Computer Support for the Analysis and Improvement of the Readability of IT-related Texts

Matthias Holdorf, 21.11.2016, Munich

Software Engineering for Business Information Systems (sebis)
Department of Informatics
Technische Universität München, Germany

wwwmatthes.in.tum.de

Administrative Setup

- **Chair**: Software Engineering for Business Information Systems
- **Company**: QAware GmbH

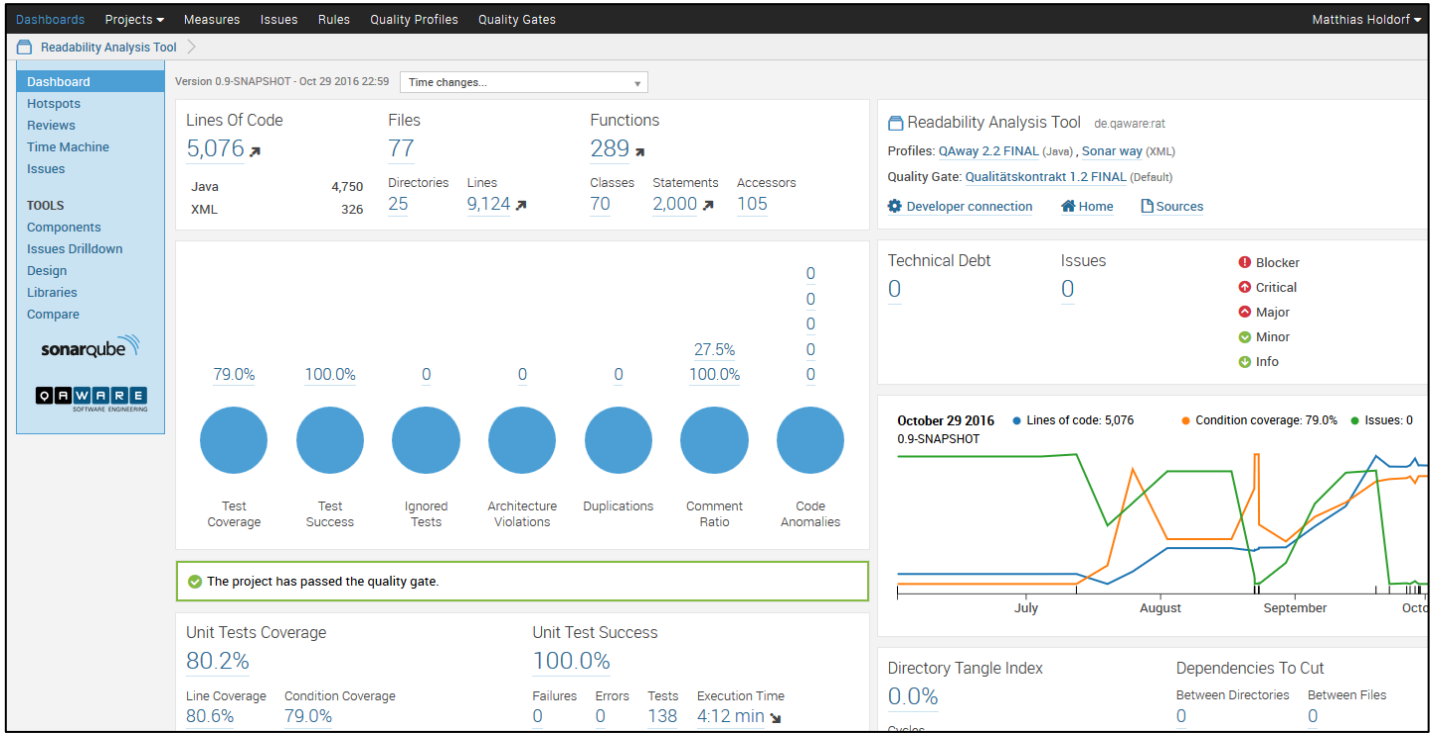
- **Title**: Computer Support for the Analysis and Improvement of the Readability of IT-related Texts
- **Advisor**: Bernhard Waltl (b.waltl@tum.de)
  Andreas Zitzelsberger (andreas.zitzelsberger@qaware.de)

- **Author**: Matthias Holdorf (matthias.holdorf@gmail.com)
- **Start**: 15. May 2016
- **Submission:** 15. November 2016

Motivation

Information Technology (IT) consists of two words: **Information** and **technology**.

Today we apply extensive measurements of the quality of IT artifacts, e.g. source code. However, we do not apply the same rigor in quality assurance of texts.

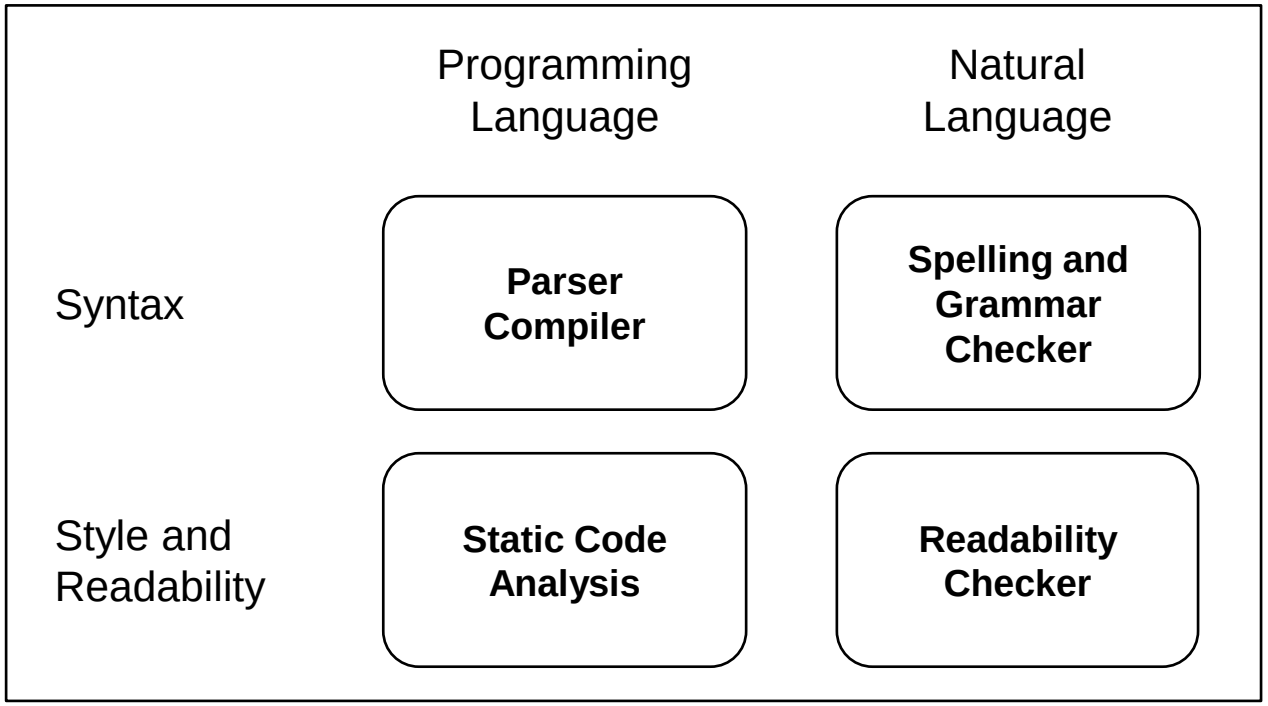


Motivation

Information Technology (IT) consists of two words: **Information** and **technology**.

Today we apply extensive measurements of the quality of IT artifacts, e.g. source code. However, we do not apply the same rigor in quality assurance of texts.

.



Research Process

Environment

People & Organizations

QAWARE SOFTWARE ENGINEERING

- Project Software Company
- Approx. 100 Member of Staff

Technologies

sonarqube

Business Needs

IS Research

Develop / Build

RAT

- Readability Analysis Tool (RAT)

Refine/Assess

Justify / Evaluate

- Interviews & Survey
- Case Study
- Controlled Experiment
- Corpus Analysis
- Performance
- Precision and Relevance

Applicable Knowledge

Knowledge Base

Foundation & Methodologies

- Spell and Grammar Checker
- Style and Readability Checker
- Controlled Language Checker
- Simplification and Paraphrasing

Frameworks

UIMA

DKPro Core

docx4j

Application in the Appropriate Environment

Addition to the Knowledge Base

RAT GitHub Repository: https://github.com/qaware/readability-analysis-tool

A framework for information system research [He04]

Flesch Reading Ease [Fl48]

$$FRE = 206{,}835 - 1{,}015\left(\frac{\text{Total Words}}{\text{Total Sentences}}\right) - 84.6\left(\frac{\text{Total Syllables}}{\text{Total Words}}\right)$$

MULTILINT a grammar and style checker developed for technical documentations in the automotive sector. [Sc98]

DeLite a readability checker which automatically assesses the linguistic accessibility of web documents. [vHH08]

TextLint a style checker that detects common errors in scientific writing. [PRR10]

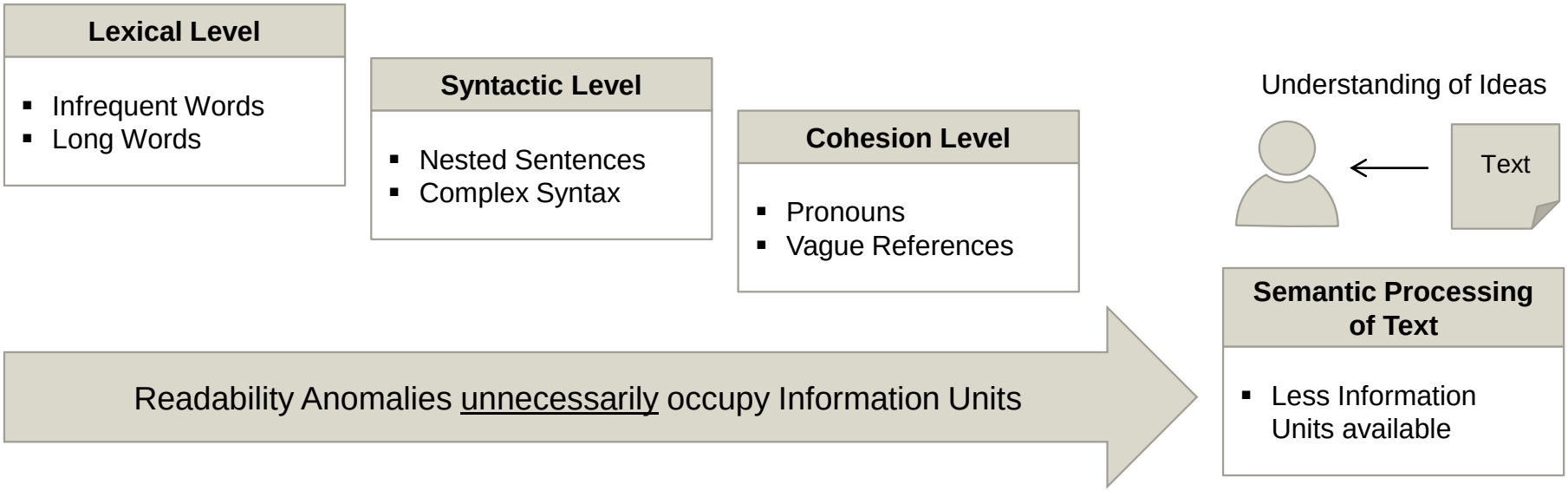
Smella transferred the concept of code smells to requirement engineering, and investigated violations of language criteria in requirement artifacts derived from ISO 29148. [Fe16]

Working Memory and Readability Anomalies

Working memory refers to „the temporary storage of information in connection with the performance of other cognitive tasks such as **reading**, problem-solving or learning.” [Mi56]

Miller found that we can only store 7 (±2) information units in our working memory.

How can Readability Anomalies be explained?



Problems caused by difficult-to-read texts in IT

Options	Results
Difficult-to-read texts take more time to read.	100.00%
	46
I do not understand parts of the content in difficult-to-read texts.	80.43%
	37
The editing of difficult-to-read texts takes more time.	71.74%
	33
Communication with team members is negatively affected by difficult-to-read texts.	69.57%
	32
Communication with customers is negatively affected by difficult-to-read texts.	65.22%
	30
Writing texts that are easy to read is difficult for me.	19.57%
	9
A text that is difficult to read does not cause problems, or if it does, they are not worth mentioning.	0.00%
	0
Total	46

Results of a survey in QAware

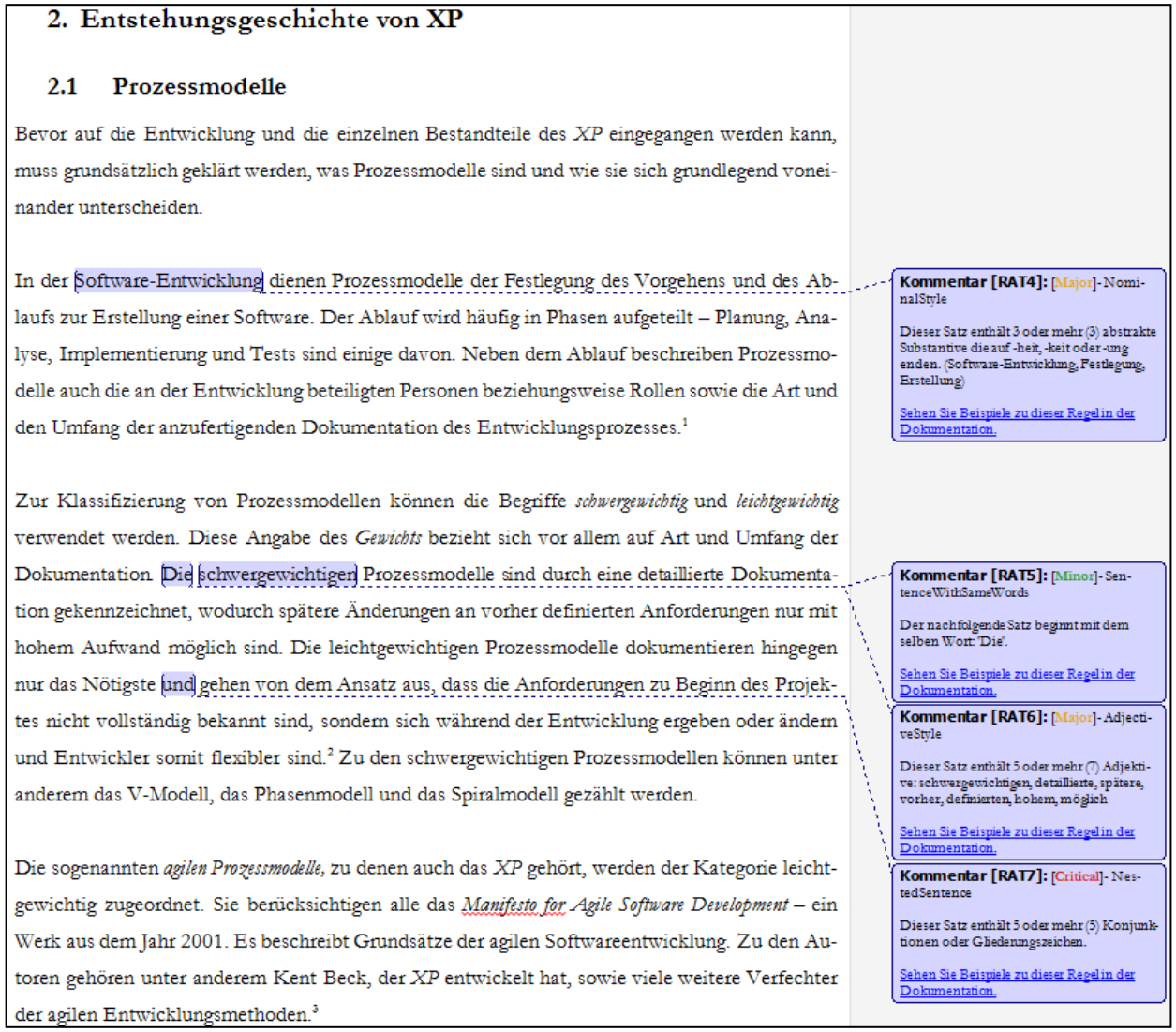
IT-related Texts

#	Categories of texts	Including types of texts
1	Commercial Document	Business Proposal, Tender, and Contract
2	Professional Concept	Rough Concept and Functional Concept
3	**Technical Concept**	Rough Concept and Technical Concept
4	Professional Documentation	User Documentation, Professional Interface Description, Project Manual, Meeting Protocol and Status, Interims and Project Report
5	**Technical Documentation**	Developer and Administrator Documentation, Specification of Systems, Architecture or Interfaces Description, Test Manual, Coding Guideline, Meeting and Status Protocol, Interims and Project Report
6	Presentation	External, Internal, Professional, and Technical Presentations
7	Scientific Article	Paper and Article in Journals or Magazines
8	Online Text	Advertising Text, Blog Post, and Text on the QAware Website
9	E-Mails	External, Internal, Formal, and Unformal E-Mail

Based on Interview Findings and Corpus Inspection

Annotations in Microsoft Word

- 22 Readability Rules
- Configuration of Readability Rules via XML (severity, threshold, or enabled)
- Hyperlink to in-depth Documentation on GitHub
- Performance:
  20 Seconds for 5,000 Words
  (30 pages of text)



2. Entstehungsgeschichte von XP

2.1 Prozessmodelle

Bevor auf die Entwicklung und die einzelnen Bestandteile des *XP* eingegangen werden kann, muss grundsätzlich geklärt werden, was Prozessmodelle sind und wie sie sich grundlegend voneinander unterscheiden.

In der Software-Entwicklung dienen Prozessmodelle der Festlegung des Vorgehens und des Ablaufs zur Erstellung einer Software. Der Ablauf wird häufig in Phasen aufgeteilt – Planung, Analyse, Implementierung und Tests sind einige davon. Neben dem Ablauf beschreiben Prozessmodelle auch die an der Entwicklung beteiligten Personen beziehungsweise Rollen sowie die Art und den Umfang der anzufertigenden Dokumentation des Entwicklungsprozesses.[1]

Zur Klassifizierung von Prozessmodellen können die Begriffe *schwergewichtig* und *leichtgewichtig* verwendet werden. Diese Angabe des *Gewichts* bezieht sich vor allem auf Art und Umfang der Dokumentation. Die schwergewichtigen Prozessmodelle sind durch eine detaillierte Dokumentation gekennzeichnet, wodurch spätere Änderungen an vorher definierten Anforderungen nur mit hohem Aufwand möglich sind. Die leichtgewichtigen Prozessmodelle dokumentieren hingegen nur das Nötigste und gehen von dem Ansatz aus, dass die Anforderungen zu Beginn des Projektes nicht vollständig bekannt sind, sondern sich während der Entwicklung ergeben oder ändern und Entwickler somit flexibler sind.[2] Zu den schwergewichtigen Prozessmodellen können unter anderem das V-Modell, das Phasenmodell und das Spiralmodell gezählt werden.

Die sogenannten *agilen Prozessmodelle*, zu denen auch das *XP* gehört, werden der Kategorie leichtgewichtig zugeordnet. Sie berücksichtigen alle das *Manifesto for Agile Software Development* – ein Werk aus dem Jahr 2001. Es beschreibt Grundsätze der agilen Softwareentwicklung. Zu den Autoren gehören unter anderem Kent Beck, der *XP* entwickelt hat, sowie viele weitere Verfechter der agilen Entwicklungsmethoden.[3]

Kommentar [RAT4]: [Major]-NominalStyle

Dieser Satz enthält 3 oder mehr (3) abstrakte Substantive die auf -heit, -keit oder -ung enden. (Software-Entwicklung, Festlegung, Erstellung)

Sehen Sie Beispiele zu dieser Regel in der Dokumentation.

Kommentar [RAT5]: [Minor]-SentenceWithSameWords

Der nachfolgende Satz beginnt mit dem selben Wort 'Die'.

Sehen Sie Beispiele zu dieser Regel in der Dokumentation.

Kommentar [RAT6]: [Major]-AdjectiveStyle

Dieser Satz enthält 5 oder mehr (7) Adjektive: schwergewichtigen, detaillierte, spätere, vorher, definierten, hohem, möglich

Sehen Sie Beispiele zu dieser Regel in der Dokumentation.

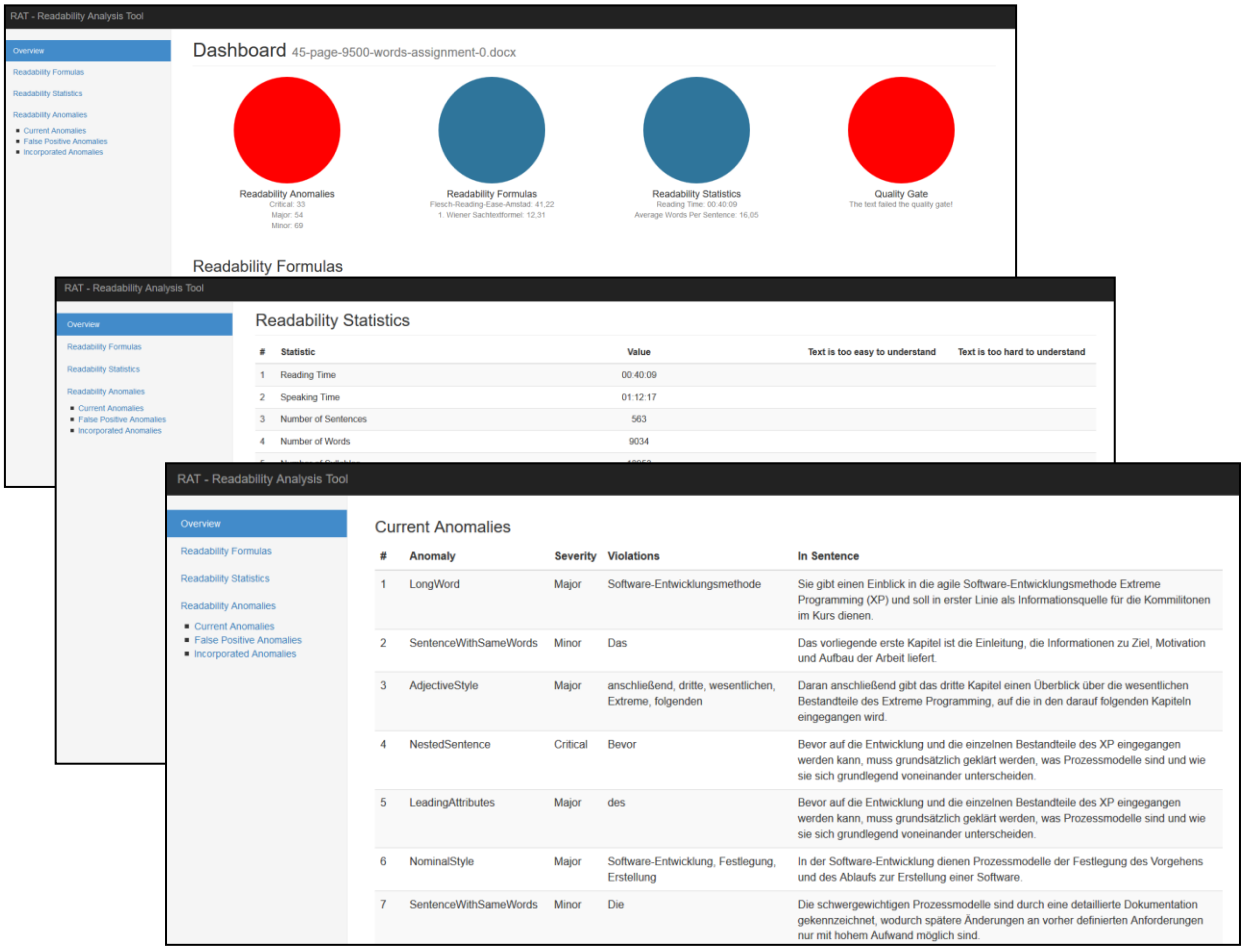
Kommentar [RAT7]: [Critical]-NestedSentence

Dieser Satz enthält 5 oder mehr (5) Konjunktionen oder Gliederungszeichen.

Sehen Sie Beispiele zu dieser Regel in der Dokumentation.

Report and Quality Gate

- 19 Quantitative Text Statics
  - Reading Time
  - Average Syllables
  - Most used Adjectives
  - Keywords in Text
- Readability Formulas
  - Flesch Reading Ease (Amstad)
  - Wiener Sachtextformel
- Classification of Anomalies
  - Declined Anomalies
  - Incorporated Anomalies
  - Current Anomalies
- Present the Text in the HTML Report with highlighted Readability Anomalies



Evaluation

Relevance

- During the application of RAT, the practitioners have incorporated 49% of the findings.
- Participants considered 64% of true-positive findings as relevant and would incorporate 59% immediately.
- Participants were not aware of 48% of the findings.

Precision

- The average precision of readability rules is 69% with high variation.
- 7 out of 17 rules had a precision greater than 70%, which is considered acceptable in static code analysis [Be10].

Corpus

- We detected 314,443 anomalies in 4,619 documents in the QAware corpus.
- On average, we found one anomaly in every 3.69 sentences or every 46.52 words.

Performance

- The analysis takes an average of 42 seconds for a text of 10,823 words.
- An initialized pipeline takes an average of 21 seconds for 10,823 words.

Conclusion and Prospects for Future Work

Conclusion

Automated readability anomaly detection provides a way to improve the readability of a text without time-consuming review cycles.

- Editors can focus on the content rather than stylistic errors
- Awareness for the importance of readability is created
- Common writing guidelines can be established

Limitations

- Precision
  - Grammatical errors in the text
  - Imprecisions in NLP libraries
  - Readability rules do not take the context into account
  - Text extraction
- We found no process model in place

Future Work

- Examine the recall of readability rules
- Impact of difficult-to-read texts
- Use RAT for more empirical studies
- Domain specific and visual perception rules
- Paraphrasing suggestions

Matthias Holdorf
B.Sc. Information Systems

Technische Universität München
Department of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel. +49 152 534 490 65
E-Mail matthias.holdorf@gmail.com

wwwmatthes.in.tum.de

Bibliography

[He04] Hevner, A. R. et al.: Design Science in Information System Research. In MIS Quarterly Vol. 28 No. 1, 2004;p. 75–105.

[Fl48] Flesch, R.: A new readability yardstick. In Journal of applied psychology 32.3, 1948;p. 221.

[Sc98] Schmidt-Wigger, A.: Grammar and style checking for German: Proceedings of CLAW, 1998.

[vHH08] vor der Brück, T.; Hartrumpf, S.; Helbig, H.: A readability checker with supervised learning using deep indicators. In Informatica, 2008, 32;p. 429–435.

[PRR10] Perin, F.; Renggli, L.; Ressia, J.: Natural Language Checking with Program Checking Tools, Bern, 2010.

[Fe16] Femmer, H. et al.: Rapid quality assurance with Requirements Smells. In Journal of Systems and Software, 2016.

[Mi56] Miller, G. A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. In Psychological review, 1956, 63;p. 81.

[Be10] Bessey, A. et al.: A few billion lines of code later: using static analysis to find bugs in the real world. In Communications of the ACM, 2010, 53;p. 66–75.

[Ke12] Kercher, J.: Verstehen und Verständlichkeit von Politikersprache: Verbale Bedeutungsvermittlung zwischen Politikern und Bürgern. Springer-Verlag, 2012.

Backup

Introduced Terminology

sebis

Readability Anomaly

The term readability anomaly refers to an indicator of difficult-to-read text passages that may negatively affect communication between stakeholders.

Readability Rule

A readability rule detects a readability anomaly.

IT-related Text

The terminology of IT-related text describes all text that comes into existence through the communication of stakeholders participating in the engineering lifecycle of an information system.

Research Questions

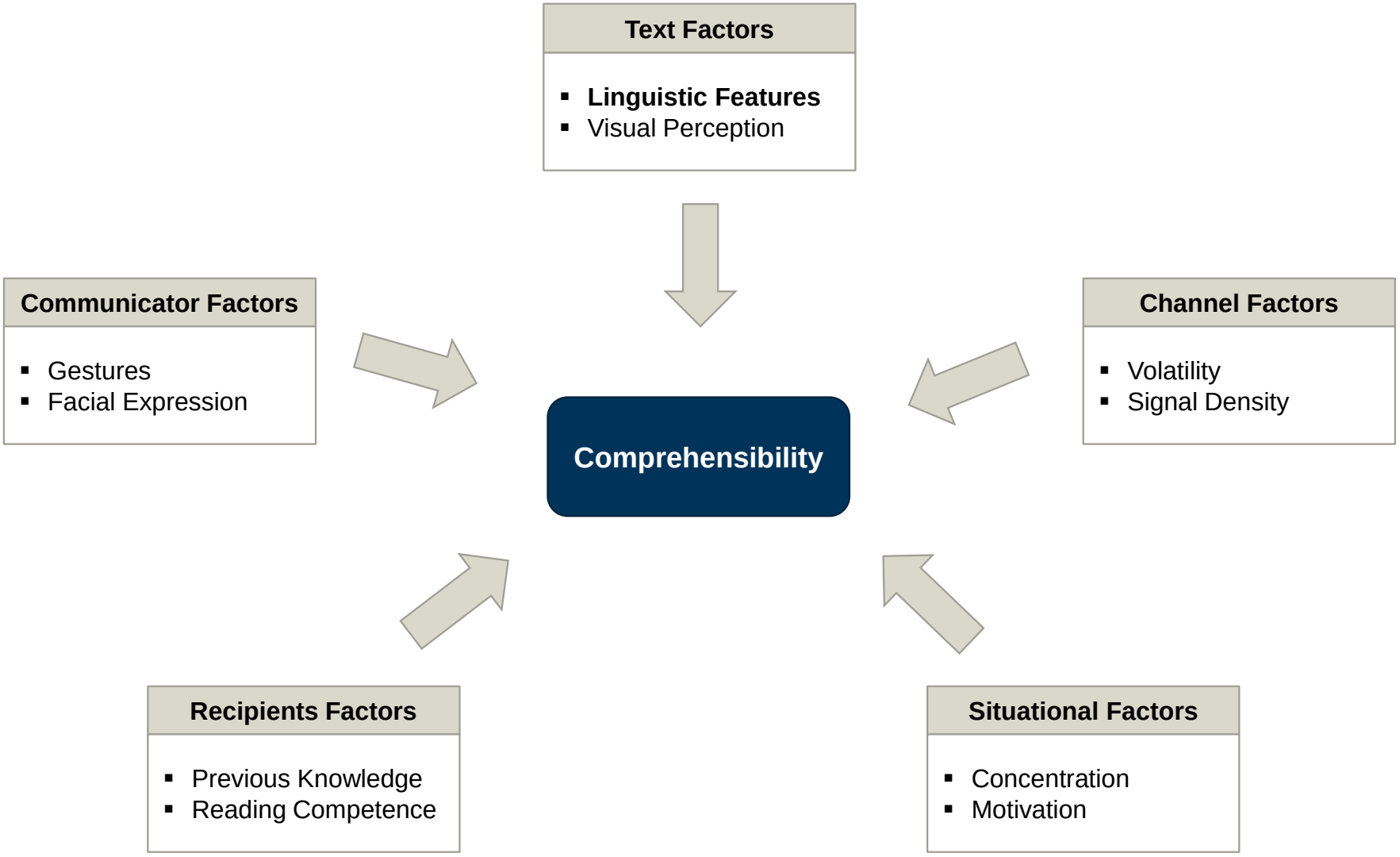
RQ1	What problems are caused by difficult-to-read texts in IT?
RQ2	How can a readability checker be integrated into the workflow of an IT company?
RQ3	How can we improve the readability of IT-related texts?
RQ4	What are functional and non-functional requirements of a readability checker for IT-related text?
RQ5	How does a prototypical implementation of a readability checker for IT-related text look?
RQ6	How accurate is the readability anomaly detection?
RQ7	How many readability anomalies are relevant?
RQ8	How many readability anomalies are present in the corpus of an IT company?

Error Classes determined by Interviews

Error Class	Linguistic Level	Fulfilled
Missing redundancy, ambiguity and vague references	Coherence	
Illustrations and missing descriptions of illustrations	Visual Perception	
Semantic errors that can be detected through lexical or syntax features, e.g. attachments in emails	Lexical / Syntax	
Implicit vocabulary and abbreviations not in the glossary	Lexical	
Vocabulary fits the target readership	Lexical / Semantic	
Using less modal verbs	POS	
Avoid the passive voice	Morphological	
Avoid long and nested sentences	Lexical / Syntax	
Avoid text passages that make a text longer without adding content	Lexical / Semantic	

Based on Interview Findings in QAware

Readability and Comprehension



Adapted Hohenheimer Modell by [Ke12 p. 136, Kl74]

Implemented Readability Rules

#	Rule	Threshold	Enabled
1	AdjectiveStyle	5	true
2	AmbiguousAdjectivesAndAdverbs		true
3	ConsecutiveFillers		true
4	ConsecutivePrepositions		true
5	DoubleNegative	2	true
6	Fillers		false
7	FillerSentence	3	true
8	IndirectSpeech		false
9	LeadingAttributes	4	true
10	LongSentence	35	true
11	LongWord	8	true
12	ModalVerb		false
13	ModalVerbSentence	2	true
14	NestedSentence	6	true
15	NestedSentenceConjunction	3	false
16	NestedSentenceDelimiter	3	false
17	NominalStyle	3	true
18	PassiveVoice		true
19	SentencesStartWithSameWord	2	true
20	SubjectiveLanguage		true
21	Superlative		true
22	UnnecessarySyllables		true

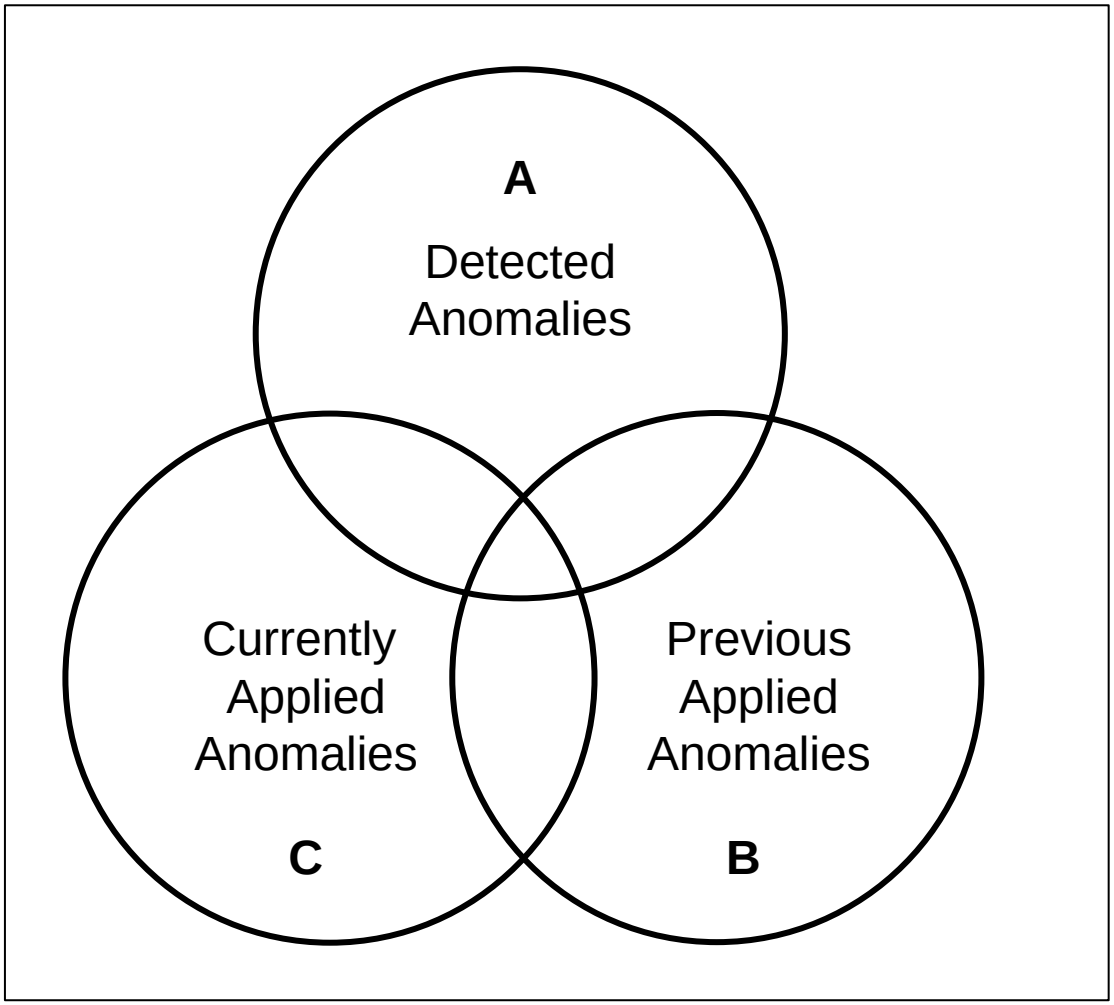
Empirical Studies

sebis

Interview Guideline and Survey Questions.

#	Interviews	Survey
1	Which problems do difficult-to-read texts cause?	What text processing programs are in use?
2	What text processing programs are in use?	What categories of texts do you write or edit?
3	What categories of texts exist?	How long should an analysis take?
4	How do employees write texts?	Would the artifact be used if the results were stored in a separate document?
5	How do employees edit texts?	How can a readability checker be integrated into the workflow?
6	What errors regarding readability occur in IT-related texts?	How much time do you spend weekly on writing texts?
7	What are the requirements of a readability checker?	How much time do you spend weekly correcting texts?
8	Will the artifact be used if the results are stored in a separate document?	What problems do difficult-to-read texts cause?
9	How long should an analysis take?	
10	How should findings be displayed?	
11	How can software support be integrated into employees' workflow?	

Classifying Readability Anomalies



Redundant Anomalies:

```
R :=        A ∩ C
```

New False Positive Anomalies:

```
NFP :=      (A \ A ∩ C) ∩ (B ∪ PFP)
```

Incorporated Anomalies:

```
IC :=       B \ NFP \ R ∪
            LS(A, B, 30)

I :=        IC \ LS(IC, R, 30)
```

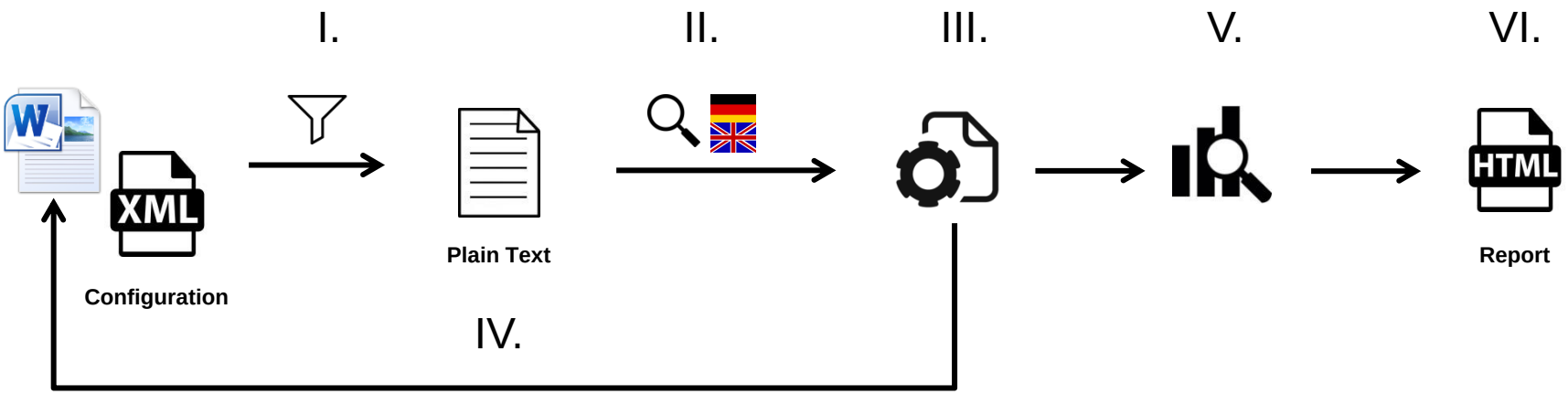
New Previous Applied Anomalies:

```
A :=        A \ ((A ∩ C) ∪ (A ∩ NFP)
            ∪ LS(A, B,30))

NPA :=      R ∪ A
```

PFP	Set of Previous False Positives
NPA	Set of Previous Applied Anomalies for the next Analysis
LS(x, y, t)	Intersection of x and y with LevenshteinDistance smaller than t

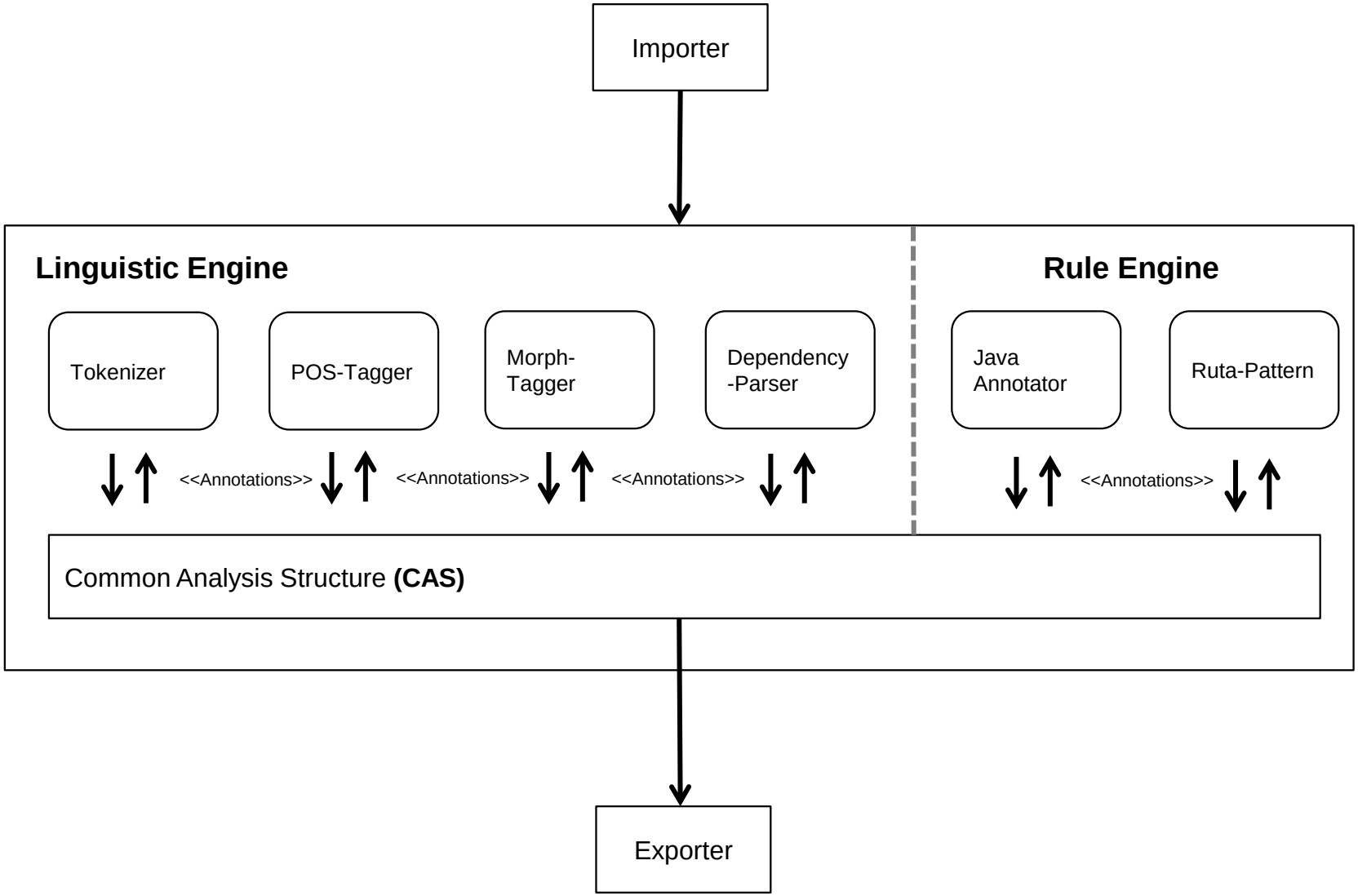
Steps of an Analysis



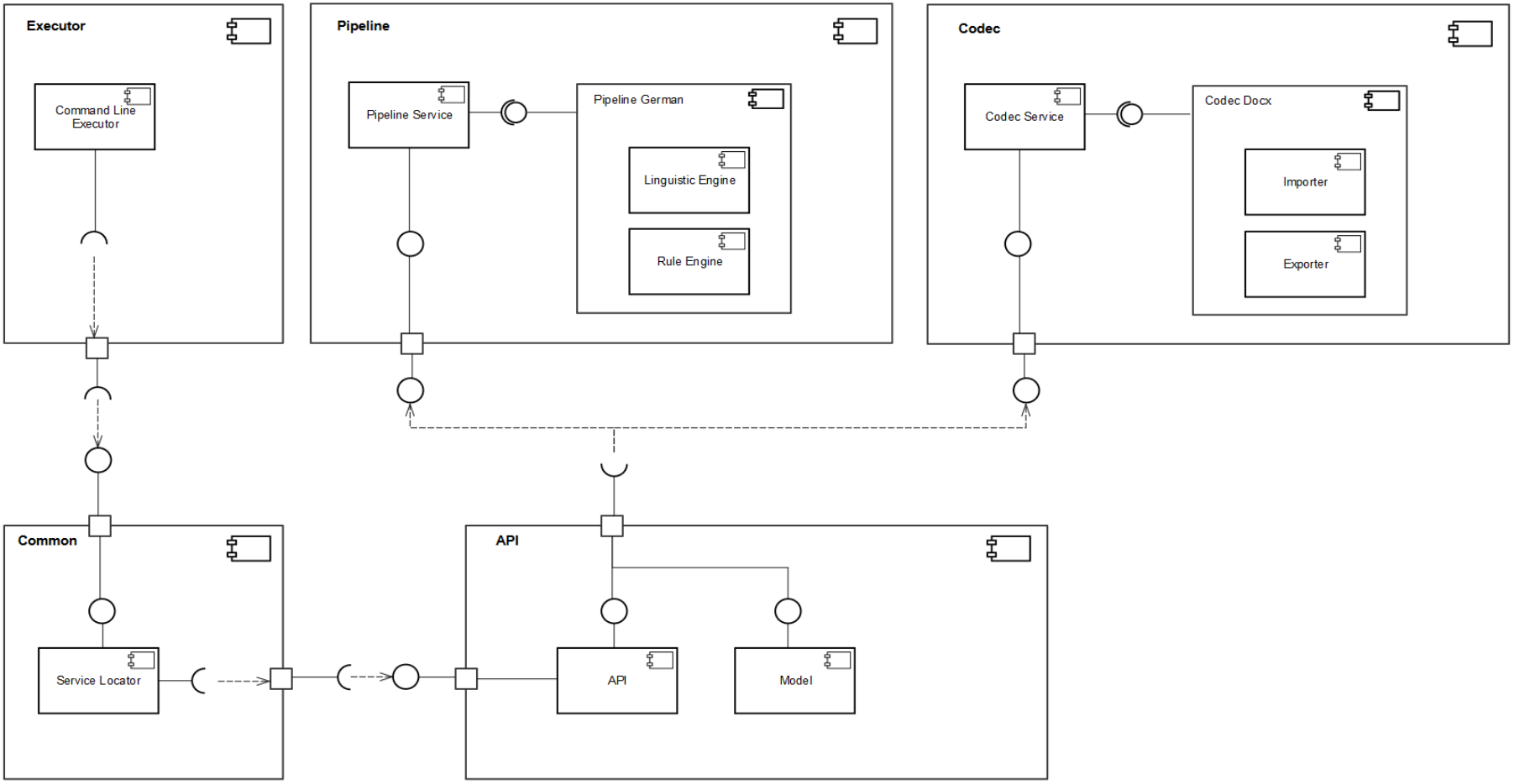
Legend

I. File Type Detection, Text Extraction, Filtering and Abstraction
II. Language Detection
III. Linguistic Analysis and Readability Anomaly Detection
IV. Applying Comments and Export of Document
V. Calculation of Quantitative Measurements
VI. Export of Report

Conceptual Overview



Simplified Component Architecture



Integration into the IS Architecture of QAware