

Methodical Improvement of Software Systems

Dr. Gernot Starke

innoQ

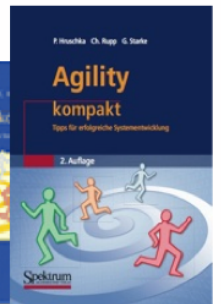
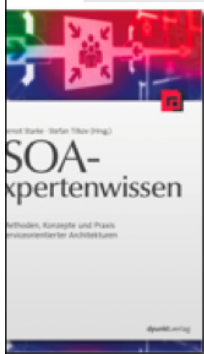
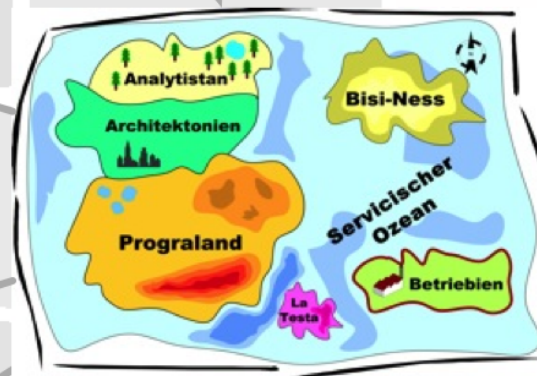
aim⁴²



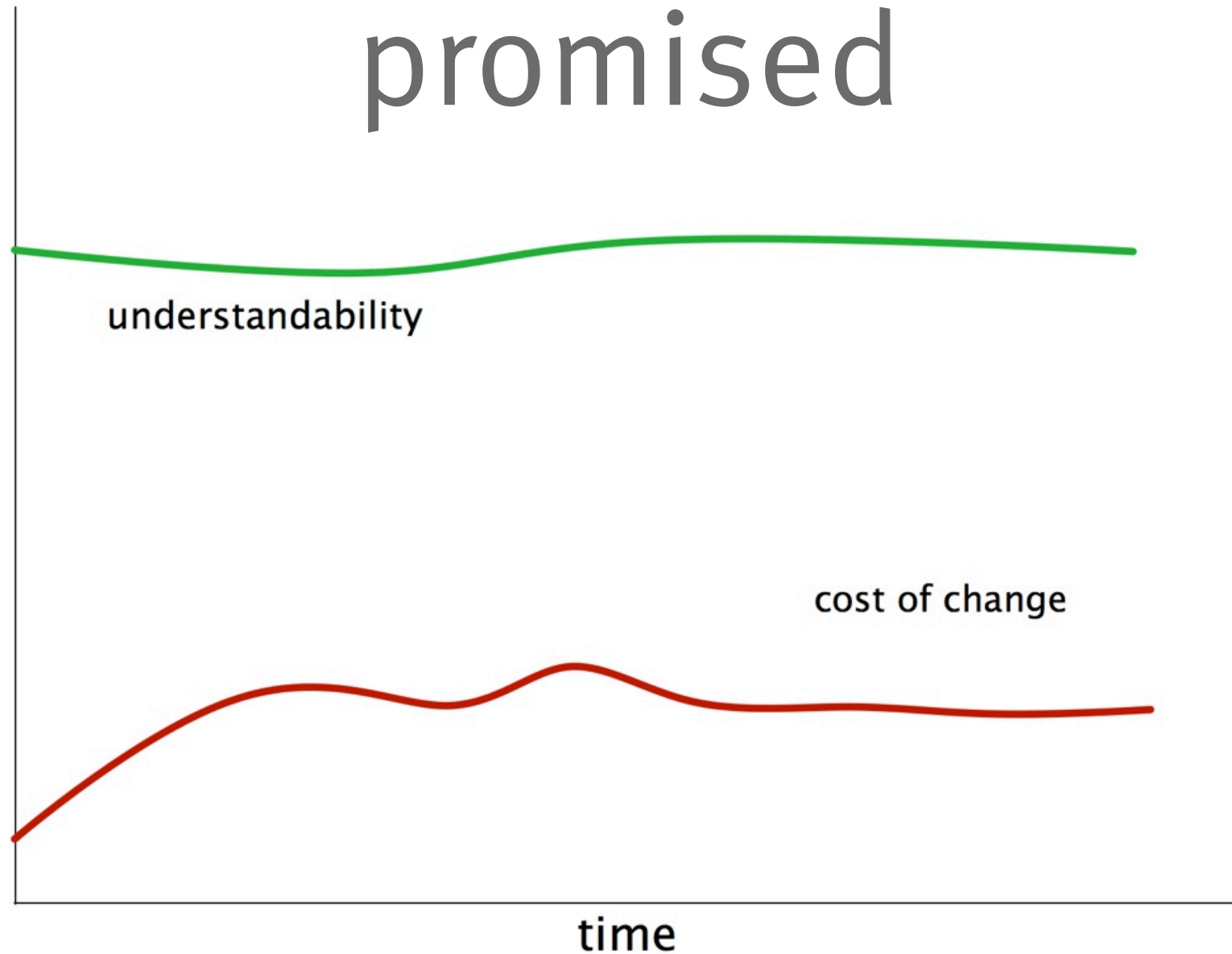
innoQ Fellow



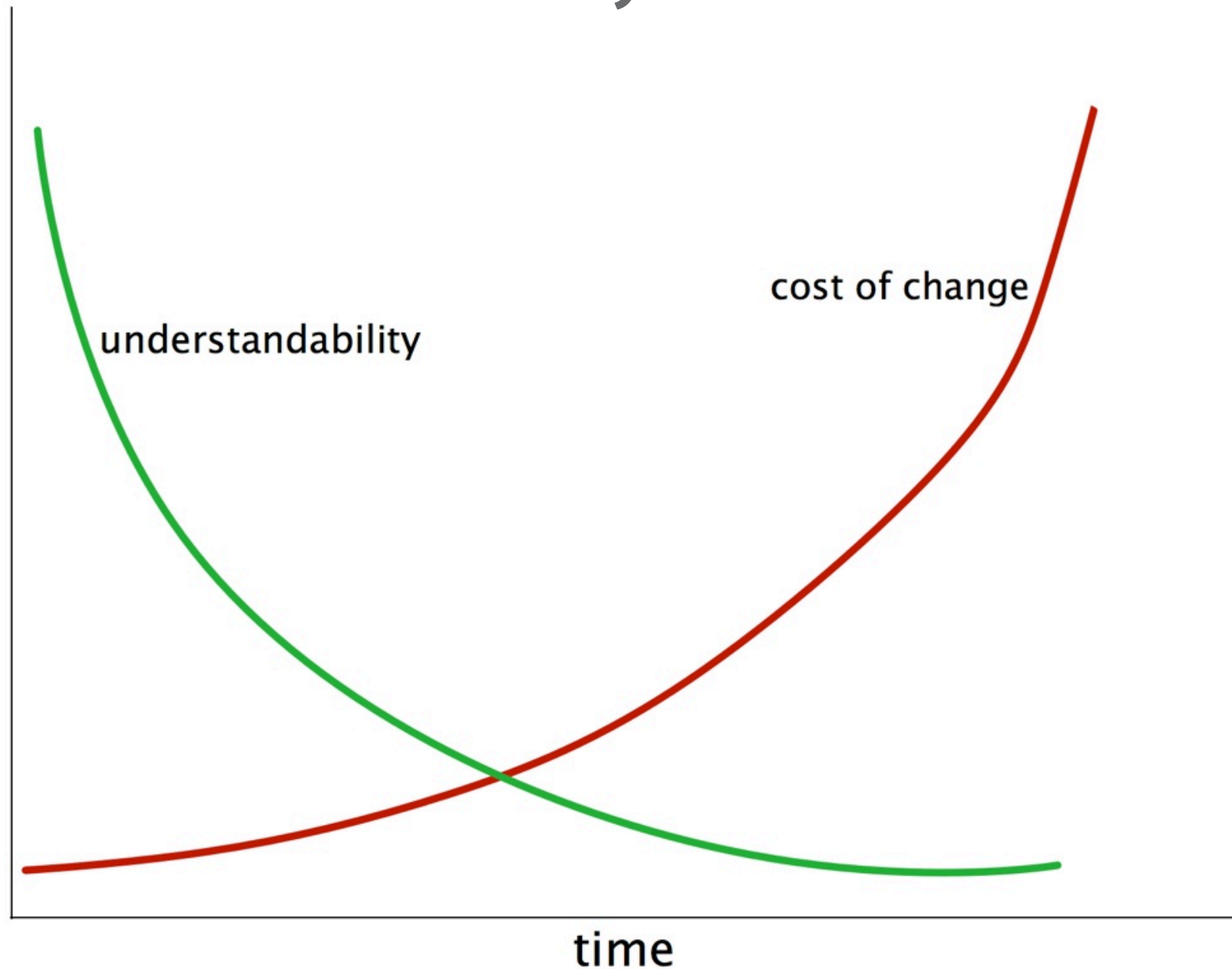
aktiv



What Software-Engineering promised



What Reality Delivered



Thesis:

Software

Education focused
on „build-from-scratch“
of systems

Thesis:

**Business requires more
maintenance
competence**

Thesis:

Improvement
of Systems
is more than Refactoring
of single classes

Thesis:

**Management
responsible for budget
ignores
architecture principles**

Thesis:

Architects improving
systems need to
„talk business“



Architecture Improvement Method



- architecture
- code
- runtime
- organization



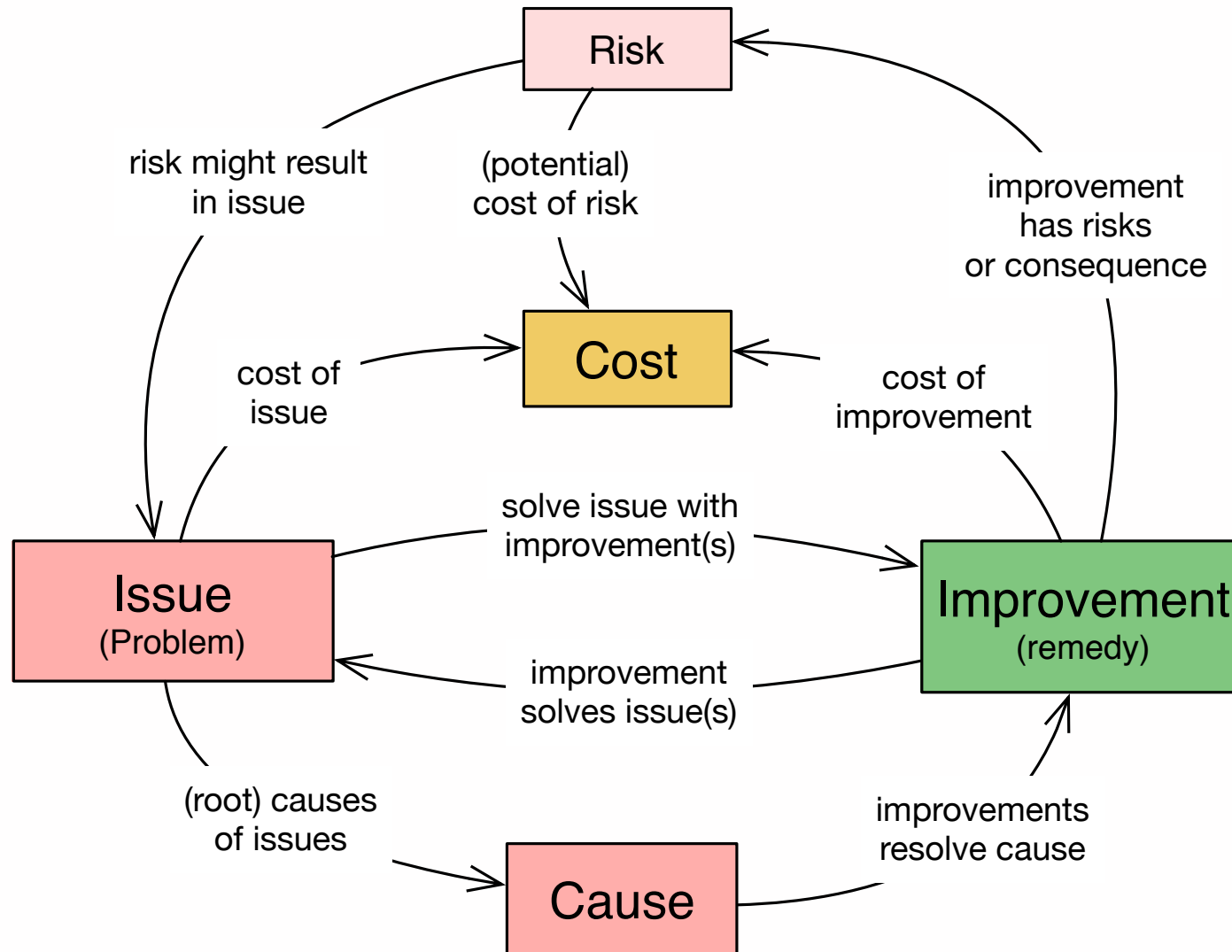
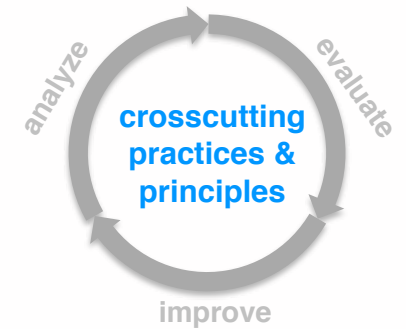
determine „value“ of
problems / risks /
issues and
their remedies



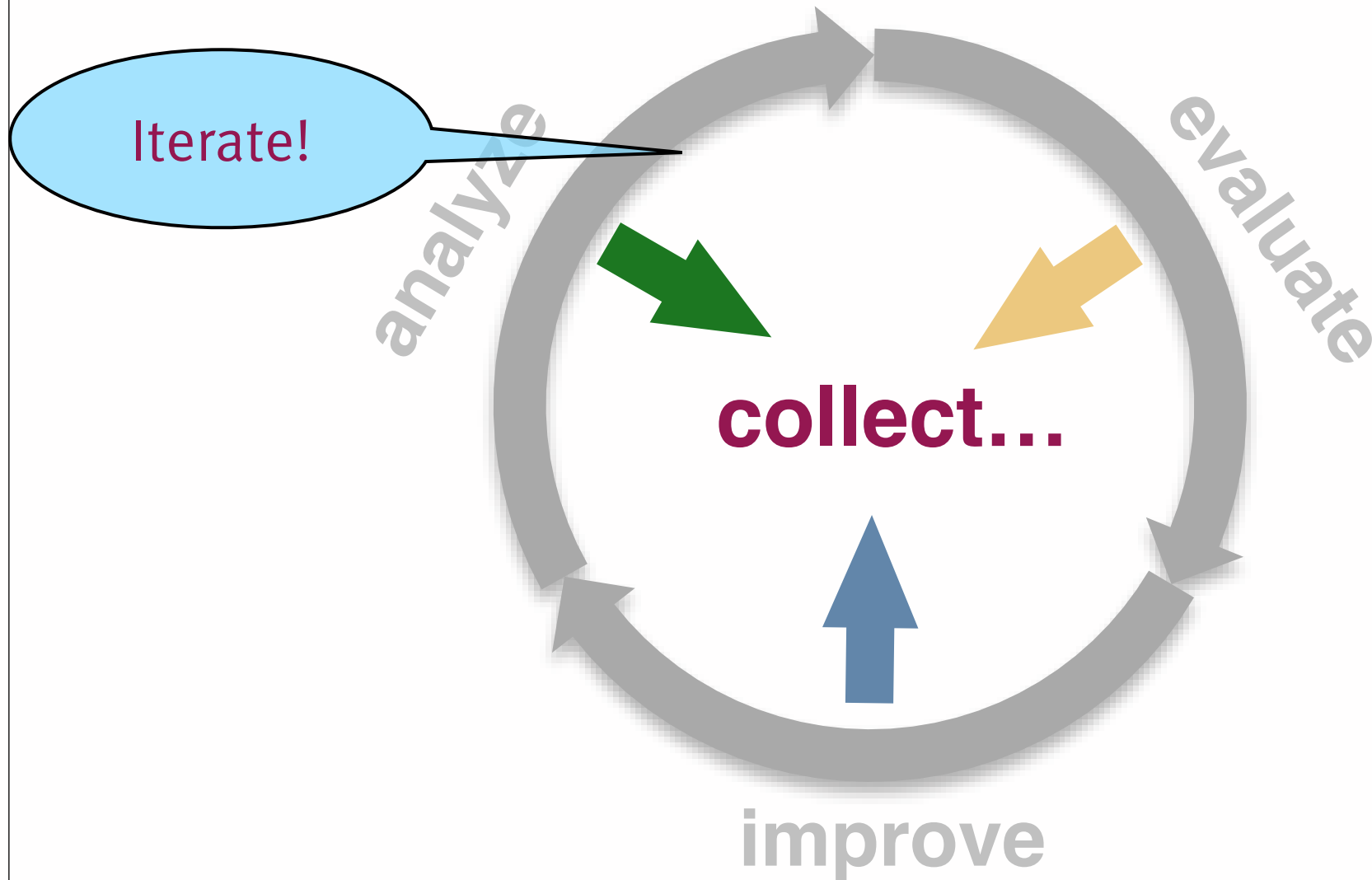
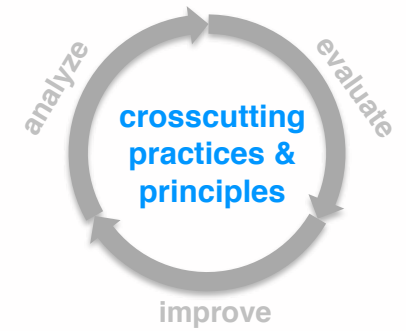


- define improvement strategy
- refactor
- re-architect
- re-organize
- remove debt

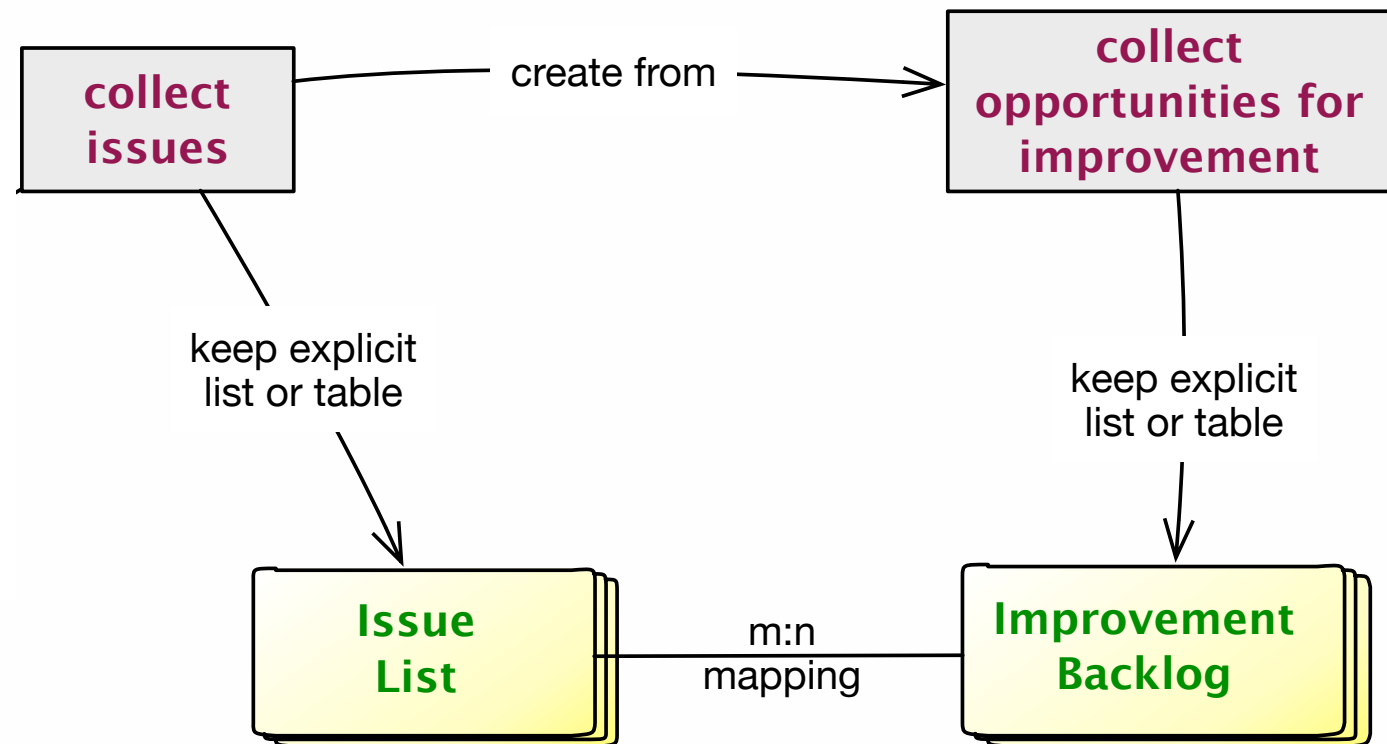
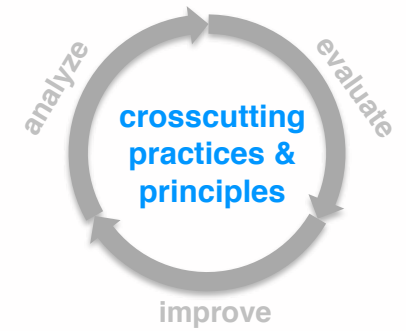
Common Wording



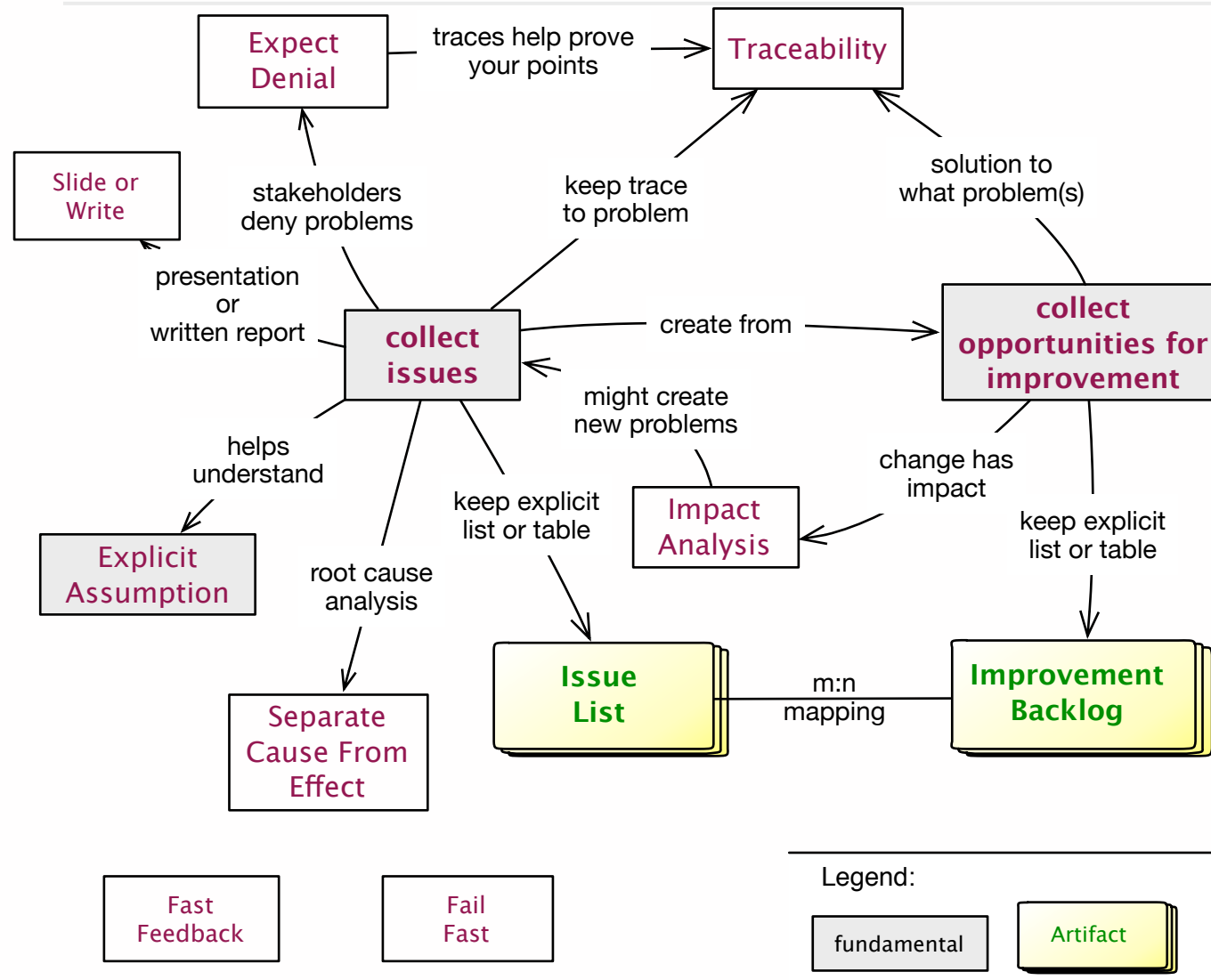
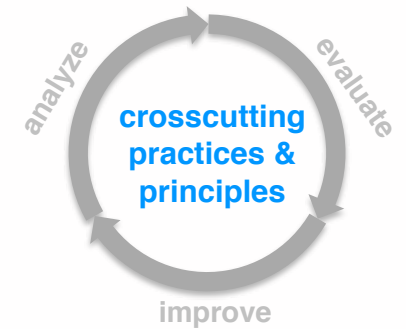
Iterative Approach



Crosscutting...



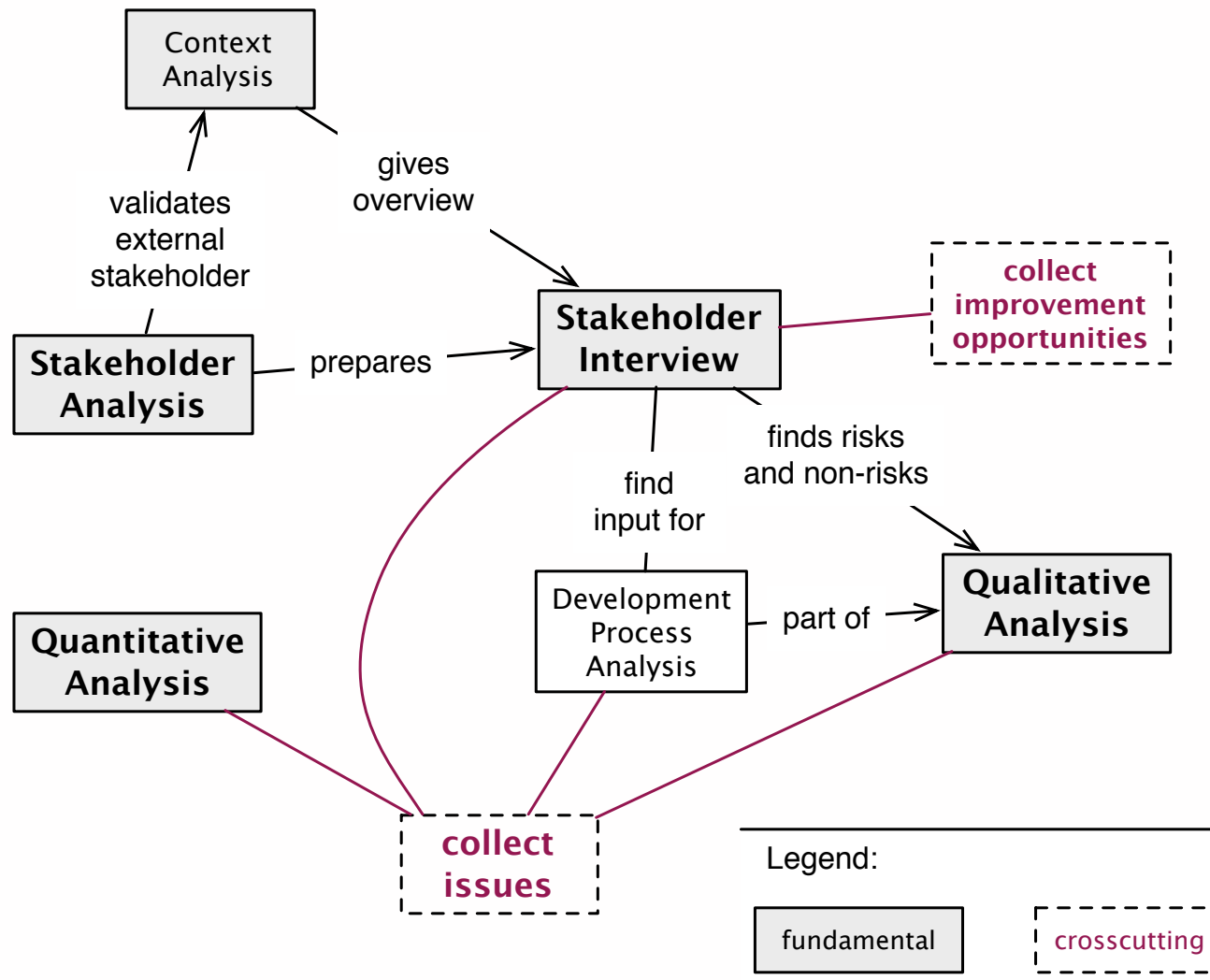
Crosscutting



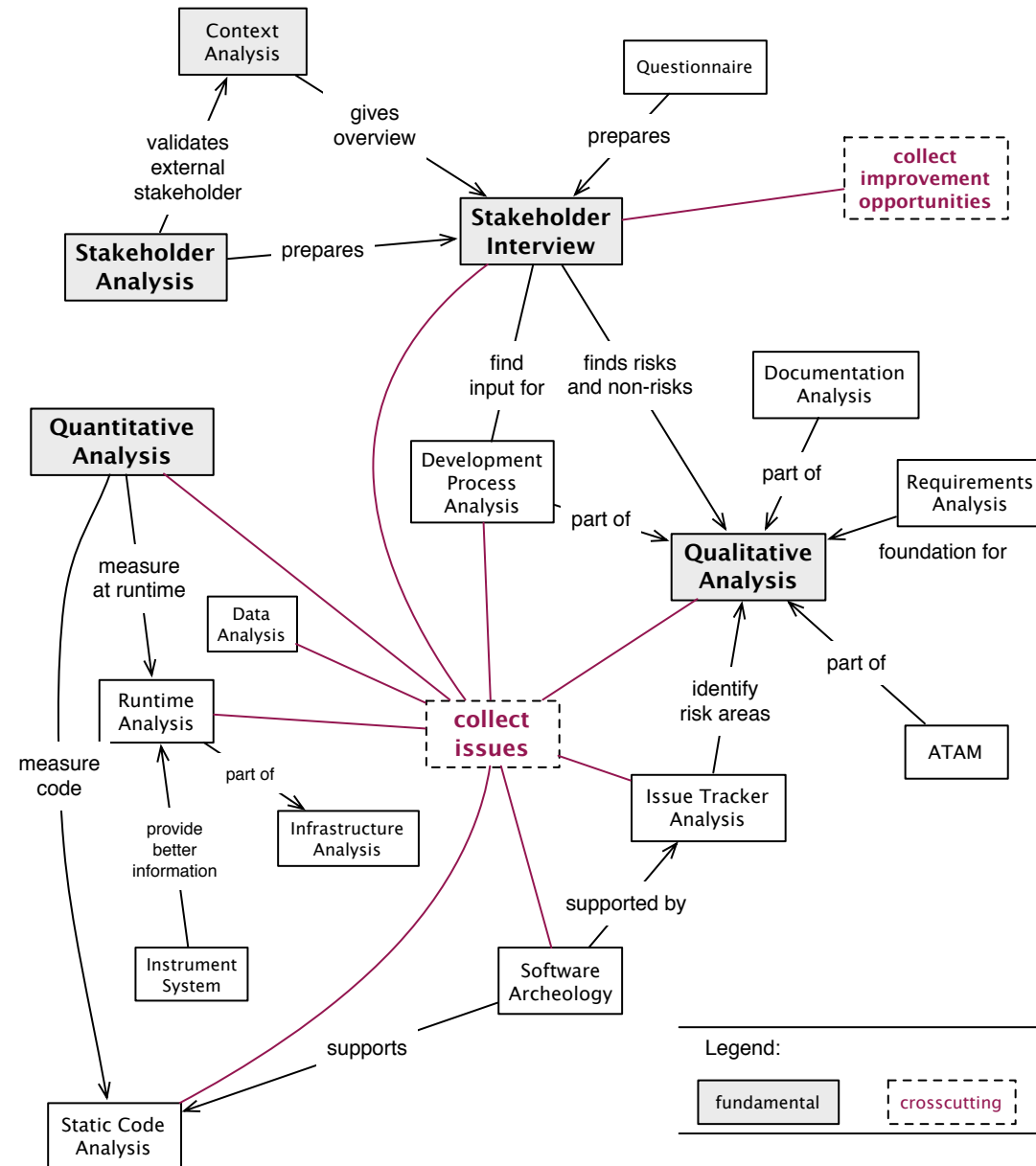
Goals of Analysis...

- › Architectural **understanding**
 - › concepts, structures, decisions + code
- › **Issues** (problems, risks, faults...)
- › Opportunities for **improvements**

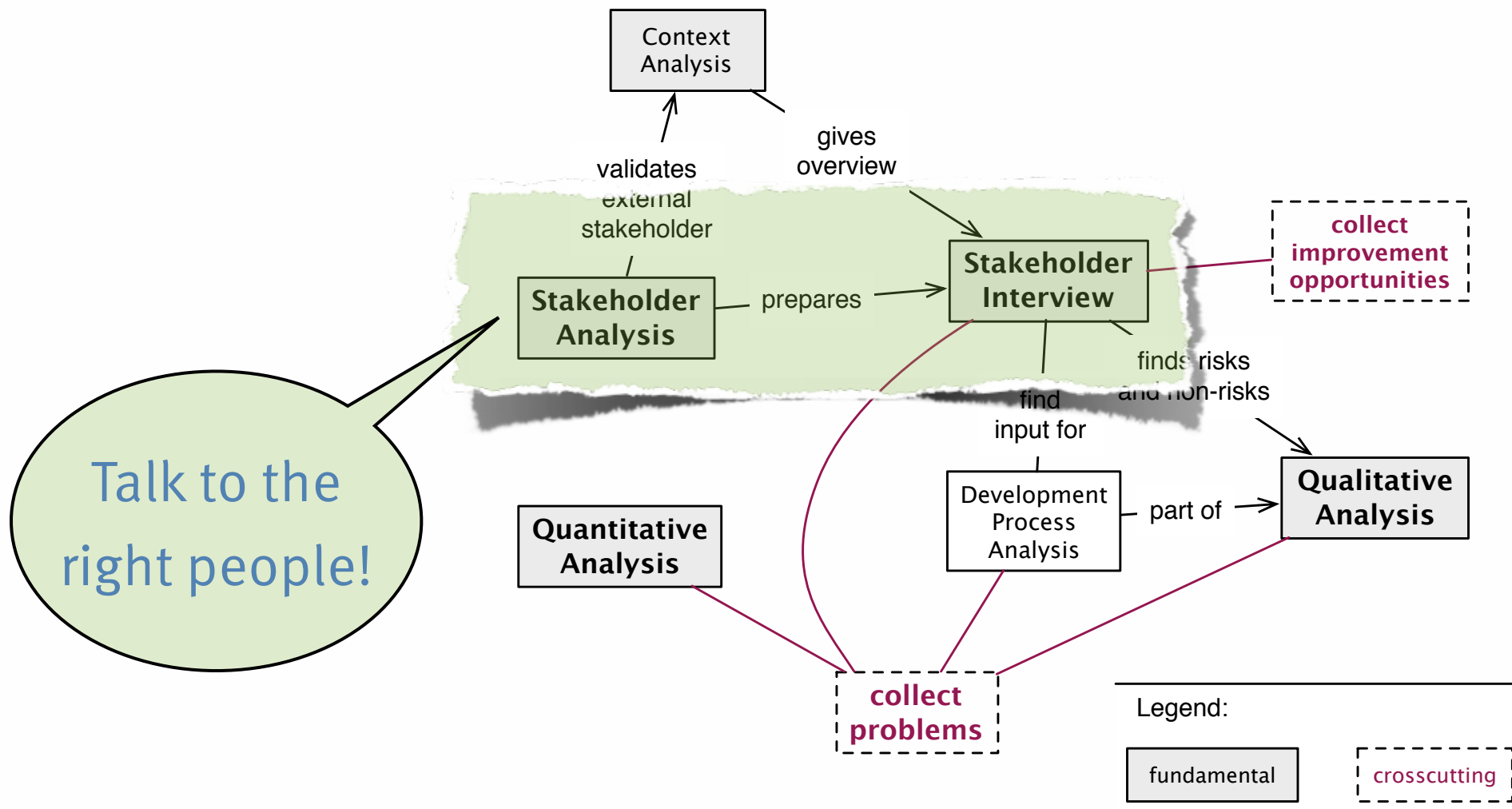
„Analyze“ Overview



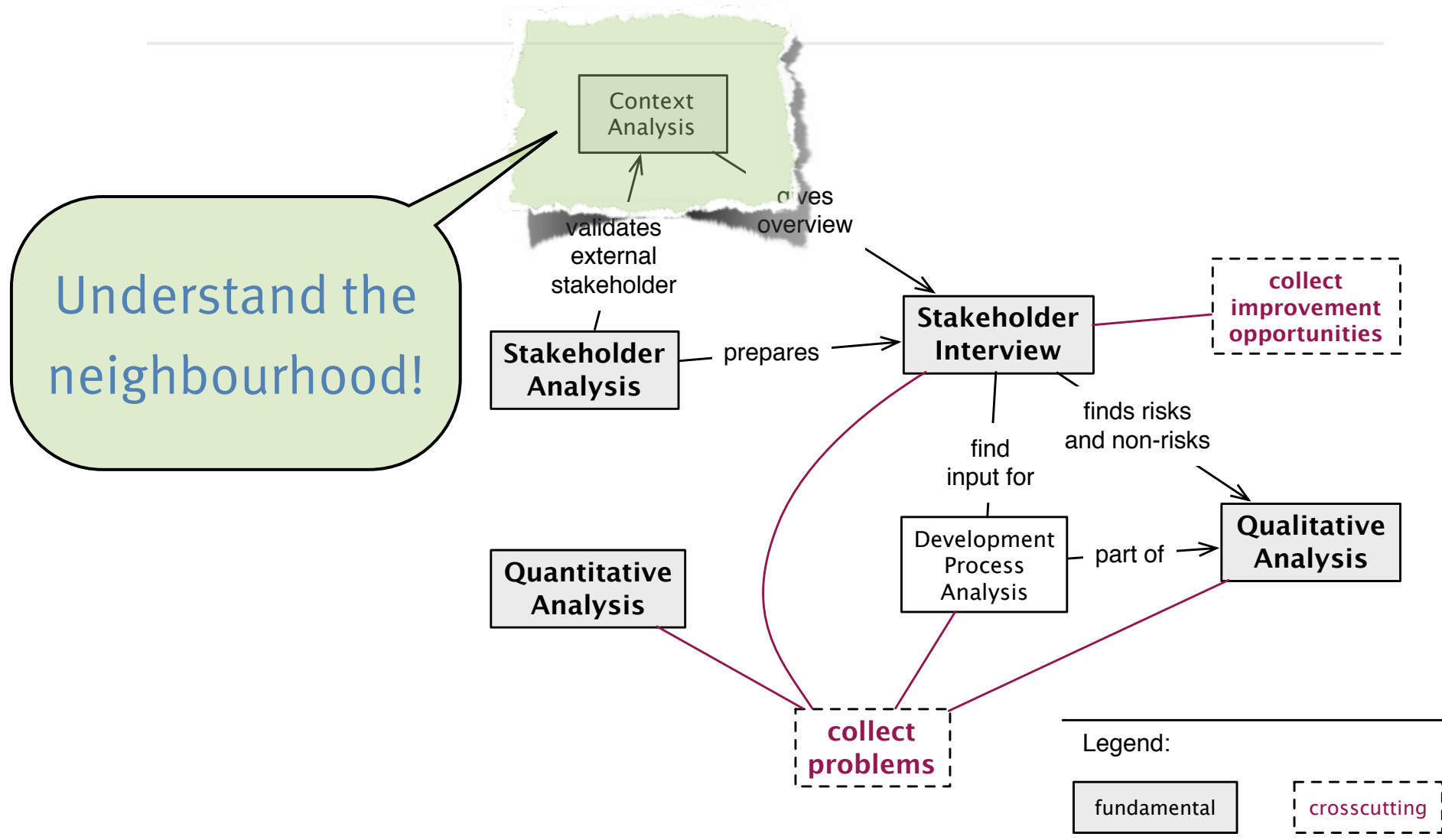
„Analyze“ Details



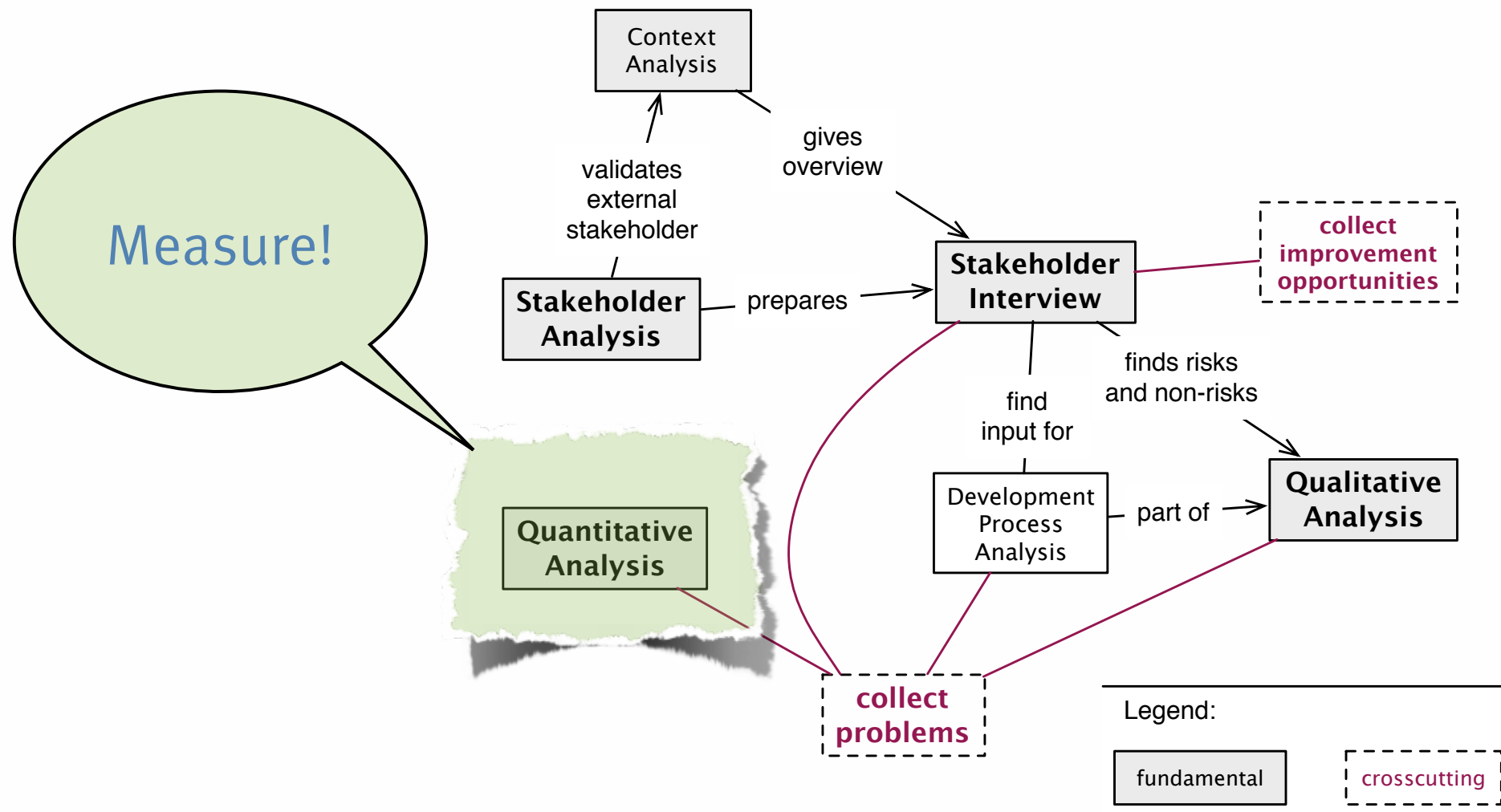
„Analysis“ Overview



„Analysis“ Overview



„Analysis“ Overview



Perishable Food Packaging



- › Embedded software + information systems
- › Regulated domain -> safety critical
- › Goal: Decrease SW development cost

Food: Analysis



-
- › Stakeholder analysis and -interviews
 - › Development Process Analysis
 - › Qualitative Analysis + View-Based-Understanding
 - › Quantitative Analysis, Static Code Analysis
-
- › Central problem areas:
 - › Lack of overview („knowledge islands“)
 - › Low code quality
 - › ad-hoc development: No systematic processes

Food: Analysis (excerpt)



issue (problem)	description	problem-cost
time-to-market	> 6 month (!) from business or government requirement to production	sales loss might be > 1M\$
production log data loss	architecture does not ensure complete production logs - data records might get lost! Large volumes of perishable food could be at risk	> 10-100k \$ per incident
scattered knowledge + low code quality	no synergy effects, no conceptual integrity, no re-use between departments, ...	>5-50k \$ per maintenance update
self-developed OR-mapper	expensive maintenance, high know-how requirements, high deviation in performance	5-10k \$ per maintenance update

Telco: Analysis



-
- › View-Based-Understanding
 - › Data Analysis
 - › (few) stakeholder interviews

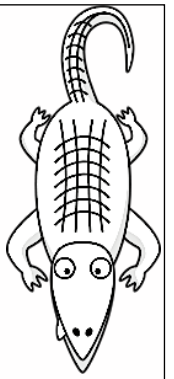
 - › Central problem areas:
 - › BI Reporting highly fragmented & diverse
 - › Report implementation details driven by business experts
(provided data models + SQL query details as „requirements“)
 - › Implementation partially based upon proprietary meta-model



Telco: Analysis (excerpt)

problem / risk	description	problem-cost
high development cost	business benchmarks showed development to be overly expensive (and slow)	per report-type 50-200%
non-transparent software and data architecture	of >50 developers and BI experts, only very few understood whole DWH	
vendor-lock-in	proprietary tools implemented to process (proprietary) meta-model, high yearly license cost,	50 k€ license fee / yr, O(1000) dev-hrs wasted
developer exodus	core developers upset as company announced large outsourcing deal, (nearly) annihilating internal development	6-18 month without new business features

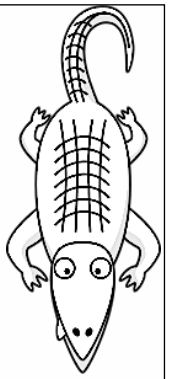
Croc: Sales & ERP Provider



Company name changed due
to anonymity requirements!

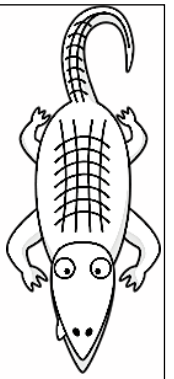
- › Niche provider for sales & ERP „standard“ solution
- › Origin in „perishable“ market - but growing
 - › 80% of clients: low-margin-high-volume
 - › 20% of clients: low-volume-very-high-margin
- › Original idea: Universal-Core + Configuration
- › Starting point:
low (dev + runtime) performance

Croc: Analysis

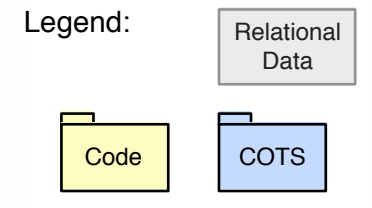
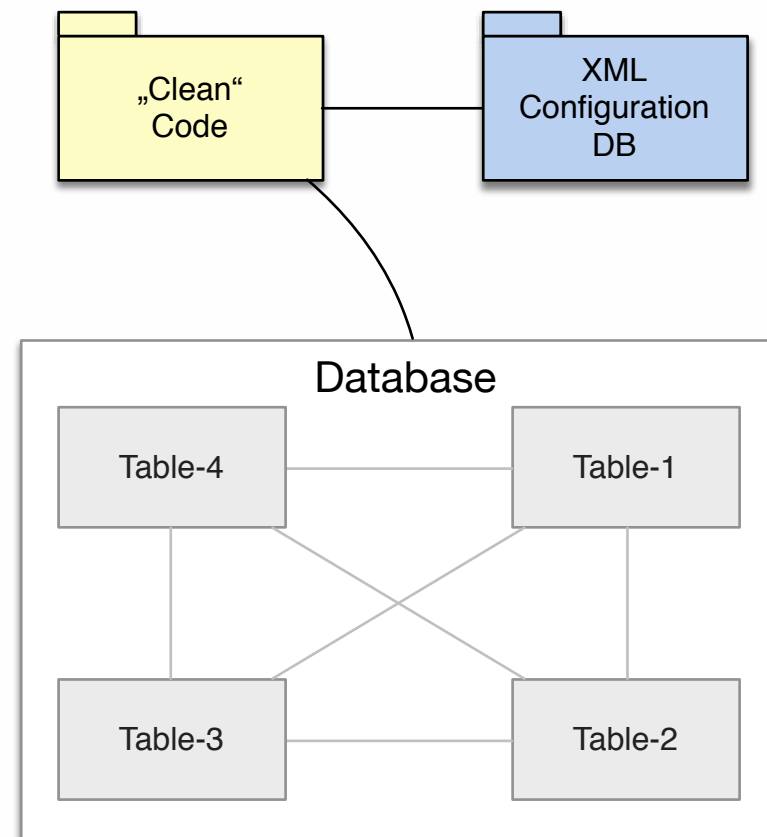


-
- › Brief stakeholder analysis and -interviews
 - › Static Code Analysis
 - › Runtime Analysis
 - › Data Analysis (including data model)
 - › Central problem areas:
 - › Excellent code quality („clean code“) - but very few unit tests
 - › Extremely high configurability of everything
 - › >150 developers with extremely different options

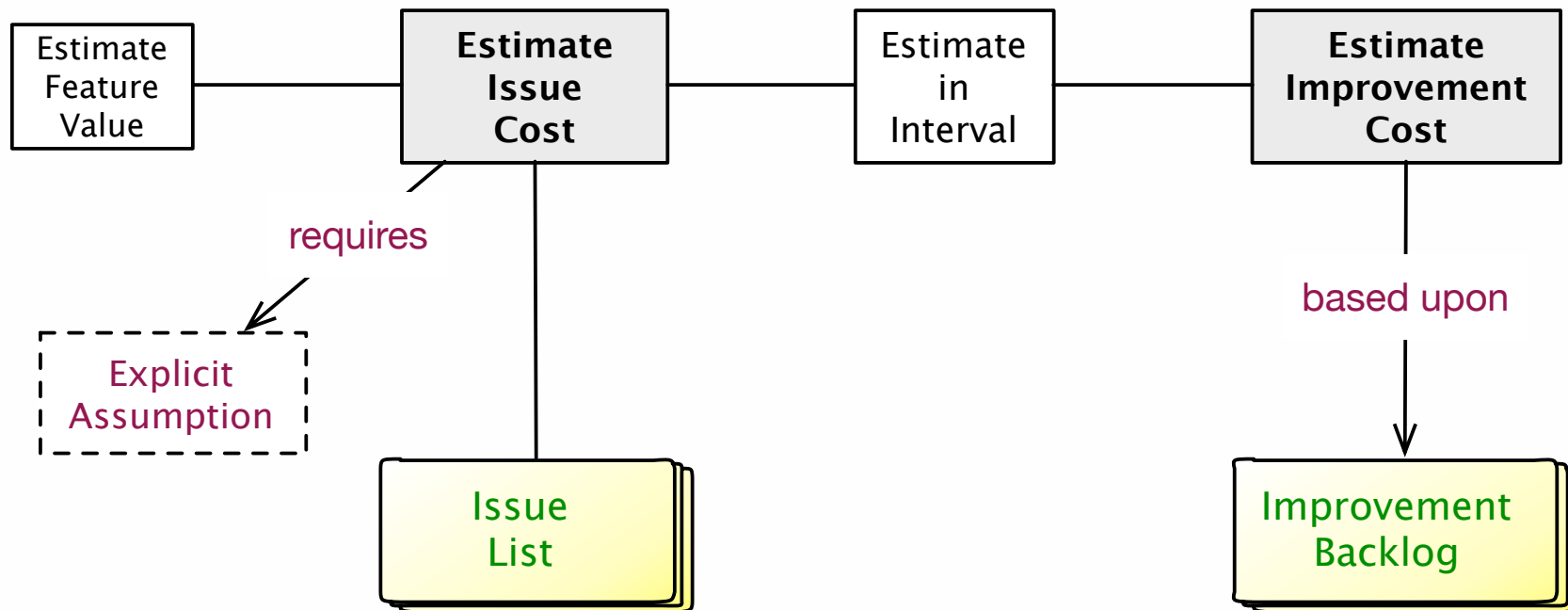
Croc: Analysis (3)



- › Few key tables with 500-700 columns (!!) each.
- › Stores complete application state - including cursor position.



„Evaluate“ Overview



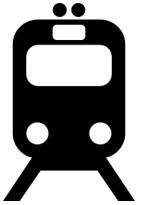
Legend:

fundamental

crosscutting

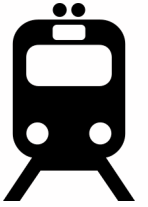
Artifact

Rail Transport Provider



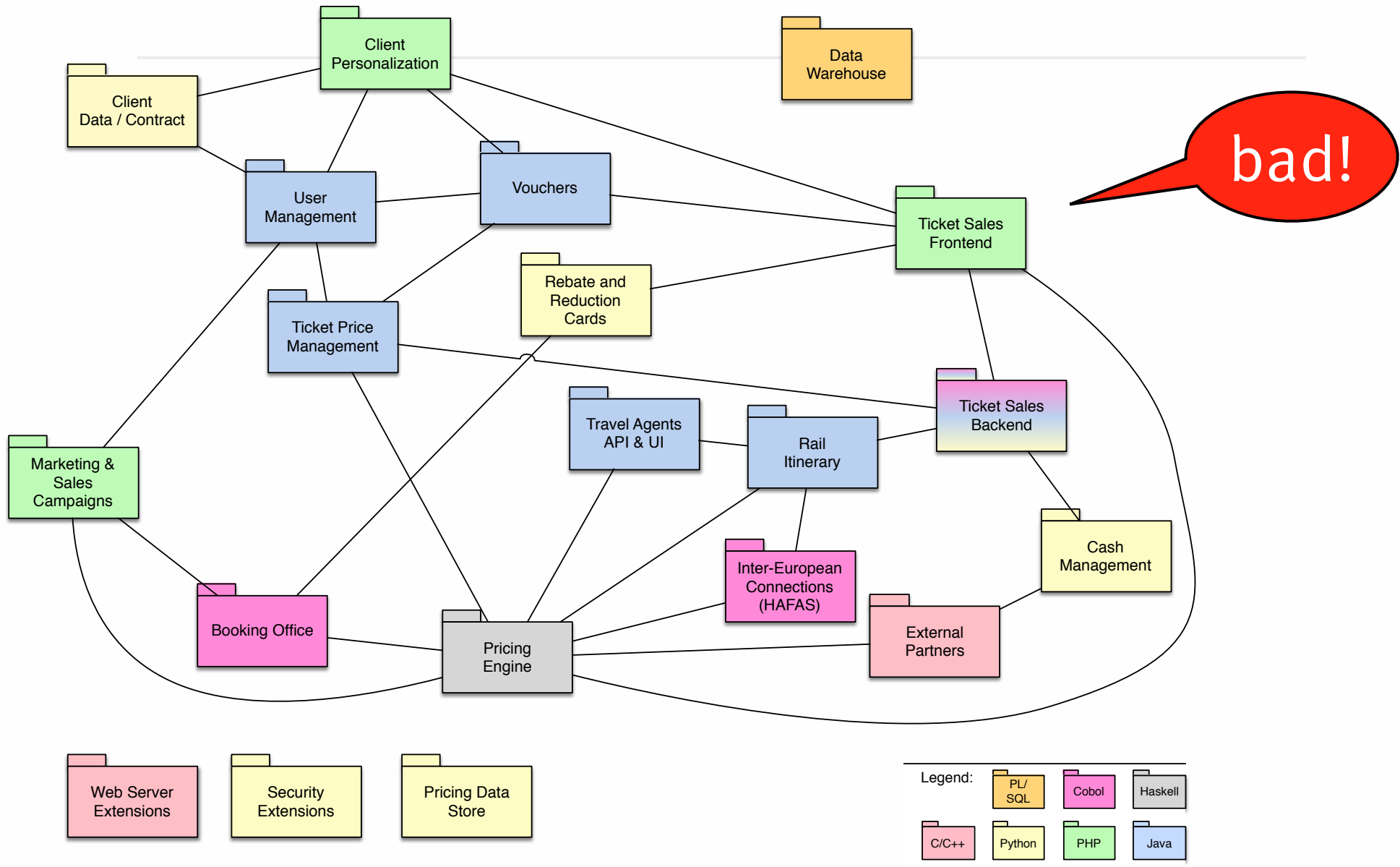
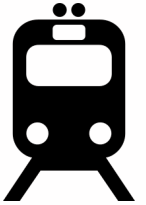
- › Heterogeneous IT landscape
- › Problem areas:
 - › 6-12 month from initial business requirement to production („time-to-market“)
 - › Stability, reliability
 - › Performance

Rail - aim42 Analysis

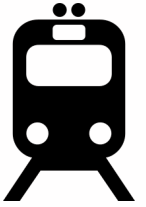


- › Stakeholder Analysis + -Interviews
 - › yielded several problems + problem-areas
 - › Issue Tracker Analysis + Software Archeology
- › Qualitative (ATAM-like) Analysis
- › Static Code Analysis
- › Development Process Analysis

Rail (1): Overview

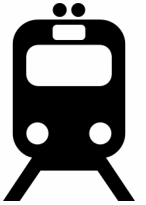


Rail (2): Challenges



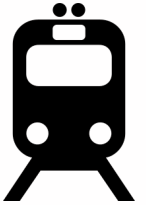
- › Embrace new sales channels (mobile)
 - › requires (much) higher availability
- › Marketing demands rapid price adjustments

Rail (4): Analysis (excerpt)



issue (problem)	description	problem-cost
time-to-market	6-12 month (!) from business requirement to production	
configuration of certain ticket types crashes backend	when either end-users or sales-clerks configure specific ticket-types (groups > 5 persons, more than one rebate reason, border crossing or >2 train changes), several backend processes crash	
know-how drain in development	many dissatisfied developers and business experts leave (development) organization, migration from internal to external development, fix-price projects	

Rail (5): Evaluation (excerpt)

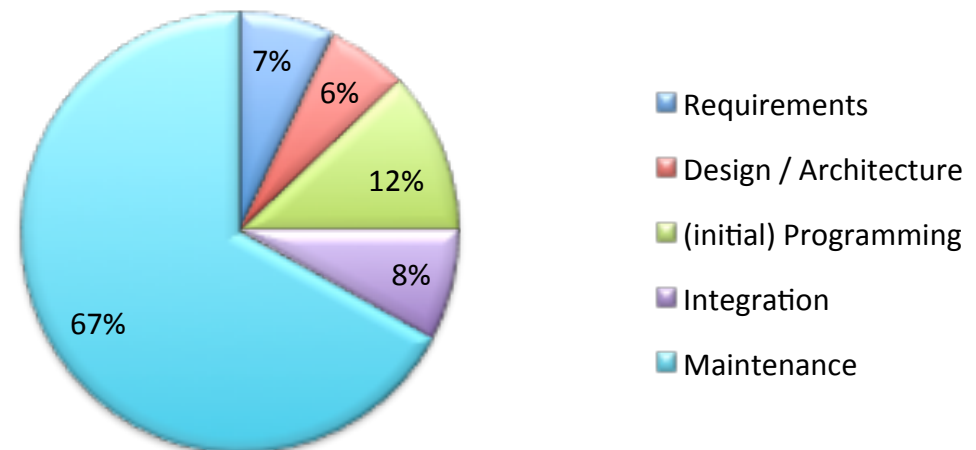


What's the (additional) cost of „heterogeneity“?

1. Explicit assumptions

- Heterogeneity „costs“ in all phases
- Phase effort is known

Cost Distribution for Software



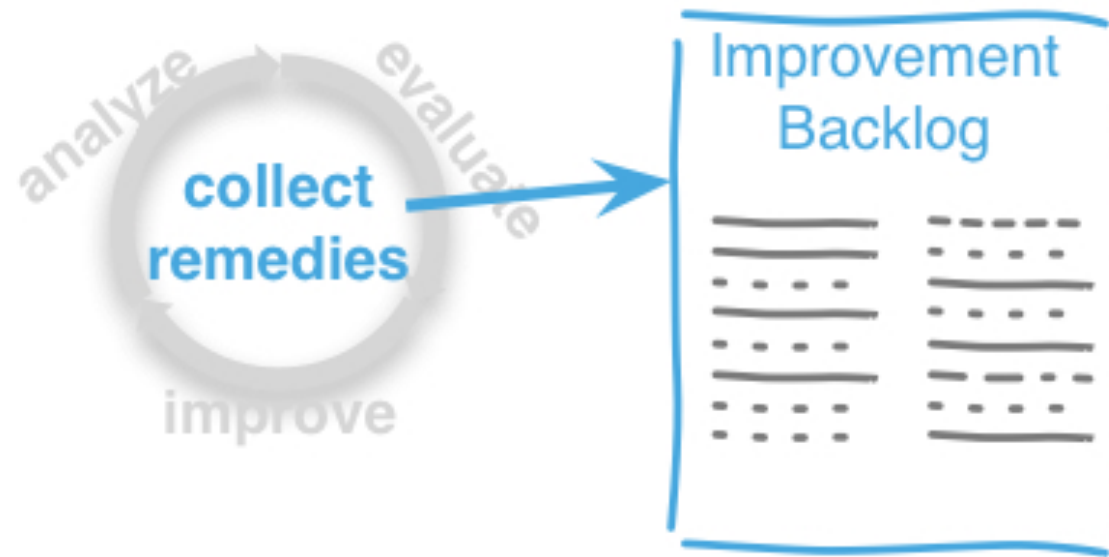
Rail (6)...

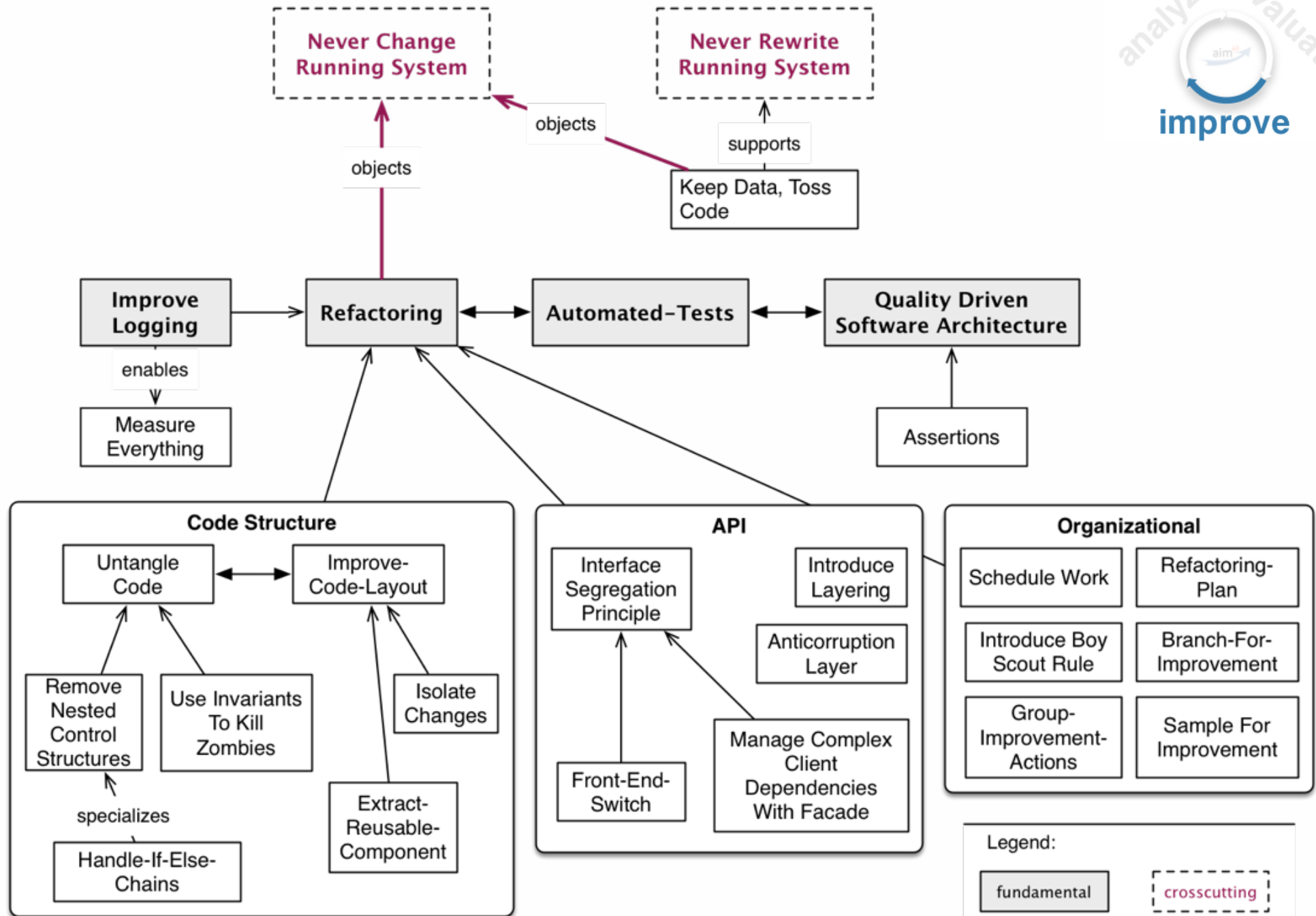


Collected tasks in which
additional effort might occur..

	A	B	C	D	E	F	G	H	I
1			Proportion	added effort			1.000 €	min	max
2				min	max			1.017,78 €	1.204,56 €
3									
4	Requirements		7%				70 €	70,00 €	70,00 €
5									
6	Design/Architecture		6%				60 €	60,42 €	61,20 €
7	10%	Additional effort at interfaces		5%	15%			0,30	0,90
8	10%	decisions across technologies		2%	5%			0,12	0,30
9	80%	Others							
10									
11	Programming		12%				120 €	122,40 €	145,68 €
12	2%	Setup, updates of environments		5%	100%			0,12	2,40
13	2%	Research, Setup		5%	20%			0,12	0,48
14	10%	searching bugs, testing		3%	100%			0,36	12,00
15	5%	Efficient solution of detailed problems		-10%	-40%		-	0,60	- 2,40
16	10%	Solution of standard problems		10%	50%			1,20	6,00
17	20%	Team-internal coordination		5%	30%			1,20	7,20
18	51%	Others							
19									
20	Integration / Test		8%				80 €	83,40 €	113,80 €
21	5%	integrate Components		5%	100%			0,20	4,00
22	30%	perform integration tests		5%	50%			1,20	12,00
23	20%	evaluate integration tests		10%	50%			1,60	8,00
24	10%	create/maintain test infrastructure		5%	80%			0,40	6,40
25	35%	Others							
26									
27	Maintenance / Operations		67%				670 €	681,56 €	813,88 €
28	3%	keep developer reserve		5%	20%			1,01	4,02
29	5%	find and incorporate developers		10%	30%			3,35	10,05
30	1%	Versions- and Security-Updates		3%	10%			0,20	0,67
31	1%	selection & maintenance of runtime environment		10%	100%			0,67	6,70
32	3%	Configuration, Installation		5%	70%			1,01	14,07
33	0,50%	Monitoring, Logging		5%	10%			0,17	0,34
34	5%	Identify and solve issues		1%	100%			0,34	33,50
35	2%	Scaling/Clustering		5%	15%			0,67	2,01
36	1%	Packaging, Deployment-preparation		2%	10%			0,13	0,67
37	30%	Enhancements, Modifications		2%	30%			4,02	60,30
38	49%	Others							

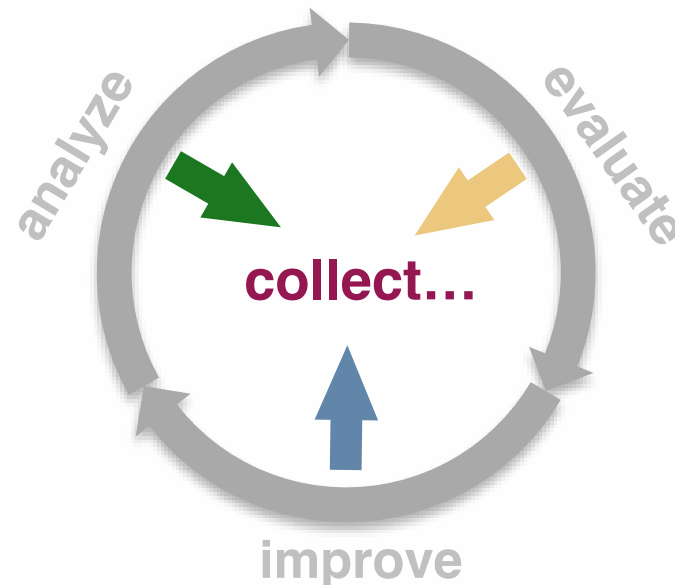
„Improve“ Overview





Systematic Improvement

... is feasible - requires skills, discipline and (some) money.



Questions? Comments?

Dr. Gernot Starke, @gernotstarke
gernot.starke@innoq.com
<http://gernotstarke.de>

Alexander Heusingfeld, @goldstift
alexander.heusingfeld@innoq.com
<https://www.innoq.com/en/staff/alex>



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Offices in:

Berlin
Darmstadt
München

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116

info@innoq.com