheit

Probleme in kompliziertem System suchen (2)

Anteile an Kohlenhydrat, Protein, Fett, Vitamin

Nachweis von Drogen, Schad- oder Giftstoffen, oder Radioaktivität



Probleme in kompliziertem System suchen (3)



Logistik: Transport-
und Lagerung

Probleme in kompliziertem System suchen (4)

Einkauf, Herstellung,
Verpackung, Vertrieb

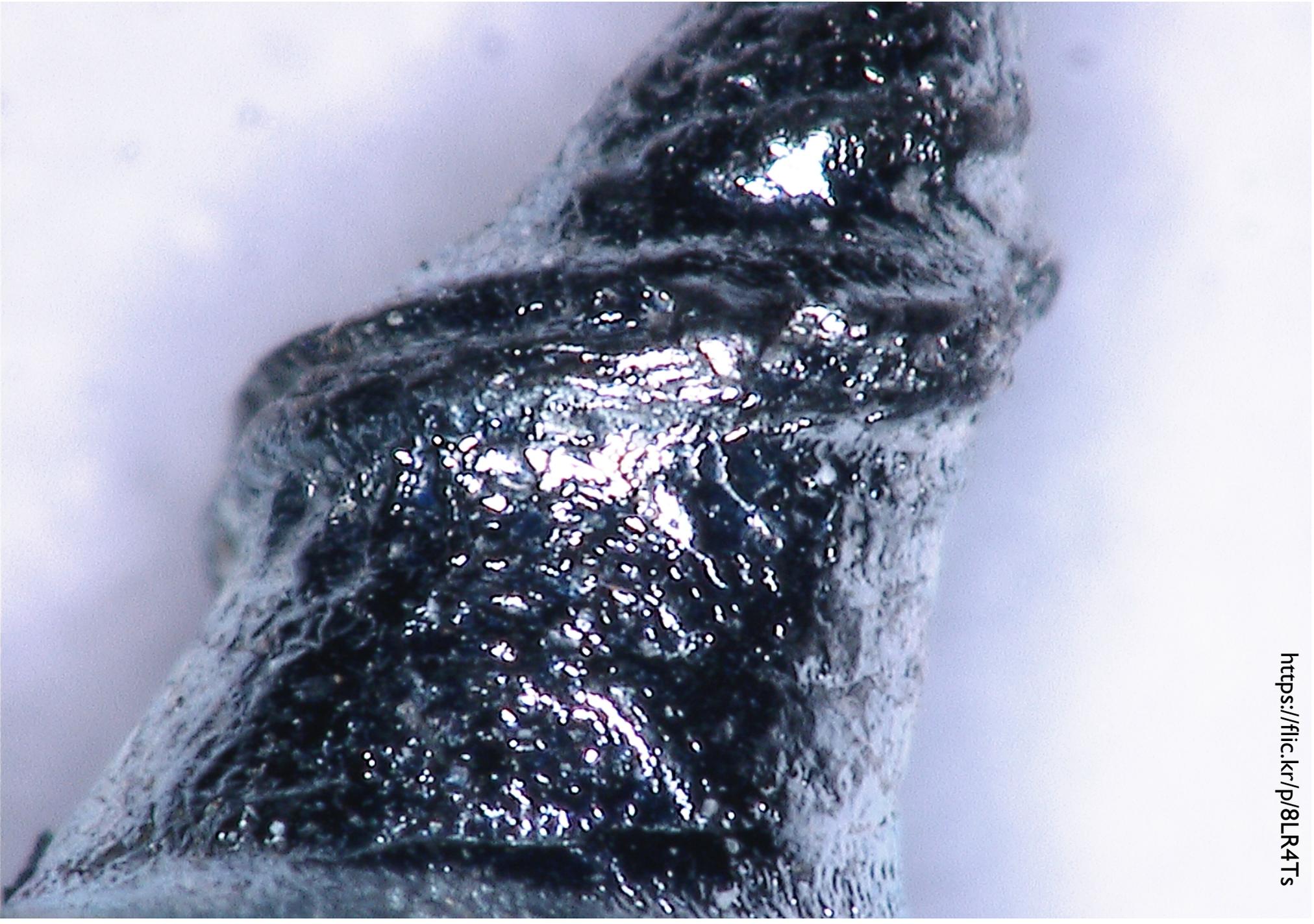


<https://flic.kr/p/dZfxi5>

Probleme in komplizierten Systeme

- › Probleme einzelner Bestandteile
- › Probleme bei Abhängigkeiten
- › Probleme mit übergreifenden Regeln
- › Probleme mit Beteiligten
- › Probleme mit Prozessen







<https://www.flickr.com/photos/jonasb/24341322>

<https://www.flickr.com/photos/emiliano-iko/>
5354414276



Die Mikroskop-Falle

Wenn Sie NUR im Code suchen,
werden Sie NUR DORT Probleme finden...

im Code suchen ist richtig und wichtig,
aber NUR dort suchen kann fatal sein!

Software...

- › Viele Bestandteile
- › Querschnittliche & technische Konzepte
- › Viele Beteiligte
- › Viele Kategorien von Problemen

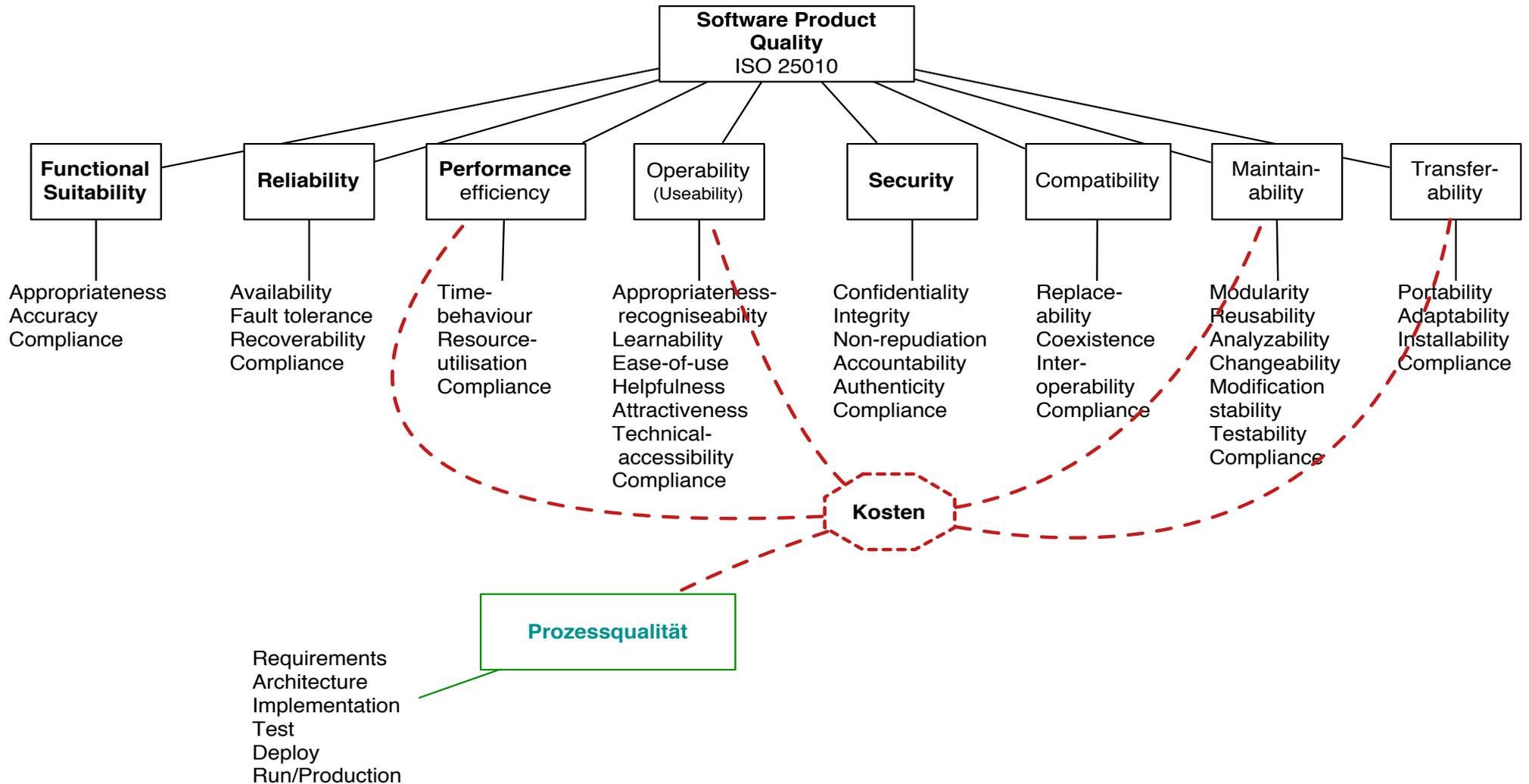
- › Oft: langjähriger Einsatz (== Änderungen)



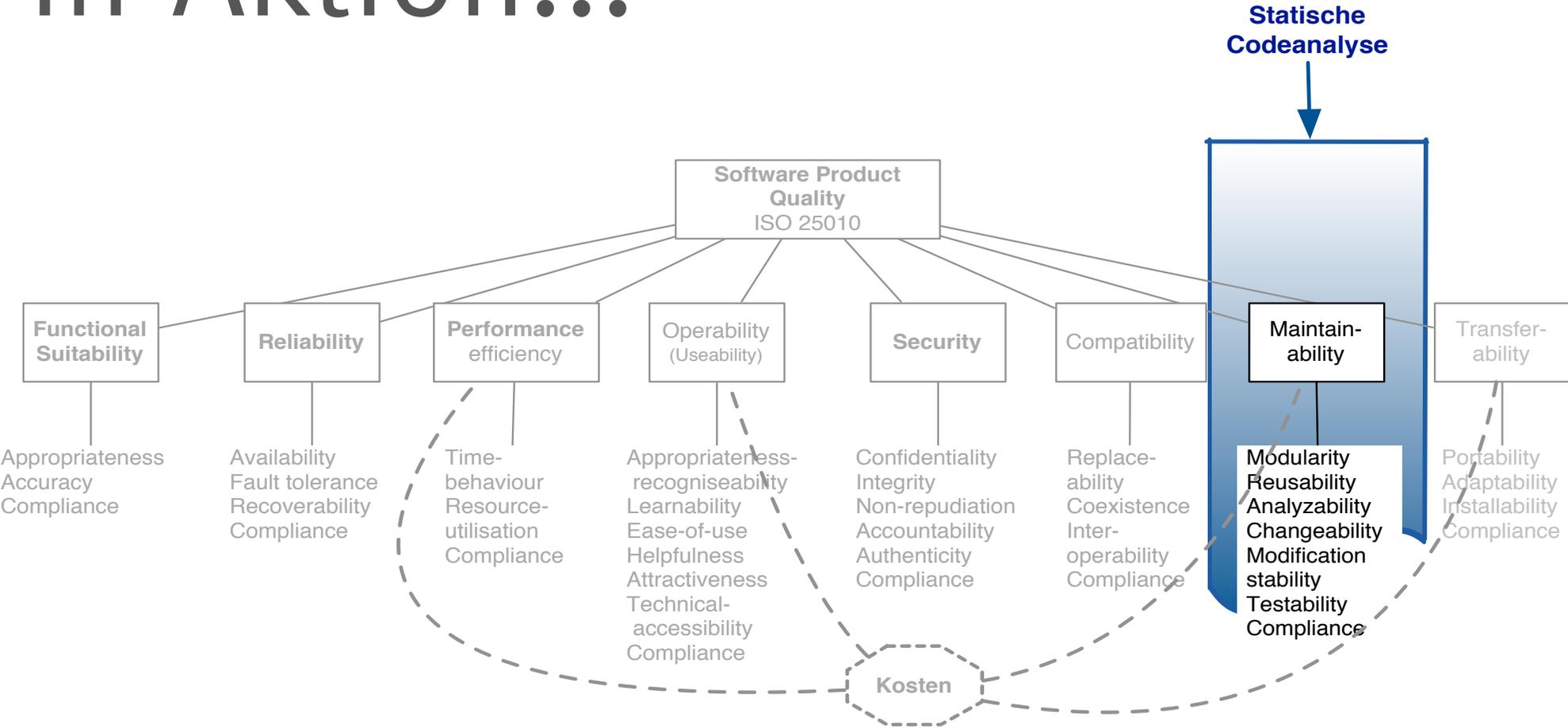
Kategorien... (1)



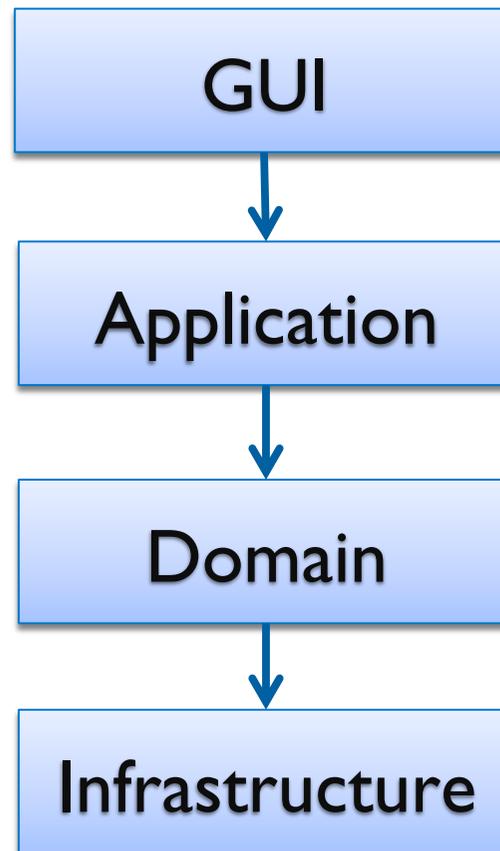
Kategorien... (2)



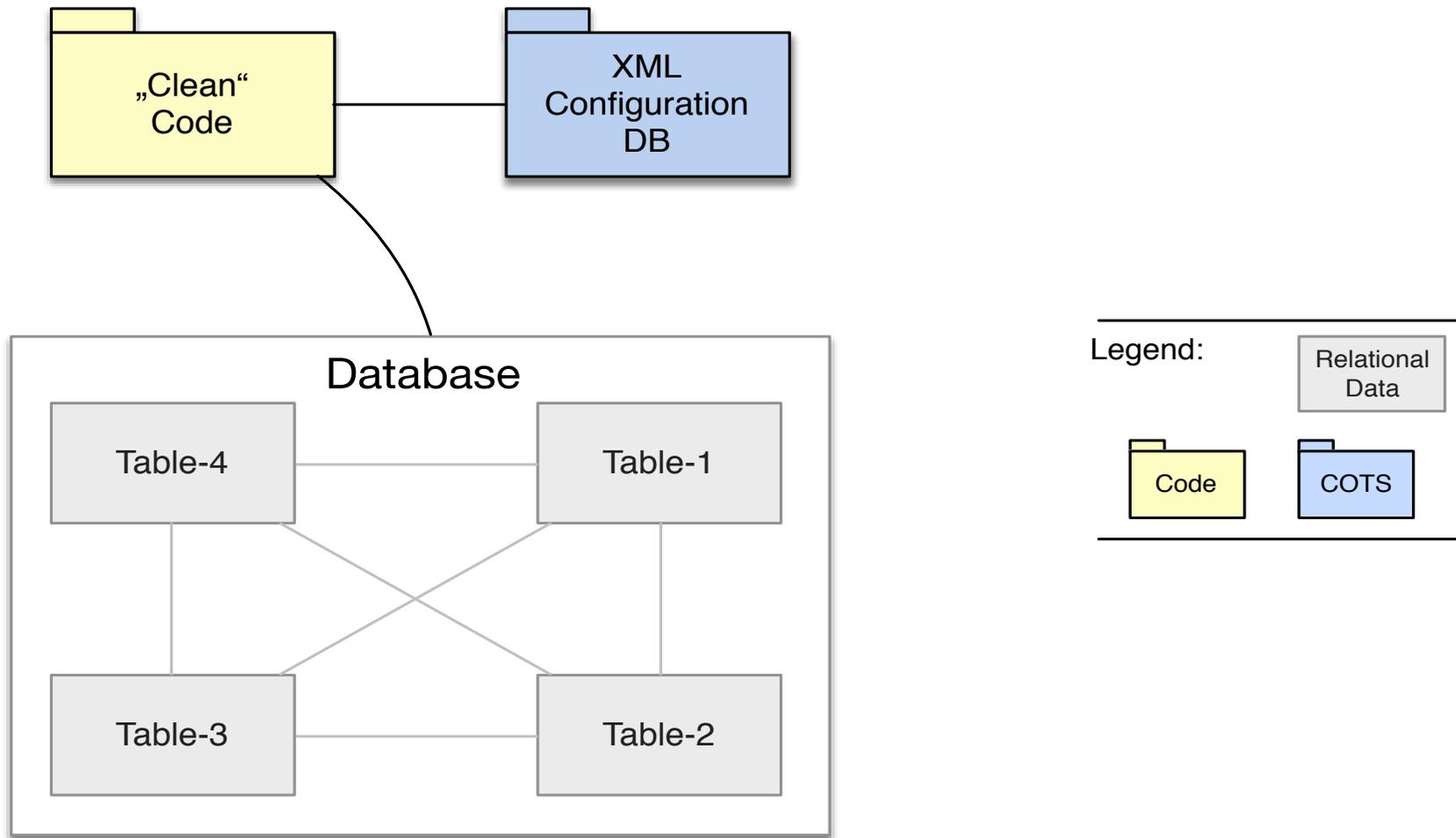
Mikroskoprisiko in-Aktion...



Beispiel: Saubere Schichtung...



Beispiel: Saubere Schichtung, aber...



Die Werkzeugfalle

Statische-Analyse-Tools finden
nur eine kleine Menge der Probleme



Gute Problemsuche...

Kombiniere Breiten- und Tiefensuche



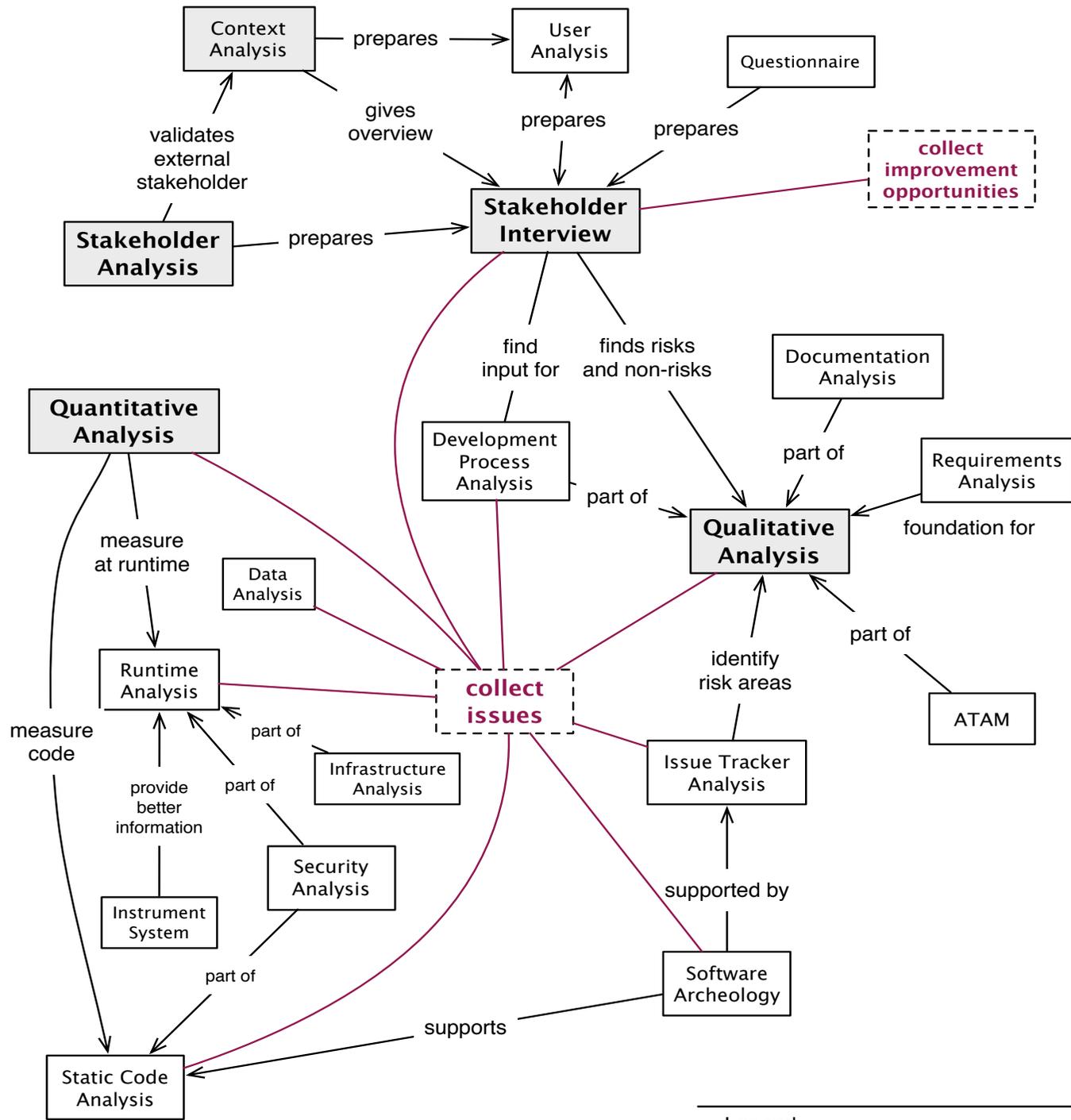
Qualitative und quantitative Analyse in erweitertem Suchraum

Wider die Mikroskop-Falle
Dr. Gernot Starke

Treffender Titel

innoQ





Legend:

Nach: aim42

Stakeholder

... wichtige Personen oder Organisationen, die:

- › Interesse am System haben,
- › von System oder Architektur betroffen sind:

– damit arbeiten

– daran arbeiten

– davon profitieren

– darunter leiden

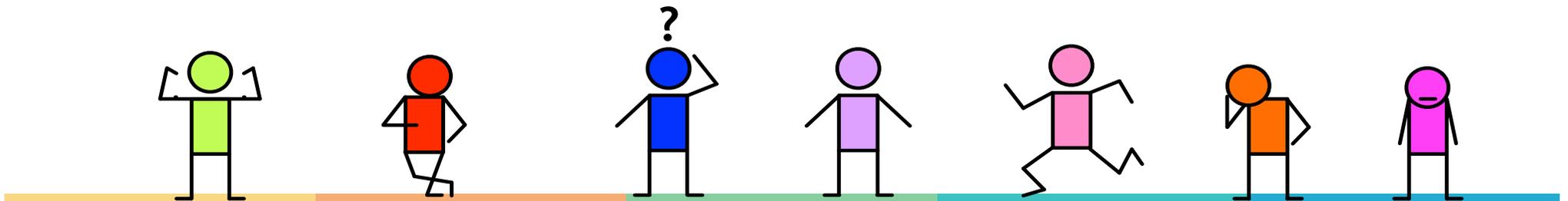
– es

- betreiben

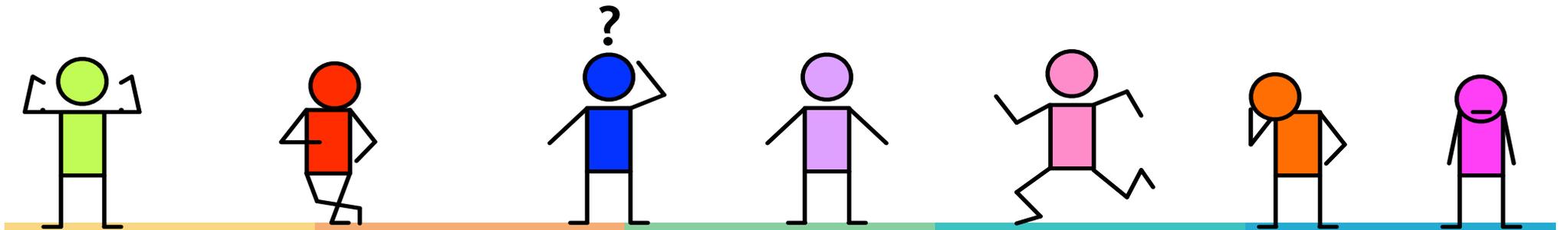
- verwalten

- regulieren

- bezahlen

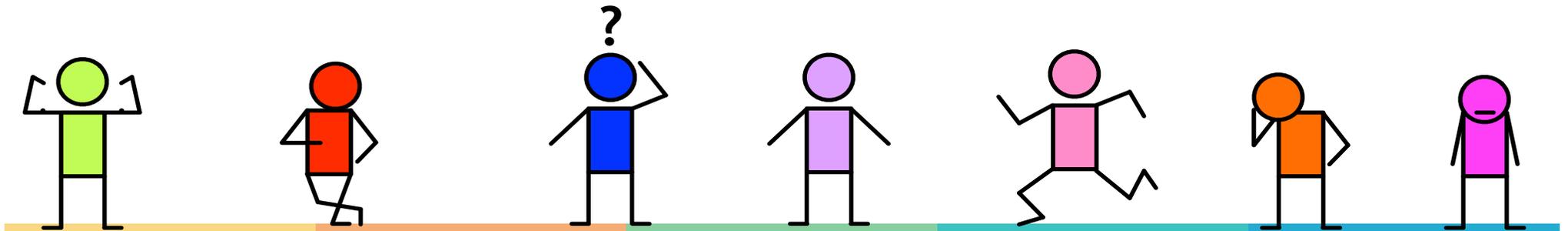


Stakeholder kennen viele Probleme



Beispiel: Stakeholder-Interviews...

Kurzzeitgedächtnis bekam mehr Fokus...



Vorsicht – Stakeholder...

Erleben **Gewöhnungseffekt**:

- › ...an Probleme oder Workarounds

Haben **Angst vor Änderung**

- › menschlich



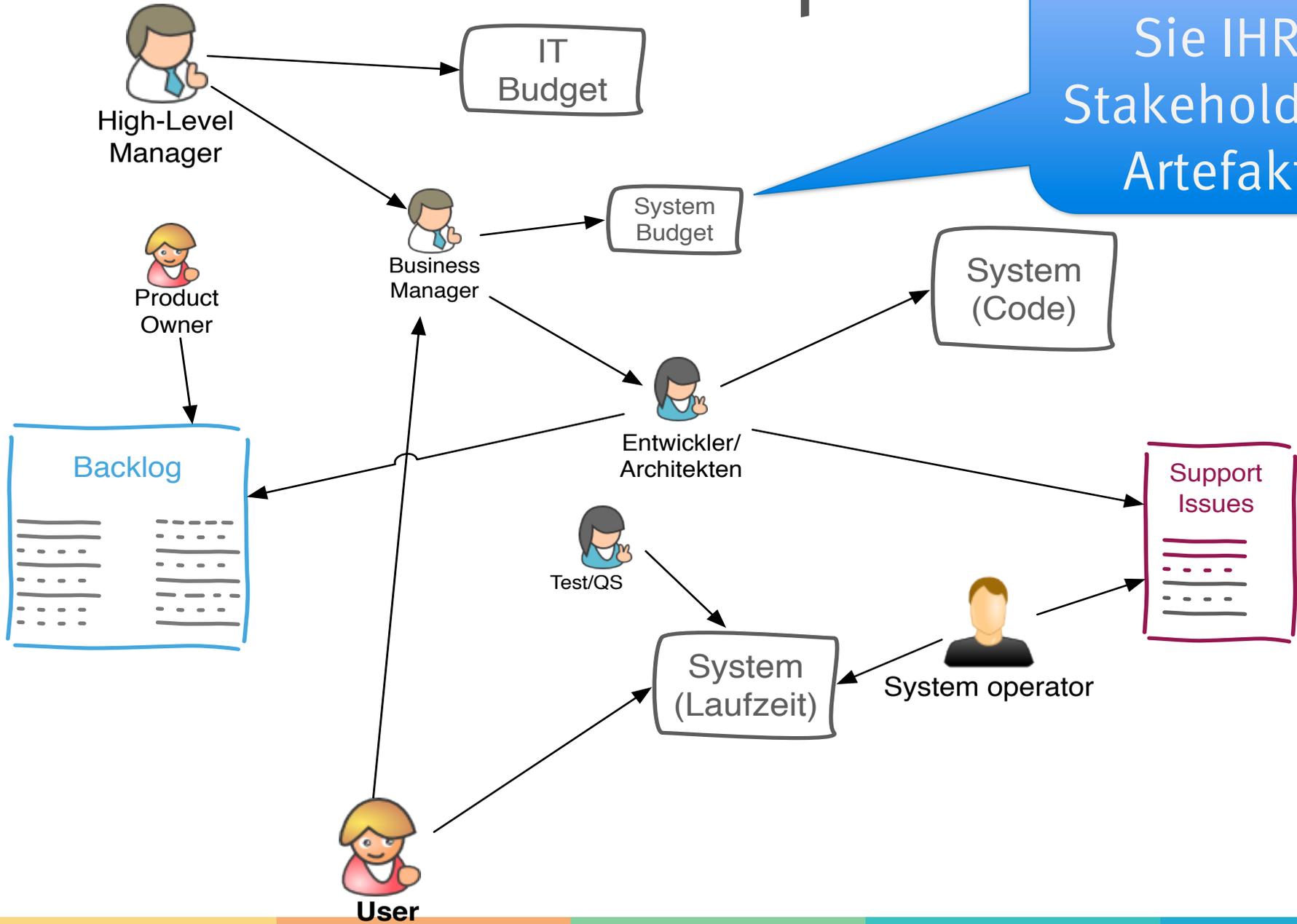
Stakeholder...

**Welche Personen
oder Rollen?**

- › nehmen Probleme **subjektiv** wahr
 - › nennen oftmals **Symptome**, keine Ursachen
 - › taktieren
 - › äußern **Vermutungen**
-

Stakeholder-Map

Visualisieren Sie IHRE Stakeholder & Artefakte



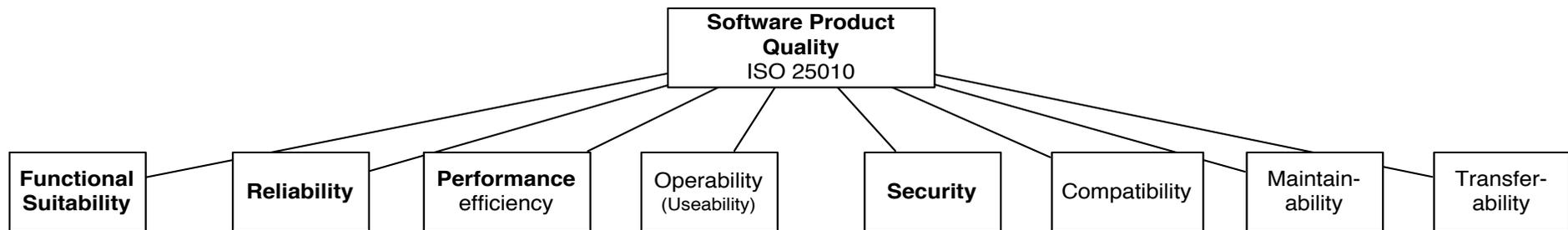
Übersicht: Systemanalyse

- › Qualitative Analyse
- › Quantitative Analyse („Vermessung“)
 - › Code
 - › Laufzeit
- › Datenanalyse
- › Kontextanalyse
- › ...



Qualitative Analyse

- › Welche Qualitätsziele sind „gefährdet“?



- › Soll-Ist Vergleich:

- › Soll: konkrete Q-Ziele

- › Ist: Lösungsansätze (Code, Konzepte, Entscheidungen...)

Genauer: Vergleich von Anforderungen mit Lösung oder Lösungsansätzen

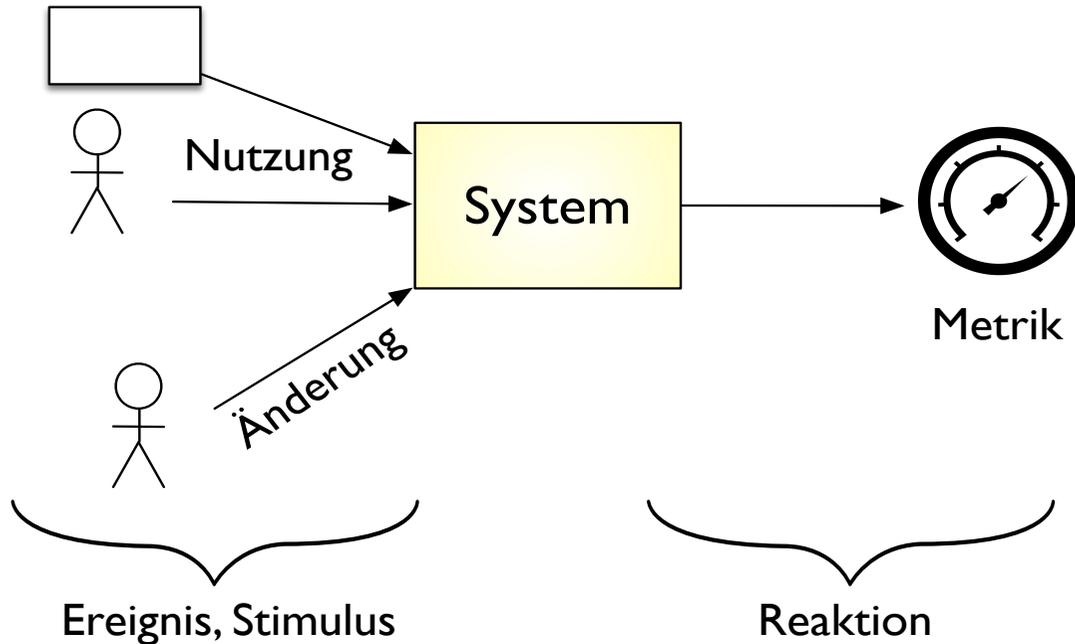


Qualitative Analyse - Vorgehen

1. Maßgebliche Stakeholder **explizieren**
Qualitätsanforderungen (als Szenarien)
2. Architekten/Entwickler benennen zugehörige
Lösungsansätze des Systems
3. Bewerter vergleichen und schätzen Risiko



Szenarien...



Interaktionen von
Stakeholder und
System

Wie reagiert das
System auf Stimulus?

- Anwendungs-/Nutzungsszenarien
- Änderungsszenarien

Qualitätsziele

| Q-Ziel | Bedeutung / Szenarien |
|--------------------|---|
| Flexibilität | <ul style="list-style-type: none">• Neues csv-Importformat in <4h konfigurierbar |
| Last / Performance | <ul style="list-style-type: none">• 250.000 eingelieferte Fotos innerhalb von 2h prozessiert |
| Sicherheit | <ul style="list-style-type: none">• Mandant kann niemals Zugriff auf Daten anderer Mandanten erhalten |

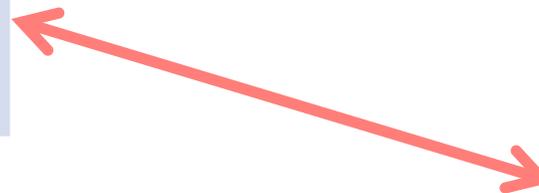
Lösungsansätze

Architektur-/Lösungsansatz

- Konfigurationssprache für CSV-Parser (Import), auf Basis ANTLR
- Syntaxgesteuerter Editor für die Sprache
- Bilder als Dateien speichern, Links in DB
- Lasttests im DailyBuild
- Generator für (Massen-)Testdaten
- Mandantenspezifische Daten grundsätzlich in (eigener) VM
- Datenlieferungen grundsätzlich in mandantenspezifische Verzeichnisse (ftp-Server)
- Unix-Kennungen spezifisch für Mandanten



Risiken?



Quantitative Analyse (1)

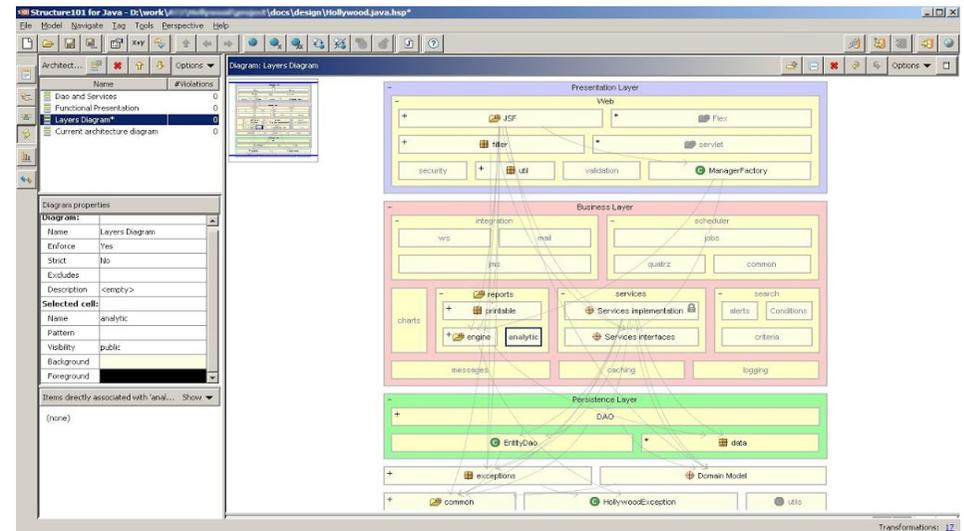
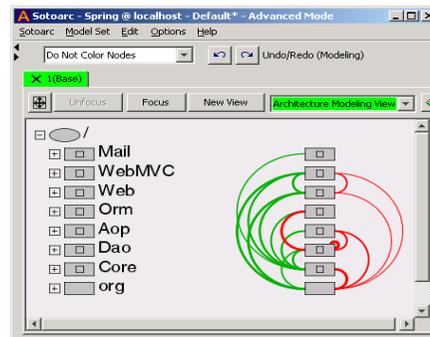
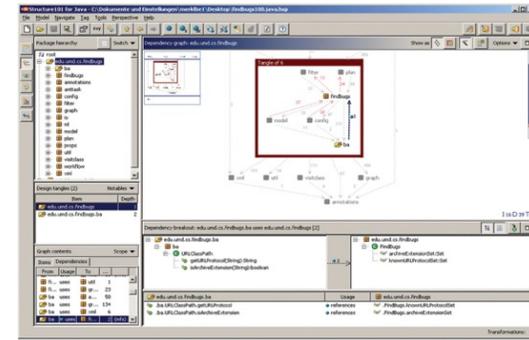
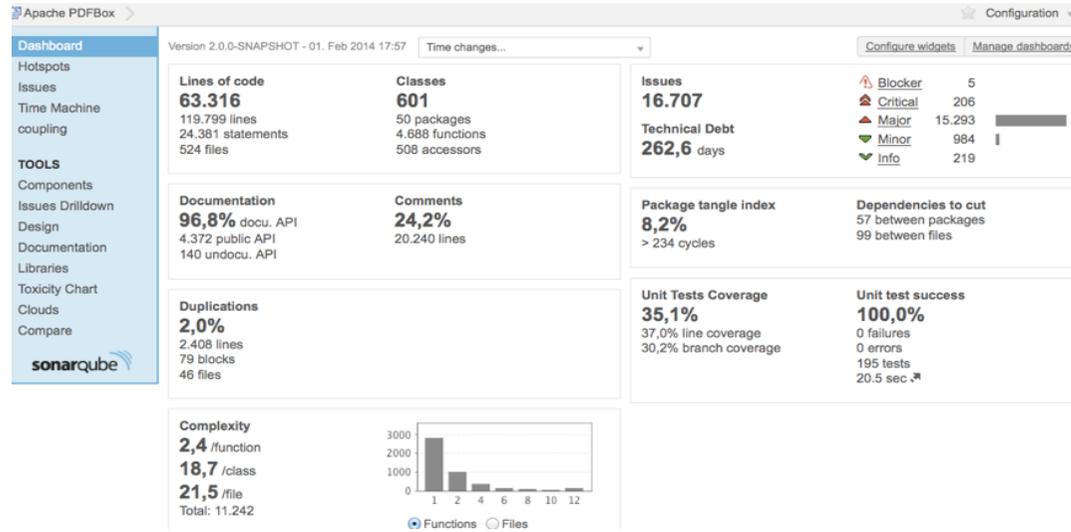
- statische Codeanalyse-

- › Kopplung
- › Komplexität
- › Konsistenz
 - › Verletzung von Idiomen (Style-Checking)



**Wichtig für
Wartbarkeit!!**

Statische Codeanalyse (2)



Beispiel (1)

Anmerkung: je höher desto schlechter

Komplexität: 2
Kopplung: 10

Komplexität: 10
Kopplung: 30

Komplexität: 9
Kopplung: 35

Komplexität: 7
Kopplung: 20

Fix
me?

Vorsicht!

Beispiel (2)

Komplexität: 2
Kopplung: 10

DTFB: 0.5
Bugs: 200

Komplexität: 10
Kopplung: 30

DTFB: 2
Bugs: 30

Komplexität: 9
Kopplung: 35

DTFB: 3
Bugs: 20

Komplexität: 7
Kopplung: 20

DTFB: 8
Bugs: 15

DTFB: Days-to-Fix-Bug



Vorsicht: einzelne Metriken



DTFB: Days-to-Fix-Bug

Quantitative Analyse (3)

Korrelierte Analyse

**Abgleich unterschiedlicher
Messungen**

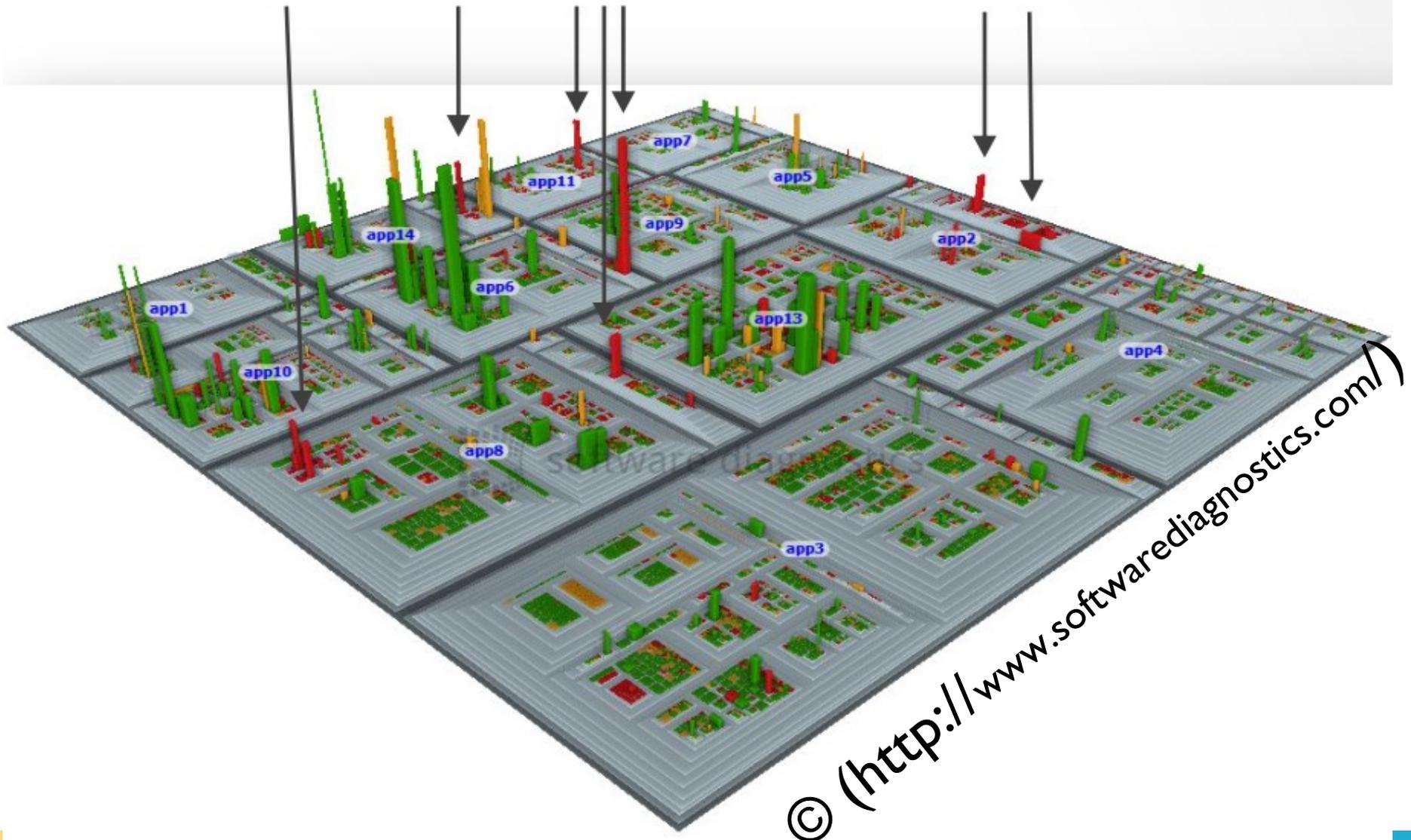
Beispiele:

- › Fehler pro Komponente
- › (Zeit pro Bugfix) pro Komponente
- › (Aufwand pro Feature) pro Komponente



Korrelierte Codeanalyse (2)

Risky situation: Very complex code and only a single developer knows it. Costly effort-bombs will explode when the knowledge-having developer leaves the team.



© (<http://www.softwarediagnostics.com/>)

Quantitative Bewertung: „Änderungshäufigkeit“

```
git log --since="90 days ago" --pretty=format:"" --name-only | \  
grep "[^\s]" | \  
sort | uniq -c | \  
sort -nr | head -10
```

GIST: <https://goo.gl/kzTgFa>

Laufzeitanalyse (1)

- › Logfile- oder Stacktrace-Analyse („Was ist geschehen“?)

- › Analyse von Ressourcenverbrauch („Profiling“)
 - › **Zeitverbrauch**
 - › **Speicherbedarf**
 - › Threading,
 - › Netzwerkressourcen

Laufzeitanalyse (2)

- › Analyse von Benutzerverhalten

- › Wo und womit verbringen Benutzer Zeit?
- › Welche Abläufe dauern ungebührlich lange?
- › Korrelieren fachliche Schwierigkeit + reale Dauer?



Kontextanalyse

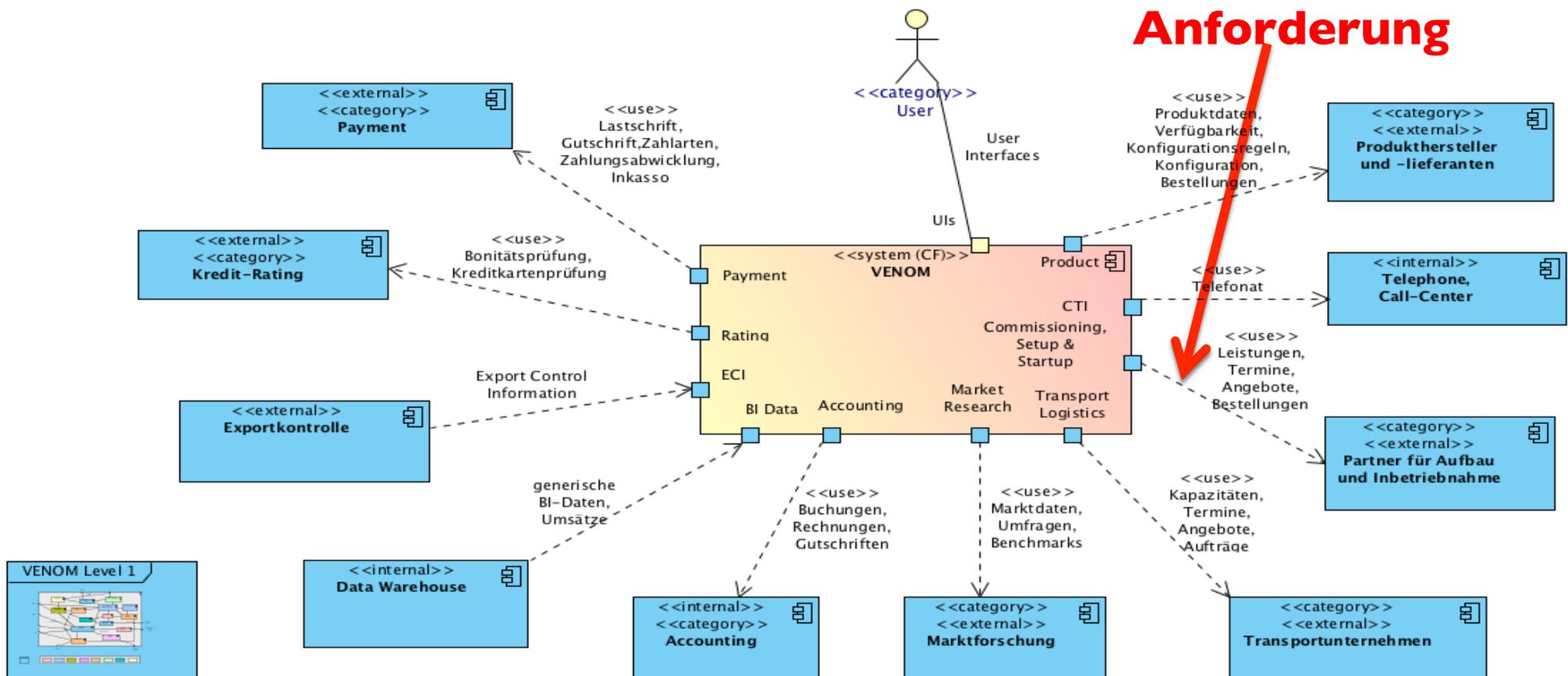
Risiken / Probleme im Kontext:

- › Abhängigkeiten bzgl. Qualitätszielen
 - › Verfügbarkeit, Robustheit,
 - › Sicherheit
 - › Kosten
 - › Performance
 - › ...
- › Fehlende Schnittstellen



Kontextanalyse - Beispiel

Risiko bzgl. Performance-Anforderung



Datenanalyse

1. Struktur

Struktur / Typen ungeeignet für das Problem

2. Typen

3. Zugriffe

› Read / write

Wonach ist Persistenz optimiert, read oder write?

4. Volumen

› auch von Query-Results + Indizes

Haben wir besonders viel? Ist etwas besonders groß?

5. Korrektheit

Haben wir falsche Daten?

6. Schutzbedarf

Haben wir sensible Daten?

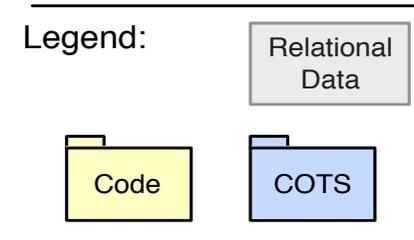
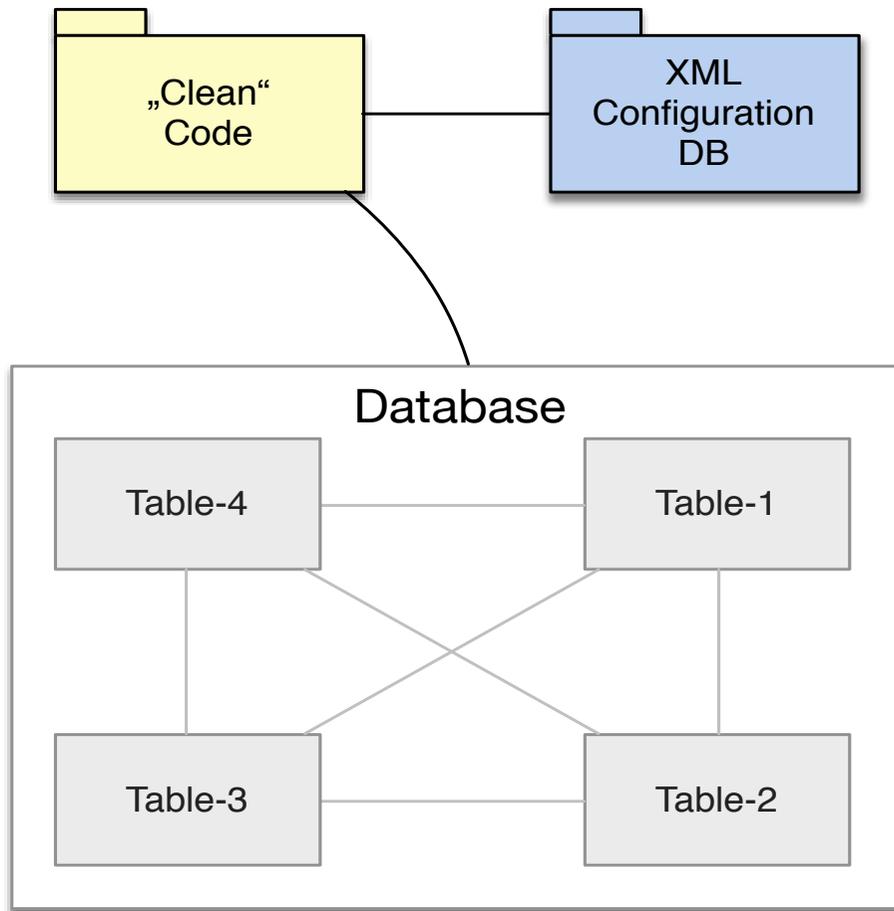
7. Verteilung/Dezentralisierung

8. Duplikation/Redundanz

Halten wir Daten mehrfach?

9. Durchsatz -> Laufzeitanalyse

Beispiel: Datenanalyse



Prozessanalyse

- › Anforderungsprozesse
 - › erheben, klären, managen
- › Entwicklungs- /Entwurfsprozesse
 - › Architektur, Implementierung, Dokumentation
- › Betrieb
 - › Deployment, Rollout, Administration, Monitoring
- › Management
 - › Team- und Taskmanagement,

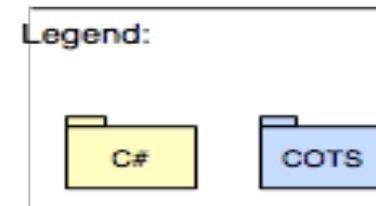
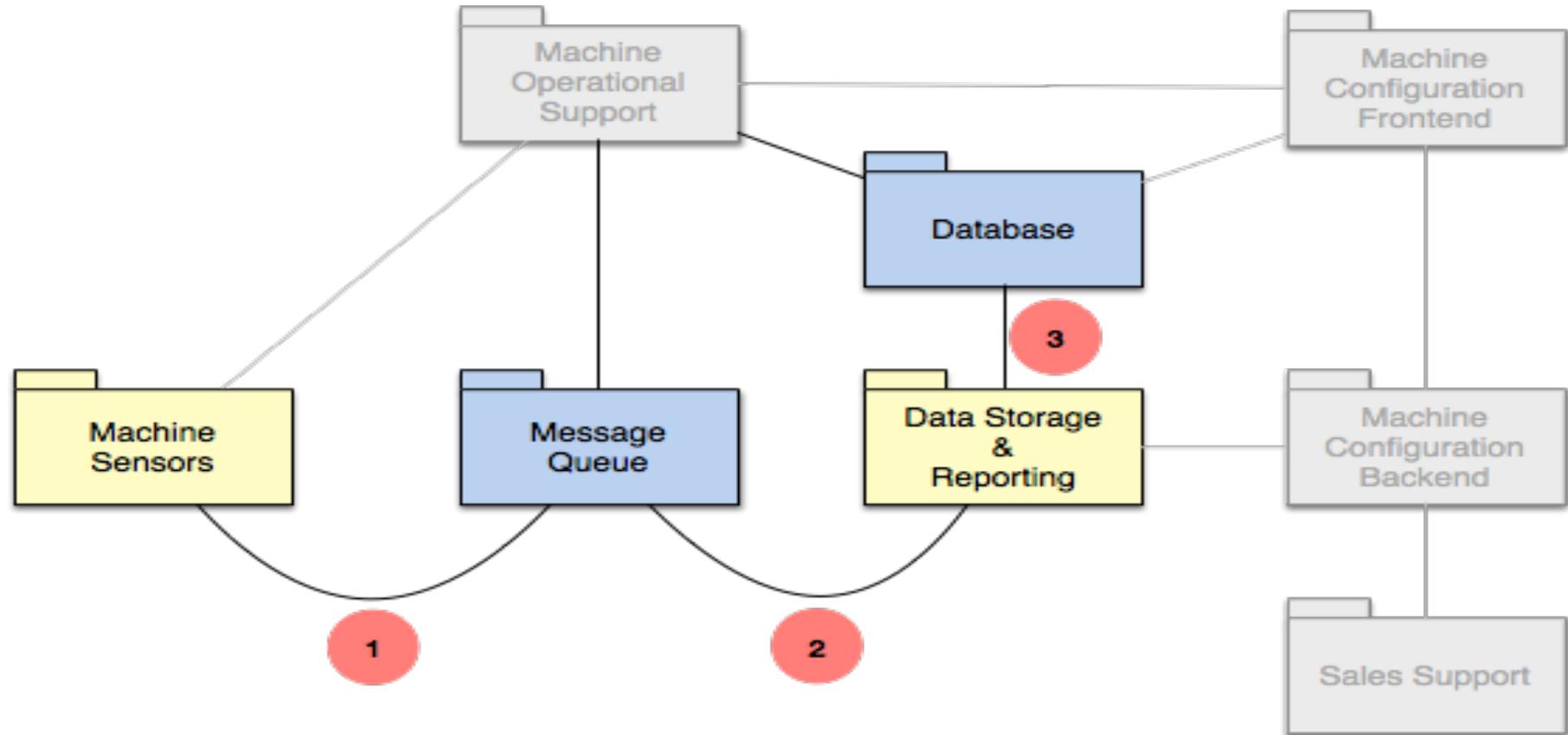
Prozessanalyse (2)

- › Welche **Voraussetzungen** müssen für X gelten?
- › Wer ist an X **beteiligt**?
- › Wer/Was ist der **Flaschenhals** bei X?
- › **Wie lange** dauert X?
- › Welche typischen Probleme treten bei X auf?

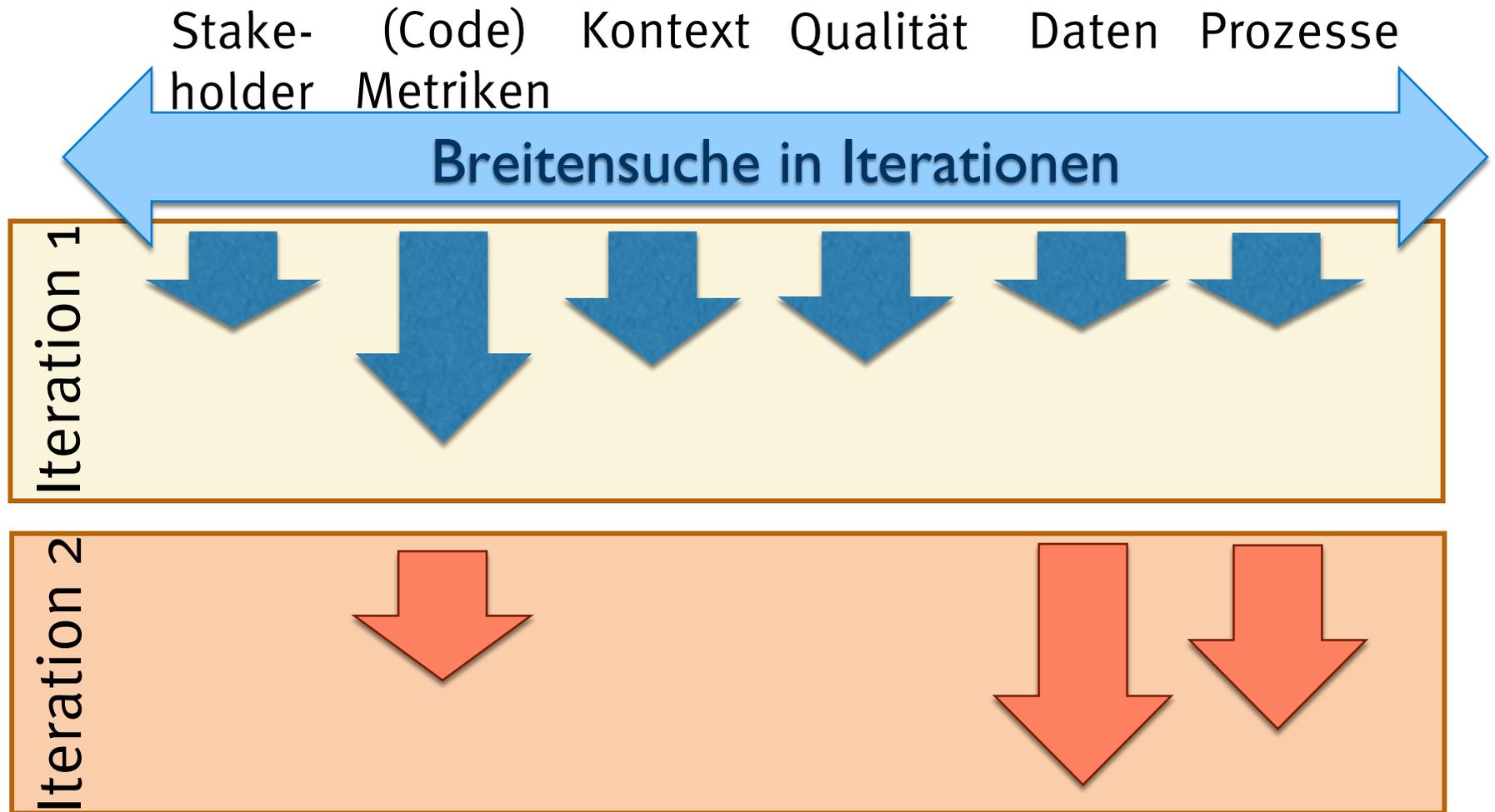
für X = „wiederkehrende Aktivität/Prozess/Aufgabe“



Prozessanalyse (Beispiel)



Zusammenfassung



Relativitäts- Falle



Des Einen
Problem
Ist des
Anderen
Freund

Mögliche Reaktionen...



profitieren vom Problem,
haben das Problem erschaffen,
greifen Sie und Ihre Vorschläge an

sind am Problem schuld,
stehen zu Ihnen in Konkurrenz,
greifen Ihre Kompetenz an

leiden nicht am Problem,
leiden unter dessen Lösung
tragen (Teil-)Schuld am
Problem

zweifeln Schlussfolgerungen an,
zweifeln Ihre Kompetenz an,
haben Angst vor Veränderung



Hostility

Resistance

Minimization

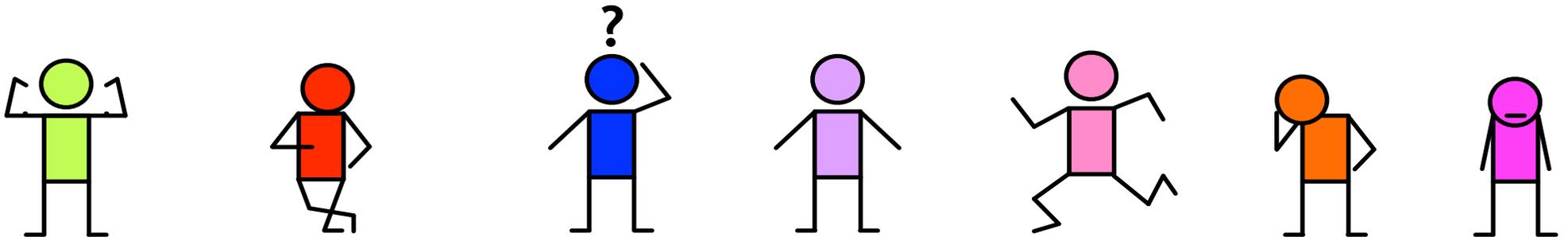
Doubt

Ausblick: Probleme lösen

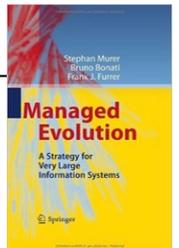
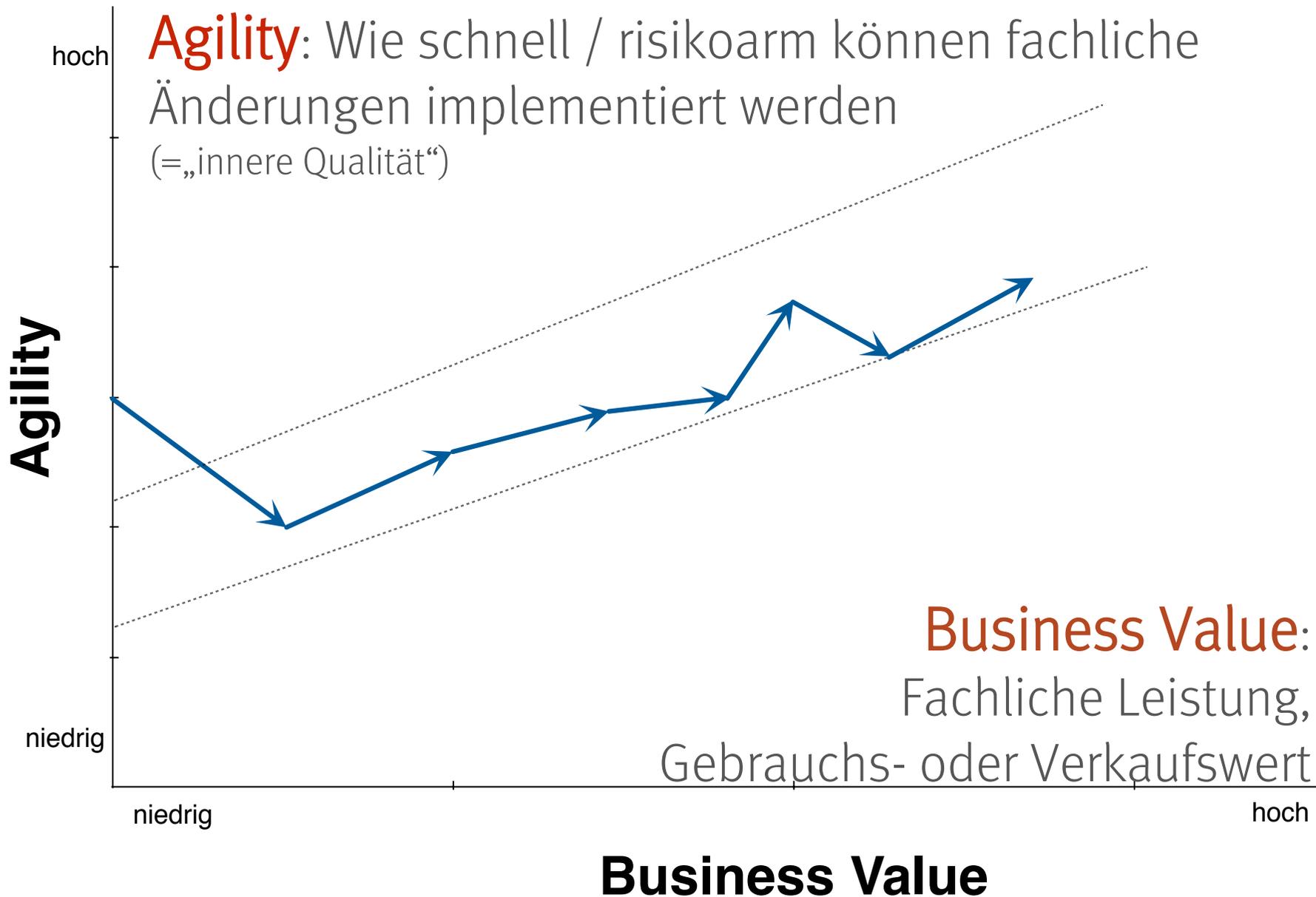
innoQ



You want to improve systems,
you need to **talk business.**

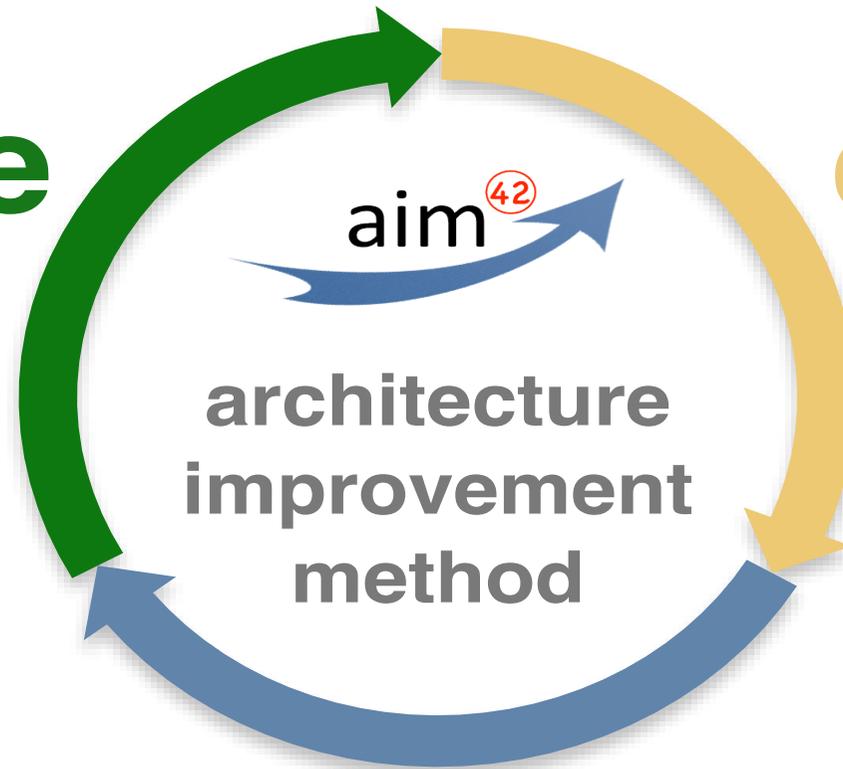


Zwei Dimensionen...



analyze

evaluate



improve



Fazit (o)

Kochen / Essen:
Kohlenhydrat-,
Protein- und
Fettgehalt prüfen.



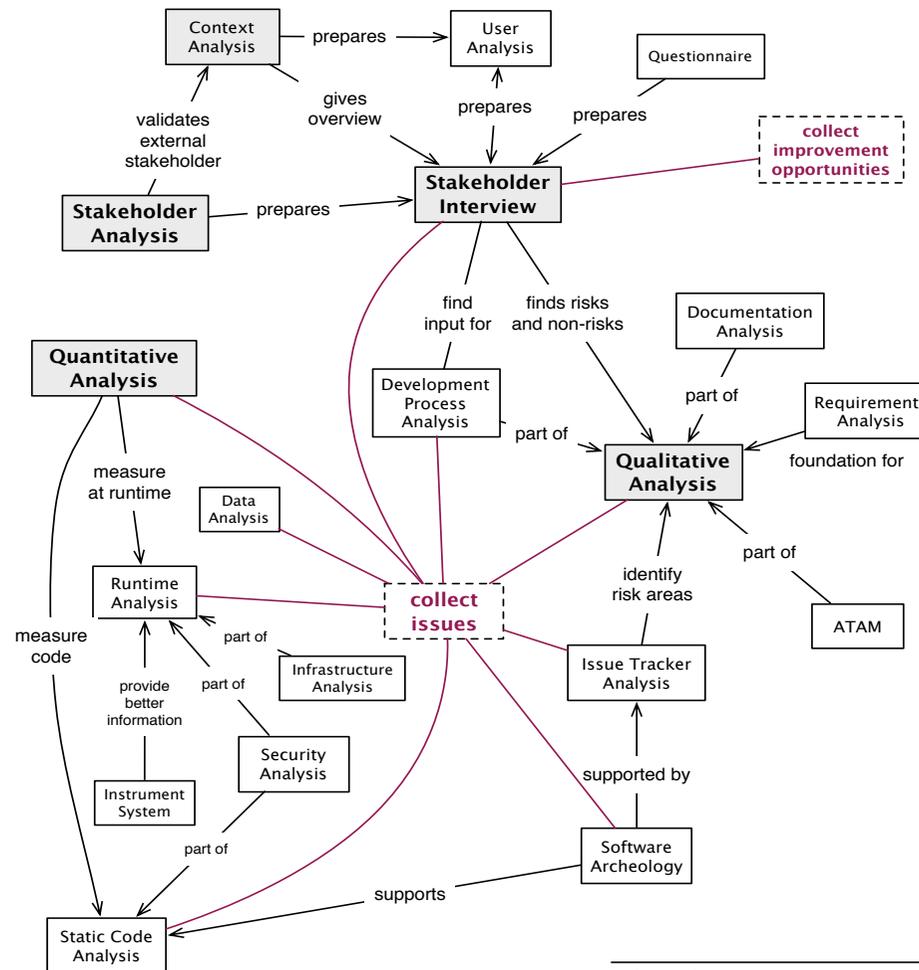
Fazit (1)



Suchraum verbreitern – bei „code-only“
droht Tunnelblick und Mikroskopfalle.

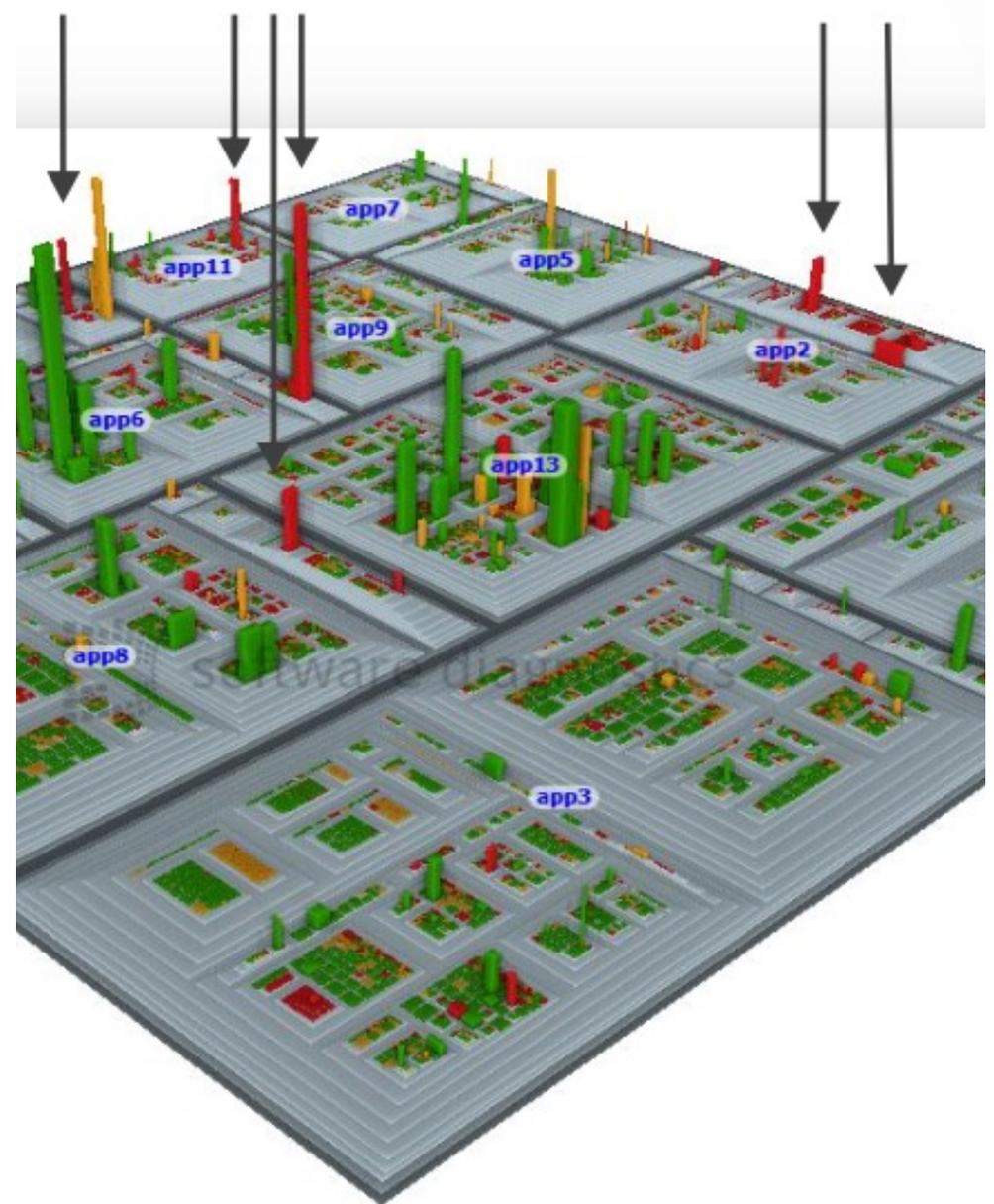


Fazit (1_{ff})



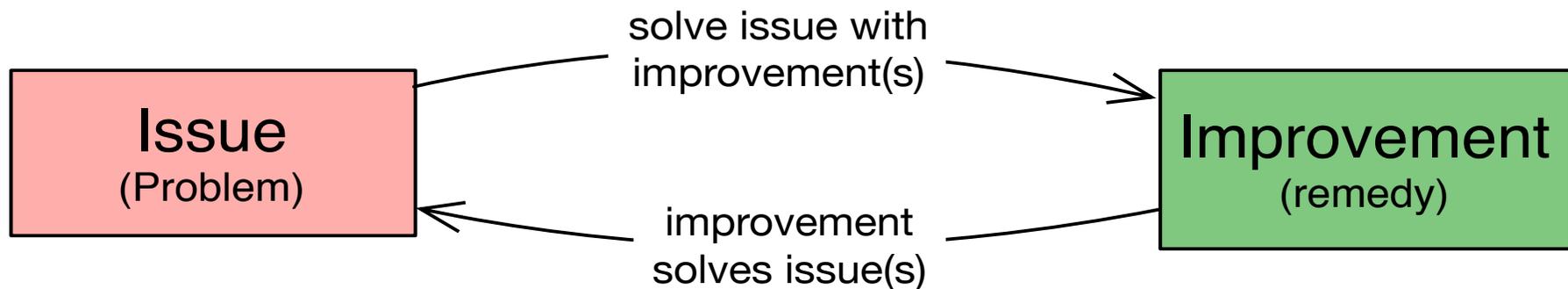
Fazit (2)

(Statische)
Metriken
korrelieren –
insbesondere mit
Zeitaufwand und
Fehlerzahl.



Fazit (3)

Sammeln Sie Ideen für Verbesserung, während Sie Probleme suchen.





Dr. Gernot Starke

gernot.starke@innoQ.com

<http://innoq.com>