A Comparative Evaluation of Requirement Template Systems

Katharina Großer
Institute for Software Technology (IST)
University of Koblenz
Koblenz, Germany
grosser@uni-koblenz.de
0000-0003-4532-0270

Marina Rukavitsyna
Institute for Software Technology (IST)
University of Koblenz
Koblenz, Germany
mrukavitsyna@uni-koblenz.de

Jan Jürjens
Institute for Software Technology (IST)
University of Koblenz
Koblenz, Germany
juerjens@uni-koblenz.de
Fraunhofer ISST
Dortmund, Germany
0000-0002-8938-0470

***Abstract*—*Context:* Multiple semi-formal syntax templates for natural language requirements foster to reduce ambiguity while preserving readability. Yet, existing studies on their effectiveness do not allow to systematically investigate quality benefits and compare different notations. *Objectives:* We strive for a comparative benchmark and evaluation of template systems to support practitioners in selecting template systems and enable researchers to work on pinpoint improvements and domain-specific adaptions. *Methods:* We conduct a comparative experiment with a control group of free-text requirements and treatment groups of their variants following different templates. We compare effects on metrics systematically derived from quality guidelines. *Results:* We present a benchmark consisting of a systematically derived metric suite over seven relevant quality categories and a dataset of 1764 requirements, comprising 249 free-text forms from five projects and variants in five template systems. We evaluate effects in comparison to free text. Except for one template system, all have solely positive effects in all categories. *Conclusions:* The proposed benchmark enables the identification of the relative strengths and weaknesses of different template systems. Results show that templates can generally improve quality compared to free text. Although MASTeR leads the field, there is no conclusive favourite choice, as overall effect sizes are relatively similar.**

***Index Terms*—Requirement Templates, Readability, Quality Metrics, Guideline Rules, Natural Language Requirements**

I. Introduction

To specify requirements, natural language is still frequently used [1]. Partially, it is preferred because formal notations can reinforce a "language barrier" between developers and stakeholders that makes it hard to evaluate if the noted requirement is equivalent to the originally intended need [2], [3]. In particular, non-technical stakeholders, e.g., legal advisers, are affected. Further, formal notations are associated with training overhead, which is rarely accepted [4]. Yet, natural language is often ambiguous and hard to process automatically. Linguistic mistakes and misunderstandings are frequent reasons for inadequate requirements [1], [5].

To phrase requirements more precisely, controlled syntaxes or syntax templates can be used, e.g., EARS [4], *MASTeR* [6], or the simple syntax in ISO/IEC/IEEE 29148 [7]. With their unified structure, templates can standardize the requirements and approximate their form to a formal notation without loss of readability. Such *semi-formal* [8], [9] approaches afford less training [4], [2], improve quality [4], and template structures can be exploited for mappings and transformations. E.g., to relate requirements to domain knowledge [10], [11] or generate models such as data-flow diagrams from them [12].

To achieve these goals, a template system matching the intended purpose must be selected and applied. In terms of effectiveness and quality benefits, most template systems are evaluated compared to free text requirements. Yet, different evaluation objectives and methods of existing studies do not allow for a systematic comparison of performances of different template systems. There exists no common benchmark and formality is rarely considered. To date, the authors are not aware of any study comparing multiple template systems.

In this paper, the following research question is investigated:

How do different template systems influence the quality of requirements?

In practice, these effects depend on various context factors, like domain, development phase, target audience, or capabilities and preferences of the requirement authors. Similarly, the notion of and expected level of quality depends on this context, too. Yet, to enable a comparison between different template systems, there is the need to establish some common ground.

In the following, we present a *comparative* evaluation of five popular template systems towards ISO/IEC/IEEE 29148's [7] quality criteria and 39 guideline rules based on syntactically rephrasing a dataset of 249 requirements from five projects.

Pursuant to experiment reporting guidelines [13], Section II gives background on requirement templates and quality criteria. Section III summarizes related work. Experiments are reported in Section IV, the outcome is discussed in Section V, and Section VI concludes this paper.

II. Background

A. Template Systems

"A generic, syntactical requirements template is the blueprint that determines the syntactical structure of a single requirement." [20] Generally, they consist of fixed text and variable parts—"holes" to be filled. Variable parts are often denoted within < > and optional parts with [ ]. By substituting

K. Großer, M. Rukavitsyna, and J. Jürjens, "A Comparative Evaluation of Requirement Template Systems," in *31th IEEE International Requirements Engineering Conference (RE'23)*, 2023, pp. 41–52. DOI 10.1109/RE57278.2023.00014

TABLE I

EXAMPLES OF TEMPLATE SYSTEMS: RESPECTIVE PHRASING VARIANTS FOR THE SAME EXAMPLE REQUIREMENT FROM THE SPECIFICATION OF ESA'S VIRTUAL EAGLEEYE SATELLITE [14, ATB-SR-EO-1050] WITH THE RESPECTIVE USED TEMPLATE STRUCTURE BELOW.

Free Text	The AOCS-GNC shall control the SC attitude with the following performances (during Imaging mode): AME (Absolute Measurement Error) in the range of 100 μrad (3s).
	-
Boilerplates (DODT) [10], [15]	While *in Imaging Mode*, *the AOCS-GNC* shall have *Absolute Measurement Error (AME)* of at most *100 μrad (3s).*
	While <state>**,** <system> **shall have** <parameter> **of at most** <quantity> <unit>.
EARS [4], [16]	While *in Imaging Mode* the *AOCS-GNC* shall *maintain the Absolute Measurement Error (AME) in the range of 100 μrad (3s).*
	While <in a specific state> **the** <system name> **shall** <system response>.
Adv-EARS [17]	While *in Imaging Mode* the *AOCS-GNC* shall *maintain the Absolute Measurement Error (AME) in the range of 100 μrad (3s)* for *SC attitude control.*
	While <in a specific state> **the** <entity> **shall** <functionality> **the** <entity> **for** <functionality>.
MASTeR [6], [18]	As long as *EagleEye* is in *Imaging mode*, *the Absolute Measurement Error (AME)* of *the AOCS-GNC shall* be $\leq$ *100 μrad.*
	As long as <system> **is in [the state]** <state>, <characteristic> <subject matter> <liability> **be** <qualifying expression> <value>.
SPIDER [19], [2]	Between *EagleEye enters Imaging mode* and *EagleEye exits Imaging mode*, it is always the case that *AOCS-GNC Absolute Measurement Error (AME)* $\leq$ *100 μrad (3s)* holds.
	Between <Q> **and** <R>**, it is always the case that** <P> **holds.**

the variable parts, a requirement is instantiated. Keyword-driven languages, like Planguage [21], differ from this scheme, as they do not form a single sentence. The same applies to user story templates. Such notations are not targeted in this study.

Template approaches usually do not consist of just one template, but constitute a whole *template system* with related templates for different requirement types, as self standing sentences or sub-clauses that can be combined.

The five template systems analysed in this work—EARS [4], MASTeR [6], Adv-EARS [17], *Boilerplates* [22], [15], and SPIDER [2]—are selected based on their prevalent use in either industry or research. E.g., publications on EARS [4] are widely cited [3] (∼500[1]) while MASTeR [6] is maintained by SOPHIST[2] and assumed to be applied by a majority of their customers. Many organisation-specific templates are adapted variants of those. All selected template systems were encountered in practice by the authors during projects with research and industry partners. Only general purpose template systems are selected, excluding domain or problem specific ones, e.g., [23], [11]. The Mazo & Jaramillo template [24] and *FRETISH* [25] are published after the selection in early 2020. Table I lists all five with an example.

Boilerplates [22, pp. 81ff] are a set of templates for different types of requirements. For example, *stakeholder* requirements—*The <stakeholder type> shall be able to <capability>.*—and *system* requirements—*The <system> shall <function> <object>*. These basic forms are varied to express various constraints. All templates specify a full sentence, although the authors suggest that sub-clauses could be recombined in different ways. The *DODT* framework [10], [15] presents a consolidated set of all main- and sub-clauses.

EARS [4], [26], [16] works similar to *boilerplates*. The basic main clause form is *The <system name> shall <system response>*. Based on this, five requirement types are distinguished by combination of the basic form with prefix conditions. Compared to boilerplates, EARS templates are more standardized and guide the elicitation through defined types. Yet, the variable parts are not as fine grained, e.g., summarizing the *<system response>* in only one element. Solely the *Optional Feature* is not expressible by DODT [15].

Adv-EARS (Advanced-EARS) [17] is an extended version of EARS. It aims to be more generic and able to handle more different types of requirements. Therefore, it substitutes some element names and an additional *Hybrid* type combines event driven and conditional requirements. This is also incorporated to EARS under the term of *Complex* requirements [16], [3].

For *MASTeR* [6], variability within one template is explicitly modelled as different paths through a combined representation, e.g., for different types of system activity—*user interaction*, *autonomous*, or *interface*. Further, the templates are more fine grained. E.g., the described functionality is expressed through a combination of *<process verb>* and *<object>*. Optionally, further details can be added to both. It is the only template system with an explicit choice of modal verbs. The optional *[condition]* is described in a separate sub-template [18], with three different types, similar to EARS's prefixes. It has three additional main templates for non-functional requirements.

SPIDER [19], [2] is a template system based on an extension of qualitative specification patterns for different logic representations, such as *Linear Temporal Logic* (LTL), with real-time specification patterns for embedded-systems needs. Furthermore, it complements the pattern definitions for different logic specification languages by a structured English grammar consisting of 26 production rules. Each sentence serves as a handle that accompanies a scoped formula of a qualitative or real-time specification pattern [2].

B. Requirements Quality

Very few primary studies address evidence-based definitions and evaluations of quality attributes for requirements and there is no consistent use in the literature [27], [28],

[1]aggregated citations on https://scholar.google.com, visited 2022/11/21

[2]https://www.sophist.de/en/, visited 2022/11/21

where ambiguity, completeness, consistency, and correctness appear to be the most intensively researched ones [27]. As we focus on single statement templates, we do not address quality attributes that are only applicable to a whole set of requirements, like semantic consistency among requirements. Due to the lack of empirically founded definitions, we base on the widely used quality attributes listed in industry standards and guidelines. ISO/IEC/IEEE 29148 [7] lists nine quality attributes for individual requirements:

Necessary. If removed, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process.

Feasible. Technically achievable and fits within system constraints (e.g., cost, schedule, …) with acceptable risk.

Appropriate. The amount of detail and level of abstraction is appropriate to the level of the entity to which it refers.

Unambiguous. Stated in a way that it can be interpreted in only one way, phrased simply and is easy to understand.

Complete. Sufficiently describes the characteristics to meet the need of the stakeholder and is measurable.

Singular. The statement includes only one requirement with no use of conjunctions respectively only one main verb. Yet, it can have multiple conditions.

Verifiable. Has the means to prove that the system satisfies the specified requirement. Should be measurable.

Correct. The statement is an accurate representation of the need from which it was transformed.

Conforming. If applicable, conforms to the approved standard template and style for writing requirements.

The first two—necessary and feasible—are not influenced by templates and are not assessable outside of the project context, thus, we focus on the other seven criteria.

For each quality attribute, a wide variety of more detailed sub-types can be identified [27], as, e.g., described in guidelines like the INCOSE Guide for Writing Requirements [29].

III. Related Work

Phrasing guidelines based on syntactic rules, such as "use active voice", are a major part of common foundations of template systems [30]. Some rules from such guidelines [29], [31] are compared in different studies [32], [33], [34] with requirements phrased without any guideline. Results indicate that pronouns and negations, which are discouraged by the guideline, are widely used and can lead to shorter, easier-to-understand sentences. Thus, rules seem to be too restrictive and experienced authors follow useful ones intuitively. Yet, their potential prior training/exposure to guidelines is not discussed. In a subsequent survey [35], experts and laymen have to choose the easiest to understand among different phrasings of requirements. Especially experts do not always favour requirements following the guidelines. For the recommended use of quantifiers, like "at least" or "all", another study indicates that negative quantifiers reduce readability and six out of nine examined quantifiers hinder correct understanding [36].

All template systems use different conditional statements to indicate their respective types of conditions. Conditionals, in particular, "if" and "when", can be a source of ambiguity, as they are interpreted in different ways by practitioners [37].

EARS [4] is originally evaluated by rephrasing the EASA Certification Specifications for Engines [38]. Results show that it reduces ambiguity and the length of the requirement sentences, while the total number of requirements increases. Later case studies in other contexts and domains support these results [3]. They further state some positive feedback on learnability, yet, this is not evaluated in detail. Exploratory research in knowledge engineering on learning a controlled natural language [39] shows that experts familiar with knowledge engineering learned the textual notation very quickly. It can be assumed that previous experience with requirements engineering also promotes the easy adoption of semi-formal notations. The question "whether EARS is *just formal enough* for automated analyses (and syntheses)" is partially addressed by early stage work with an adapted version of EARS [12] and examples of controllers with corresponding data-flow diagrams. The initial results are promising, but have limitations in expressiveness for complex states.

For SPIDER, "[f]eedback from industry has indicated that a structured English representation is less intimidating than the temporal logic notation" [2]. For formality and expressiveness, equivalence of the grammar to the pattern catalogue in different formal methods , like LTL [40], is demonstrated.

Controlled experiments with students [41], comparing Boilerplates [15] to free text, on the ability to spot errors within requirements do not deliver clear results due to a small sample size and outliers in the dataset. Despite the assumption that Boilerplates can prevent authors from writing too complex requirements, the experiment on writing requirements shows low quality levels in the results. Most students fail to preserve the meaning of the original task description. It is assumed that the results are negatively impacted by the low experience level of the participants and some bias through the used examples.

Starting from expressions not covered by the older 2007 version of the MASTeR templates [42], an extension is developed [24], combining MASTeR templates with additional concepts from other template systems, e.g., EARS and Adv-EARS. This union is more complete and robust, as tested in two industrial case studies. Yet, due to the constructive iterative approach there is no comparison between the original baseline template systems.

IV. Comparative Evaluation of Template Systems

In the following, we report how we constructed our benchmark to compare different template systems as well as results from the experiment with the five selected template systems.

A. Methodology

To construct our metric suite, we followed IEEE 1061 [43] and systematically extracted 39 rules applicable to *individual* requirement statements together with the quality attributes they are related to from the union of six relevant domain standards and guidelines [7], [44], [29], [45], [46], [47], as shown in Table II. Rules are included if they are individually mentioned in

Fig. 1. Attribution of Rules & Metrics to Quality Factors for Template Systems

at least one of the six sources. Figure 1 shows the non-disjoint attribution of these rules to the quality attributes and, where applicable, assigned quantifiable metrics. Where no simple counting metric is found to be directly applicable, the rules are interpreted as being boolean with respect to their fulfilment per requirement and percentages over the examined requirement sets. In addition to the guideline rules, we included *readability* scores [48] that directly measure comprehensibility, which is among the most relevant quality attributes in practice [49], and investigated *formality* [8], [9] to meet concerns of model-based development. Here, we use the *F-Score* [50], to measure *deep formality*—the "attention to form for the sake of unequivocal understanding of the precise meaning of the expression" [51], which is closer to the meaning of formality in formal methods of computer science than *surface formality*, such as formal speech. Yet, we attribute the F-Score to *completeness*, as this "means that a maximum of meaning is carried by the explicit, objective form of the expression [. . .] rather than by [. . .] context" [50]. The detailed definitions of all metrics in our metric suite are contained in the complementary material [52].

TABLE II
REQUIREMENT PHRASING GUIDELINES AND THEIR RULES

Phrasing Rules	Phrasing Guidelines for Requirements	[7] ISO29148	[44] SOPHIST	[29] INCOSE	[45] ECSS E10	[46] ECSS DR	[47] NASA
1	use only one sentence		X	X	X		
2	avoid unnecessary words		X			X	X
3	*use only one process verb*		X	X			X
4	avoid extensive punctuation		X	X			
5	use defined modal verb for liability	X		X	X	X	X
6	*use simple structured sentence*	X	X	X	X	X	X
7	*use appropriate abstraction level*	X		X	X	X	X
8	use active voice	X	X	X		X	X
9	*use precise verb*		X				
10	avoid nominalization		X				
11	*avoid light verb construction*		X				
12	use full verb		X				
13	avoid comparison	X	X				
14	*use clear comparison*		X				
15	use definite articles		X	X			
16	use defined units			X		X	
17	avoid vague terms	X	X	X	X	X	X
18	avoid escape clauses	X		X			
19	avoid open ended clauses	X		X			
20	avoid superfluous infinitives			X			
21	use correct grammar + spelling			X			X
22	avoid negations	X		X	X		X
23	avoid /			X			
24	avoid combinators		X	X	X		
25	*separate rationale from sentence*	X	X	X	X		X
26	avoid parentheses			X			
27	*avoid group-nouns*		X	X			
28	avoid pronouns	X		X			X
29	*use context free phrasing*	X	X	X	X	X	X
30	avoid absolutes	X		X			
31	*use explicit conditions*	X	X	X			
32	*use clear condition combinations*		X	X			
33	*use solution free phrasing*	X	X	X	X		X
34	use clear quantifiers		X	X			X
35	use value tolerances			X	X	X	X
36	*express one atomic need*	X	X	X	X	X	X
37	*use clear preconditions*		X	X		X	X
38	*use clear business logic*	X	X			X	X
39	*use clear subject*		X			X	X

Metrics are evaluated per individual requirement and aggregated per requirement set that forms the respective control or treatment group. The majority of metrics is binary true or false on the level of individual requirements. Here, aggregated %-values correspond to the *risk* of having this defect/smell in this group. The raw effect of treatment with a template system is measured by the *risk difference* = $R_{treatment} - R_{control}$ [53] and the strength of this effect can be judged by the *relative risk* (RR) [53]—the ratio of the risk in the exposed group to the risk in the unexposed group. We further calculate corresponding 95% confidence intervals to test statistical significance. We provide more detailed explanations, e.g., of our treatment of zero values, in the supplemental material [52]. For those metrics that return decimals, effect size is based on *means*, where the raw effect is the mean difference between the treatment and the control groups $\mu_{treatment} - \mu_{control}$. To judge the strength of the effect, we calculate Cohen's *d* [54]. Significance is judged by an unpaired two tailed t-test [55] with a 95% confidence interval.

To enable a comparison of effect sizes of the two types among the different metrics, we matched value ranges for the relative risk with the six level magnitude "rules of thumb" for Cohen's *d* values [56]. Although Cohen emphasized that these values should be handled flexible [54], they have become a de-facto standard in research [56]. The categorization allows us to compare different effect size measures on a scale of more coarse grained magnitudes, which abstracts from small insignificant differences in absolute values that might be misleading. Table III lists how we matched relative risk values to the already established *d*-values from "rules of thumb" [56] to mirror their non-linear increasing interval sizes.

TABLE III
EFFECT SIZE MAGNITUDES FOR COHEN'S *d* AND RELATIVE RISK

Magnitude Category		**Cohen's *d*** [56] (|*d*|)	**Relative Risk** (|1-RR|)
0 - No Effect	(-)	0.0	0.0
1 - Very Small	(XS)	≥ 0.01	≥ 0.005
2 - Small	(S)	≥ 0.2	≥ 0.1
3 - Medium	(M)	≥ 0.5	≥ 0.25
4 - Large	(L)	≥ 0.8	≥ 0.4
5 - Very Large	(XL)	≥ 1.2	≥ 0.6
6 - Huge	(XXL)	≥ 2.0	≥ 1.0

We aggregate effects over several metrics via mean values of the magnitude categories' ordinal numbers $\in [0..6]$. E.g., effect sizes of magnitudes *S*, *M*, *L*, & *L* for four metrics would have a summary effect size of $\frac{2+3+4+4\ (=13)}{4} = 3.25$, thus, *medium*. Insignificant results are treated as zero. This is less precise than a mean over the actual RR or d values However, it allows to combine RR and Cohen's d effect sizes, what is otherwise not possible as these have different value ranges. We calculate this separately for positive and negative effects.

In the following, we provide effect size values as 3-tuples in the form *(effect size, magnitude* $\in [XS..XXL]$*, raw effect)*, e.g., (0.62, M, -15%) for an RR or (0.29, S, -3) for a *d*-value.

We evaluate seven hypotheses, which we derived from the template systems' goals (H_2–H_4) and findings in related work (H_1 [4], [6] & H_5 [36]–H_7 [32], [33], [34], see above):

H_1 *Usage of templates leads to more requirements.*
H_2 *The quality of template requirements is improved.*
H_3 *Different template systems have different effect.*
H_4 *Different template systems match to different guidelines.*
H_5 *Quantifiers negatively correlate with readability.*
H_6 *Pronouns correlate with shorter requirements.*
H_7 *Pronouns do not negatively correlate with readability.*

While H_1–H_4 aim at a quantitative and qualitative evaluation of the template systems themselves, H_5–H_7 aim at findings in related work that question parts of their foundations.

We accept H_1 if the total number of requirements increases in the respective treatment groups.

We accept H_2 for each individual metric and template system if observed effect sizes for positive effects are $\geq$ XS

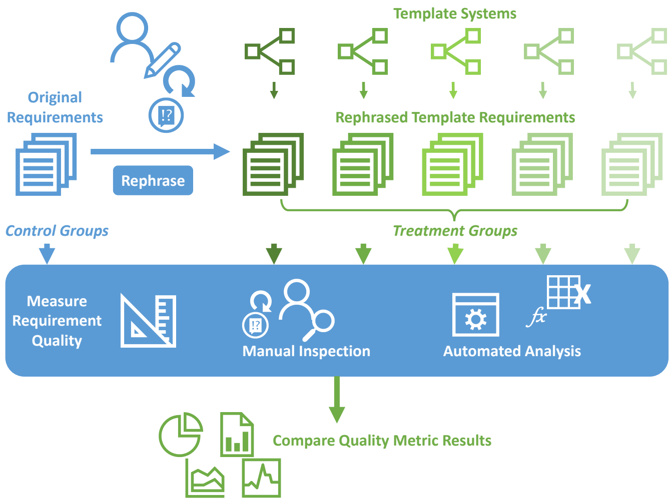

Fig. 2. Experiment Set-Up for Template Quality Comparison

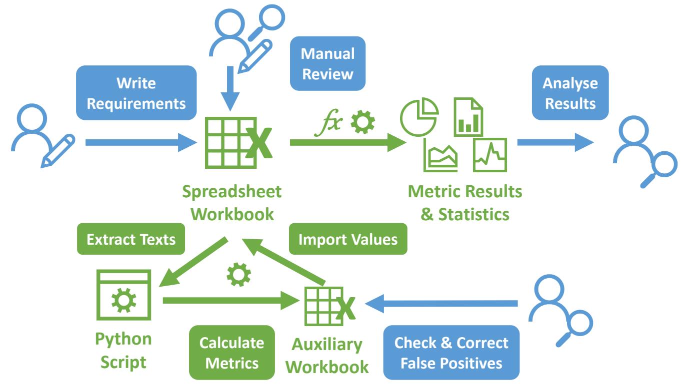

Fig. 3. Workflow for Quality Metric Calculation

and statistically significant with $p \leq 0.05$. Otherwise we accept the null-hypothesis $H_{0\text{-}2}$ that there is no effect on this metric by this template system or accept the alternative hypothesis $H_{A\text{-}2}$ that the template system has a negative impact on this quality respectively. Equivalently, for a metric group attributed to a specific quality or guideline, we accept H_2 or $H_{A\text{-}2}$ for a template system respectively, when the mean over all positive or negative significant effect size categories is $\geq$ XS.

We accept H_3 and H_4 if there are differences in effect size magnitudes between the examined template systems for the respective metric or metric group.

For H_5–H_7, we investigate Spearman rank correlation [57], [58] between metric results for the different document groups and accept (or reject for H_7) the hypothesis if the respective correlation is significant with $p \leq 0.05$ [59].

B. Experimental Setup

Figure 2 illustrates the general experimental setup, which consists of the dataset creation and the metric calculation.

Dataset Creation. For the free-text control groups, we choose five real-world documents with in total 249 requirements from different abstraction levels: the Certification Specifications for Engines (CSE) [38] with 25 requirements, which is already used in the original EARS evaluation [4], a similar standard from the space sector—E-ST-60-30C [60] (33 requirements), the high-level system requirements of the FLEX[3] space segment (18 requirements), and two detailed specifications of projects from practical software engineering and programming courses—a time sheet system (TSS) with 63 and an electronic voting system (EVS) with 110 requirements.

To complete the dataset with the template variants, the requirements in free text form are rephrased following the guidelines of the five examined template systems. Each of these 25 rephrasings represents a treatment group. For quality assurance, an iterative approach was applied. Initially, the first three documents—the standards and system requirements from the aerospace domain—were transformed to EARS and MASTeR by a bachelor-level student. The results were reviewed and revised by the second author and complemented with the rephrasings of the two larger documents from the lecture example projects and the remaining three template systems for all five documents. Finally, all rephrasings were reviewed and revised by the first author. Cases of doubt were settled in discussions between the first and the second author.

One difficulty is to only improve the requirements up to the extent really encouraged by the template description and syntax structure. We tried to follow as strictly as possible the syntax descriptions from the original publications and transform as much as possible from the original requirement in a "naïve" way, to be able to compare the different syntax structures provided by the templates. Nevertheless, we are well aware that all these approaches usually come with additional training that encourages a mind set for further improvements, which are not bound by the syntax structure. We corrected grammar and spelling mistakes while rephrasing and spell-checked the new template requirements.

In addition to the five control groups per original document, we reshuffle the free-text requirements to five randomized control groups with respective 25 randomized treatment groups and one big pooled control group over all free-text requirements with five respective treatment groups, to compensate for effects specific to the original documents.

Metric Calculation. The quality metrics are evaluated as illustrated in Figure 3. For all individual requirements, 23 rule-metrics, four readability scores, and eight auxiliary metrics, such as number of characters or syllables, are evaluated automatically by Excel formula or Python script [52]. Values are manually cross-checked and corrected for false positives, which in particular occur for missing units and value tolerances. Additional 16 metrics are evaluated through manual review (marked *italic* in Table II). These values were first assigned by the second author and then reviewed and revised by the first author, equivalently to the rephrasings.

All metrics are automatically aggregated to sums, means, or percentages for the groups to compare. In addition, the F-Score

[3] https://earth.esa.int/eogateway/missions/flex, visited on 2021/11/23

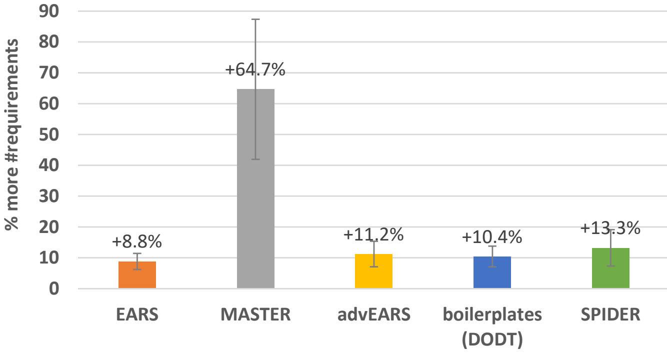

Fig. 4. Percent Increase in Requirements per Template System. Error Bars indicate Standard Deviation σ over all Random Groups.

and 12 readability scores are calculated per group respectively, as individual sentence samples are too small. While the F-Score is covered by a Python script, readability scores are calculated by Excel & an external tool [61].

C. Results

In total, the resulting dataset contains 1764 requirements, which is more than $6 * 249 = 1494$, as some original requirements need to be split into several new requirements when rephrased. Over all original requirements, the rephrasing leads for all template systems to an increase of the total number of requirements between 8.8% (EARS) and 64.7% (MASTeR). Thus, H_1 holds. As can be seen from Figure 4, the large increase through MASTeR sticks out, while all other template systems induce more requirements in a very similar magnitude.

In the following, due to space restrictions, detailed results are only presented for unambiguity, which we choose as a representative excerpt of individual results, because it comprises the majority of examined metrics (c.f. Figure 1) and is among the most relevant quality attributes in practice [49]. Table IV summarizes results in effect sizes for all 37 attributed metrics. Detailed results can be found in the replication package [52].

From Table IV one can see that all template systems influence most of the unambiguity related metrics in a positive way (H_2 holds). The respective best value is marked **bold**. For cells marked with "-", effects are not significant ($p > 0.05$).

For avoidance of nominalizations (R10) as well as the usage of defined units (R16) and quantifiers (R34), we have to reject H_2 over the full dataset. However, examining the effects on the original document groups, EARS, DODT, and SPIDER have even large effects (0.42-0.43, L, -14%) on the usage of clear quantifiers for the Certification Specifications for Engines. Similar, for usage of full verbs (R12), there are small and very small effects of MASTeR on both standards.

All template systems entirely enforce requirements to be phrased as single (R1) and structured (R6) sentences equally. Here, H_3 has to be rejected. Further, all template systems lead to shorter requirements (R2) and increase the amount of requirements with only one process verb (R3), clear condition combinations (R32), and phrased on the appropriate abstraction level (R7), while vague terms (R17) and open ended clauses (R19) are decreased. Yet, H_3 holds, as we see substantial differences between the template systems for all metrics where effects are significant.

For the use of full verbs, solely DODT and SPIDER show some effect, while all but SPIDER significantly reduce the risk of passive voice (R8). Solely MASTeR has a significant, yet, small effect on the amount of punctuation per 1000 words (R4). From several corpus studies, 209/1000 words is estimated as the average for free natural language texts [62]. Considering the effect on the percentage of requirements that stays below this value, solely SPIDER has no significant effect, while all others have a medium, MASTeR even a very large, effect. Yet, the effect appears to reside in-equally strong in specific requirements, as it is not observable for all random groups.

To not overate readability, we selected only one representative score in Table IV: the Flesch-Kincaid grade level [63]. Yet, the impact on readability is neglectable, despite the worsening by SPIDER, which can be observed for 10/14 scores over the pooled as well as random treatment groups. Curiously, no significant effect can be observed over the document groups.

Overall, MASTeR has a positive effect on the most metrics (26/37) and the strongest effect (medium). All other template systems have a small aggregated effect. In five cases, MASTeR is the only template system that shows a measurable positive effect. E.g., solely MASTeR reduces significantly and strongly the risk of an unclear subject (R39). Besides MASTeR, only SPIDER has such distinctive features, but just two—the use of clear comparison (R14) and context free phrasing (R29).

However, for SPIDER, we see four cases (marked **red**) where the respective metric is not improved, but impaired, namely usage of modal verb (R5), avoidance of pronouns (R28), avoidance of absolutes (R30), and the Flesch-Kincaid grade level readability. Thus, here we reject H_2 and accept $H_{A\text{-}2}$ that the quality is decreased by the template system. In addition, the F-Score [50], attributed to completeness, is decreased (3.9, XXL, -1.59), too, so in total five metrics are negatively effected by SPIDER. All other template systems enforce the use of a modal verb (R5). As SPIDER templates do not contain modal verbs, this rule is violated by 100%.

Table V summarizes overall results with respect to H_2 & H_3 for each examined quality. All template systems have some positive effect on each quality (H_2 holds). MASTeR leads the field over most categories. Solely for correctness, DODT achieves the highest average over all related metrics. However, for appropriateness and correctness, several template systems share the highest effect size rank. Here, as well as for unambiguity, differences in effect sizes (H_3) are still noticeable, but relatively small. SPIDER negatively influences metrics related to multiple categories, thus, a very small negative effect can be observed, besides for unambiguity, for completeness and verifiability. In addition, it has to be noted, that the very small correctness effect for EARS is solely based on corrected grammar and spelling, thus, no proper effect can be observed for this template system in this category.

Table VI summarizes results with respect to H_4. MASTeR has the strongest positive effect for all guideline-specific

TABLE IV
EFFECT SIZES OF UNAMBIGUITY METRICS OVER ALL REQUIREMENTS (EFFECT SIZE, MAGNITUDE $\in [XS..XXL]$, RAW EFFECT)

	%Risk / ∅ control	EARS			MASTER			Adv-EARS			DODT			SPIDER		
R1 use only one sentence	16%	0	XXL	-16%	0	XXL	-16%	0	XXL	-16%	0	XXL	-16%	0	XXL	-16%
R2 #words	23.1	0.29	S	-3	0.53	M	-5	**0.56**	**M**	**-6**	0.48	S	-5	0.27	S	-3
R3 use one process-verb	39%	0.54	L	-18%	**0.03**	**XL**	**-38%**	0.40	XL	-23%	0.27	XL	-28%	0.40	XL	-23%
R4 a) #punctuations/1k words	*145*		-		***0.43***	***S***	***-39***		-			-			-	
b) #punctuations/1k words < 209	18.9%	0.72	M	-5%	**0.38**	**XL**	**-12%**	0.69	M	-6%	0.62	M	-7%		-	
R5 use modal verb for liability	0%		-			-			-			-		**∞**	**XXL**	**+100%**
R6 use simple structured sentence	8.8%	0	XXL	-9%	0	XXL	-9%	0	XXL	-9%	0	XXL	-9%	0	XXL	-9%
R7 use appropriate abstraction level	8.8%	0.33	XL	-6%	0.36	XL	-6%	**0.25**	**XL**	**-7%**	0.41	L	-5%	0.44	L	-5%
R8 use active voice	39%	0.61	M	-15%	**0.39**	**XL**	**-24%**	0.47	L	-21%	0.47	L	-21%		-	
R9 use precise verb	39.4%		-		0.76	S	-9%		-		**0.62**	**M**	**-15%**	0.68	M	-13%
R10 avoid nominalization	37%		-			-			-			-			-	
R11 avoid light-verb constructions	4.4%		-		**0.39**	**XL**	**-3%**	0.41	L	-3%	0.49	L	-2%		-	
R12 use full verb	59%		-			-			-		**0.76**	**S**	**-14%**	0.84	S	-9%
R13 avoid comparison	10%		-		**0.61**	**M**	**-4%**		-			-			-	
R14 use clear comparison	3.6%		-			-			-			-		**0**	**XXL**	**-4%**
R15 definite articles	46.2%		-		**0.67**	**M**	**-16%**	0.77	S	-11%		-		0.71	M	-14%
R16 use defined units	0%		-			-			-			-			-	
R17 avoid vague terms	31.7%	0.74	M	-8%	0.59	L	-13%	0.61	M	-12%	0.52	L	-15%	**0.47**	**L**	**-17%**
R18 avoid escape clauses	0.8%	**0**	**XXL**	**-1%**	**0**	**XXL**	**-1%**	**0**	**XXL**	**-1%**		-			-	
R19 avoid open-ended clauses	8.4%	0.48	L	-4%	**0**	**XXL**	**-8%**	0.09	XL	-8%	0.09	XL	-8%	0.04	XL	-8%
R20 avoid superfluous infinitives	9.6%		-			-		0.04	XL	-9%		-		**0**	**XXL**	**-10%**
R21 use correct grammar/spelling	10.8%	0	XXL	-11%	0	XXL	-11%	0	XXL	-11%	0	XXL	-11%	0	XXL	-11%
R22 avoid negations	17.7%		-		**0.66**	**M**	**-6%**		-			-			-	
R23 avoid /	7.2%		-		**0.61**	**M**	**-3%**		-			-			-	
R24 avoid combinators	51%		-		**0.42**	**L**	**-30%**	0.83	S	-9%	0.84	S	-8%		-	
R27 avoid group-nouns	20.5%		-			-			-			-			-	
R28 avoid pronouns	20.5%	0.67	M	-7%	**0.39**	**XL**	**-12%**	0.48	L	-11%	0.44	L	-11%	**2.77**	**XXL**	**+36%**
R29 context free	23.7%		-			-			-			-		**0.72**	**M**	**-7%**
R30 avoid absolutes	15.7%		-		0.73	M	-4%		-		**0.65**	**M**	**-6%**	**3.83**	**XXL**	**+44%**
R31 use explicit conditions	5.2%		-		0.56	L	-2%		-		0.56	L	-2%	**0.07**	**XL**	**-5%**
R32 use clear condition combination	2.8%	0.13	XL	-2%	**0**	**XXL**	**-3%**	0.13	XL	-3%	0.39	XL	-2%	0.25	XL	-2%
R34 use clear quantifiers	15.3%		-			-			-			-			-	
R35 use value tolerances	8%		-		**0.46**	**L**	**-4%**	0.58	L	-3%	0.63	M	-3%		-	
R36 express one atomic need	34.5%		-		**0.08**	**XL**	**-32%**	0.71	M	-10%	0.79	S	-7%	0.76	S	-8%
R37 use clear preconditions	8%		-			-			-			-			-	
R38 use clear business logic	2.8%		-			-			-			-			-	
R39 use clear subject	8%		-		**0**	**XXL**	**-8%**		-			-			-	
Flesch-Kincaid Grade Level	12.3		-		**0.2**	**S**	**-1**		-			-		**0.5**	**M**	**+2**
Summary Effect Size			small			**medium**			small			small			small	
Negative Effect															**very small**	

TABLE V
∅ EFFECT SIZE MAGNITUDES PER QUALITY (POSITIVE/NEGATIVE)

Quality	EARS	MASTER	Adv-EARS	DODT	SPIDER
Appropriateness	**medium**	**medium**	**medium**	small	small
Unambiguity	small	**medium**	small	small	small / very small
Completeness	very small	**medium**	small	very small	small / very small
Singularity	medium	**very large**	large	medium	large
Verifiability	very small	**medium**	small	small	small / very small
Correctness	very small	small	small	**small**	very small
Conformity	medium	**huge**	very large	large	large
Summary	small	**large**	medium	small	small / very small

TABLE VI
∅ EFFECT SIZE MAGNITUDES PER GUIDELINE (POSITIVE/NEGATIVE)

Guideline	EARS	MASTER	Adv-EARS	DODT	SPIDER
ISO/IEC/...29148 [7]	small	**large**	small	small	small / very small
INCOSE GWR [29]	small	**large**	medium	small	small
SOPHIST Rules [44]	very small	**medium**	small	small	small
ECSS-ST-E-10-06 [45]	small	**large**	medium	small	medium / very small
ECSS Drafting Rules [46]	small	**medium**	small	small	small / very small
NASA Checklist [47]	small	**medium**	medium	small	small / very small

metric compilations. However, Adv-EARS falls into the same effect size magnitude for the NASA guide. Although it also has positive effects, SPIDER has negative effects on metrics contained in each guideline except SOPHIST. Yet, for the INCOSE Guide, though, the mean effect size category is < 0.5.

TABLE VII
SPEARMAN RANK CORRELATION MATRIX OF SELECTED METRICS OVER THE DOCUMENT GROUPS ($n = 30$, $\alpha = 0.05$ OR **0.01** [59]).

	% avoid pronouns (R28)	% avoid negations (R22)	% use clear quantifiers (R34)	F-Score [50]	Coleman-Liau Index [64]	∅ Readability Scores	∅ #words per requirement
% avoid pronouns (R28)	**1**	-	-	**0.81**	0.4	-	-
% avoid negations (R22)	-	**1**	-	-	-	-	-0.38
% use clear quantifiers (R34)	-	-	**1**	-	-	-	-
F-Score [50]	**0.81**	-	-	**1**	-	-	-
Coleman-Liau Index [64]	0.4	-	-	-	**1**	**0.84**	**0.56**
∅ Readability Scores	-	-	-	-	**0.84**	**1**	**0.78**
∅ #words per requirement	-	-0.38	-	-	**0.56**	**0.78**	**1**

Table VII summarizes selected results from the correlation analysis of all metrics. The full matrix can be obtained online [52]. For hypotheses H_5, that quantifiers impair readability, as well as H_6, that the use of pronouns correlates with shorter sentences, we do not observe any significant correlation. Thus, we have to reject both hypotheses. H_7, that there is no negative correlation of readability and the use of pronouns, holds for all readability scores. Yet, solely for the Coleman-Liau Index [64], a small correlation ($r_s \approx 0.4$, $\alpha = 0.05$, $n = 30$) supports the idea that pronouns can even improve readability [32], [33], [34]. For negations, where similar assumptions are made [32], [33], [34], we only observe a small debilitating correlation of shorter sentences with the avoidance of negations ($r_s = 0.37$, $\alpha = 0.05$, $n = 30$).

V. DISCUSSION

A. Assumptions and Implications

As expected, we can confirm the main observations from earlier studies [4], [3] on reduction of ambiguity and length as well as increase in total number of requirements, not only for EARS but all examined template systems (*H_1 & H_2*). The stronger increase in requirement quantity for MASTeR sticks out. It is rooted in a more consequent ban of lists of subjects, objects, and conditions, which also reflects in its stronger effect on R24 *"avoid combinators"*. However, our phrasings of the Certification Specifications for Engines [38] with EARS differ from the original examples [4], where more (reasonable) improvements were applied, which are not explicitly covered by the syntax description. We took the "naïve" approach on purpose to compare the different syntax structures, although the mindset associated with the use of templates tends to encourage further improvement. Yet, this might be less accessible by novices than if it is immediately part of the structure. In turn, a more complex structure might impair usability and learnability. Further, it is assumed that constraining a natural language inevitably reduces its expressiveness [65]. Conceptually, MASTeR are the only ones that explicitly support non-functional categories. Nevertheless, unlike in [24], all requirements from our dataset are expressible in all template systems. Yet, this is achieved with different effort and in some cases it seems "affectedly". Future research should investigate these trade-offs with usability and expressiveness.

In general, the observed effects of templates on requirements quality are smaller than expected. This might be due to the fact, that the original requirements in our dataset are not preliminary ones of low quality, but final versions that are already partially structured. Specifically, the TSS and EVS requirements, which make up half of the dataset, already score well for most metrics in their original form. Yet, there are stronger effects for documents with initially lower quality, particularly EASA's CSE [38]. Though, the mostly insignificant results for readability are surprising. Yet, we expected lower scores for SPIDER, as there is a negative correlation between readability and formality assumed [66]. Meanwhile, SPIDER has a negative effect on the F-Score [50] and no significant correlation of readability with the F-Score can be found. Thus, the kind of formality relevant to model-based development is presumably better covered with other metrics on template meta-models.

Nevertheless, positive effects are observed for all template systems in all seven quality categories (*H_2*, c.f. Table V). Thus, potentially, by using templates, several risks of quality deficiencies can be reduced and the conformance to guidelines can be enhanced. MASTeR appears to have the strongest effect for the examined guidelines, while SPIDER has some structural non-conformance (*H_3 & H_4*). However, adjustments, e.g., to add a modal verb to SPIDER, are easily made, and it is up to the users to argue, if some rule is crucial or really violated: e.g., if SPIDER's scope indicators are considered to be *absolutes* (R30) in the negative sense. The selection and weighting of relevant quality rules is always dependent on the notion of quality that prevails in the project phase & context.

This is why only limited insights can be gained from combined metrics [67] and any general weighting of different qualities or rules seems high handed. This also applies for the relativity of effect sizes towards the raw effect or the baseline risk/mean in the control group. A very common smell, like in our data the use of indefinite articles (R15), could be considered much less important than a rather rare one, like e.g., the lack of explicit conditions (R31). Thus, we did not offset the aggregated effect size against these indicators. These summaries only provide some tendency, while detailed results enable users to make an informed decision based on their individual quality needs. The comparison of guideline rule sets, presented in Table II, reveals not only different focus of the guidelines, but also potential gaps. Further, it becomes apparent that the rules, even when individually stated in the same guideline, are not fully disjoint. E.g., "use full verb" (R12) includes the avoidance of nominalizations as well as light-verb constructions and others have strong correlations. This could be used to improve or create custom guidelines. Yet, our results neither clearly support nor refute doubts about quality rules for quantifiers, pronouns and negations (H_5–H_7) as raised in earlier work [36], [32], [33], [34].

B. Threats to Validity

Threats to experimentation in software engineering [13] are:

Construct Validity. Selected quality factors are derived from template system goals and metrics are chosen from literature. Using templates includes a learning effort and need to understand the underlying semantics [3], [68]. While syntactic compliance can be checked [69], [12], the correct choice/use of templates remains a manual, error-prone task. It is difficult to solely improve requirements as really encouraged by the templates. By discussions among the researchers, we try to reduce such effects. In practice, templates are used by people with similar education and experience level facing the same ambiguities. Complete separation from effects of original document context is impossible. To reduce this influence, we use documents from different domains and levels of abstraction.

External Validity. The sample comprises 249 requirements out of five projects. This is not enough to generalize from differences of individual documents to performance on different categories of specifications. E.g., the majority of requirements in the dataset are functional requirements on the system level and already roughly comply to MASTeR, where $\approx 50\%$ are from one project (EVS). Yet, the experiments use real world data. Even the lecture projects TSS & EVS are specified to be implemented as usable tools for university routine.

Internal Validity. Potential threats arise from the interrelation with the original documents and human interaction. The same measures to reduce threats to construct validity apply. Further, effect size, correlation, and other statistical measures are evaluated, to test, if observed effects are significant and truly correlate with the treatment. The light weight spreadsheet-based approach is chosen to avoid bias by external tools and their limitations.

Repeatability. The original and retrieved data, as well as custom scripts, are available online [52] and metrics definitions as well as data collection procedures are documented. Analysis results can thus be reproduced by independent researchers as well as repeated on independent data.

Conclusion Validity. By following the IEEE 1061 [43] methodology, metrics and critical values are defined before the data collection to avoid bias through selection of criteria to observe. Yet, other researcher may come to different results in initial rephrasing and rule evaluation.

C. Future Work

In future research, additional recent templates systems, like that of Mazo & Jaramillo [24] or *FRETISH* [25] should be covered. Further, tools need to be explored that support a higher degree of automation, e.g., in evaluation of correct pattern usage [69], [12] or quality analysis [70], [71], [72], [73] to enable replication on larger data sets, as *PROMISE* [74] *NFR* [75] or *PURE* [76], The other way round, obtained data about inter-dependencies of metrics and quality attributes can be used to improve such analysis tools.

Moreover, user experiments could evaluate the practical usability of the different notations. Based on preliminary pilots, we currently work on a study design to address this. One further aspect of usability is how the user experience of supporting editor tools influences the acceptance of such more formal notations and their learnability [77]. In addition, we currently conduct experiments to compare *expressiveness* and *formality* of template systems based on their meta-models.

VI. Conclusion

We identify relevant quality factors to compare the phrasing quality achieved with different requirement template systems and present a respective metric suite and experimental setting. Initial experiments are conducted with EARS, MASTeR, Adv-EARS, boilerplates (DODT), and SPIDER templates applied to 249 requirements from five real-world projects. Re-phrased to the different variants, this leads to a dataset of in total 1764 requirements with five control and 25 treatment groups.

With respect to the research question, it can be shown, that the usage of templates is generally an appropriate means to raise requirements quality in many facets and that the template systems perform different for various quality rules. MASTeR leads the field in terms of aggregated effect size for all six examined guidelines and 6/7 quality aspects.

Yet, only limited insights can be gained from aggregated metrics [67]. The individual definition of high quality, e.g., by selection and/or prioritization of quality rules, is highly context dependent. In general, the template systems perform relatively similar, what supports the assumption that it is important to follow *some* phrasing guideline to obtain uniform requirements standardized towards specified quality criteria, but the specific notation is potentially subordinate and a matter of personal preference. However, results from our experiments enable practitioners to make an informed decision in selecting a template system that fits (better) with guidelines relevant to a domain, project, or organization context. The choice of an adequate template system can be based on individual detailed results for a custom selection of quality factors. Similar, this information can be used to develop pinpoint improvements or domain specific adaptions for template systems. Further, insights on dependencies between different metrics and coverage of different guidelines could be used to improve guidelines as well as quality analysis tools.

Yet, limited insights to formality & expressiveness motivate further research on the meta-model level of template systems.

Acknowledgment

We gratefully acknowledge financial support from the ESA NPI program No. 4000118174/16/NL/MH/GM and project "NaWi" line of funding "(Post-)Doktorand*innen mit Kind", as well as fruitful discussions, data, and tech support from S. Ahmadian, C. Braun, F. Caballero, T. Dabbert, C. Hartenfels, A. Jung, R. Naujokat, S. Peldszus, and V. Riediger.

References

[1] SwissQ. (2014) Software Development Trends & Benchmarks Report Schweiz. [Online]. Available: https://swissq.it/wp-content/uploads/2016/02/Agile_RE_Testing-Trends_und_Benchmarks2014.pdf

[2] S. Konrad and B. H. C. Cheng, "Real-Time Specification Patterns," in *27th International Conference on Software Engineering (ICSE'05)*, 2005, pp. 372–381.

[3] A. Mavin and P. Wilkinson, "Ten Years of EARS," *IEEE Software*, vol. 36, no. 5, pp. 10–14, 2019.

[4] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy Approach to Requirements Syntax (EARS)," in *17th IEEE International Requirements Engineering Conference (RE'09)*, 8 2009, pp. 317–322.

[5] T. Arnuphaptrairong, "Top ten lists of software project risks : Evidence from the literature survey," in *International MultiConference of Engineers and Computer Scientists (IMECS)*, vol. I, 2011.

[6] C. Rupp and R. Joppich, "Anforderungsschablonen," in *Requirements-Engineering und -Management*, 6th ed. Carl Hanser Verlag München, 2014, pp. 215–246.

[7] *ISO/IEC/IEEE 29148: Systems and software engineering–Life cycle processes–Requirements engineering*, ISO/IEC/IEEE Std. ISO/IEC/IEEE 29 148:2018(E), 11 2018.

[8] R. Wieringa, E. Dubois, and S. Huyts, "Integrating semi-formal and formal requirements," in *Advanced Information Systems Engineering*, A. Olivé and J. A. Pastor, Eds., 1997, pp. 19–32.

[9] C. Schauer and H. Schauer, "Modellierungstechniken für das Wissensmanagement," *LOG IN*, vol. 30, no. 166/167, pp. 28–37, 2010.

[10] S. Farfeleder, T. Moser, A. Krall, T. Stålhane, I. Omoronyia, and H. Zojer, "Ontology-driven guidance for requirements elicitation," in *The semantic web: research and applications*. Springer, 2011, pp. 212–226.

[11] O. Türetken, O. Su, and O. Demirörs, "Automating software requirements generation from business process models," in *1st Conference on the Principles of Software Engineering (PRISE)*, 2004.

[12] L. Lúcio, S. Rahman, C.-H. Cheng, and A. Mavin, "Just Formal Enough? Automated Analysis of EARS Requirements," in *NASA Formal Methods*, C. Barrett, M. Davies, and T. Kahsai, Eds. Springer, 2017, pp. 427–434.

[13] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting experiments in software engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer London, 2008, pp. 201–228.

[14] F. Pace and V. Barrena, "EARTH OBSERVATION REFERENCE MISSION - SW SPECIFICATIONS," ESA - ESTEC, Tech. Rep. ATB-RAC-D8-D, 2010.

[15] S. Farfeleder, "Requirements specification and analysis for embedded systems," Ph.D. dissertation, Vienna University of Technology, 2012.

[16] A. Mavin, "Listen, Then Use EARS," *IEEE Software*, vol. 29, no. 2, pp. 17–18, 2012.

[17] D. Majumdar, S. Sengupta, A. Kanjilal, and S. Bhattacharya, "Automated Requirements Modelling with Adv-EARS," *International Journal of Information Technology Convergence and Services (IJITCS)*, vol. 1, no. 4, pp. 57–67, 8 2011.

[18] R. Joppich. (2014, 8) MASTER-Schablonen für Bedingungen. [Online]. Available: https://www.sophist.de/publikationen/requirements-engineering-und-management/

[19] S. Konrad and B. Cheng, "Facilitating the construction of specification pattern-based properties," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, 01 2005, pp. 329–338.

[20] C. Rupp. (2014) Requirements templates - the blueprint of your requirement. [Online]. Available: https://www.sophist.de/publikationen/requirements-engineering-und-management/

[21] E. Simmons, "Quantifying Quality Requirements Using Planguage," in *Intel's Quality Week*, 2001.

[22] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*, 2nd ed. Springer, 2005.

[23] O. Daramola, G. Sindre, and T. Stalhane, "Pattern-based security requirements specification using ontologies and boilerplates," in *2nd IEEE International Workshop on Requirements Patterns (RePa)*, 2012, pp. 54–59.

[24] R. Mazo, P. Vallejo, C. A. Jaramillo, and J. H. Medina, "Towards a new template for the specification of requirements in semi-structured natural language," *Journal of Software Engineering Research and Development*, vol. 8, pp. 3:1–3:16, 2020.

[25] D. Giannakopoulou, A. Mavridou, J. Rhein, T. Pressburger, J. Schumann, and N. Shi, "Formal Requirements Elicitation with FRET," in *Joint Proceedings of REFSQ-2020 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track*, M. Sabetzadeh, A. Vogelsang, S. Abualhaija, M. Borg, F. Dalpiaz, M. Daneva, N. C. Fernández, X. Franch, D. Fucci, V. Gervasi, E. Groen, R. Guizzardi, A. Herrmann, J. Horkoff, L. Mich, A. Perini, and A. Susi, Eds., 2020. [Online]. Available: http://ceur-ws.org/Vol-2584/PT-paper4.pdf

[26] A. Mavin and P. Wilkinson, "Big Ears (The Return of "Easy Approach to Requirements Engineering")," in *18th IEEE International Requirements Engineering Conference*, 2010, pp. 277–282.

[27] L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej, "Empirical research on requirements quality: a systematic mapping study," *Requirements Engineering*, vol. 27, no. 2, pp. 183–209, 2022.

[28] J. Frattini, L. Montgomery, J. Fischbach, M. Unterkalmsteiner, D. Mendez, and D. Fucci, "A live extensible ontology of quality factors for textual requirements," in *30th IEEE International Requirements Engineering Conference (RE'22)*. IEEE, 2022.

[29] Requirements Working Group, "Guide for Writing Requirements," International Council on Systems Engineering (INCOSE), Tech. Rep. INCOSE-TP-2010-006-03, 2019.

[30] A. Mavin, "Applying requirements templates in practice: Lessons learned," in *Requirements-Engineering und -Management*, 6th ed. Carl Hanser Verlag München, 2014, pp. 244–246.

[31] *ISO/IEC/IEEE 29148: Systems and software engineering–Life cycle processes–Requirements engineering*, ISO/IEC/IEEE Std. ISO/IEC/IEEE 29 148:2011(E), 12 2011.

[32] A. Condamines and M. Warnier, "Linguistic analysis of requirements of a space project and their conformity with the recommendations proposed by a controlled natural language," in *4th International Workshop Controlled Natural Language (CNL)*, B. Davis, K. Kaljurand, and T. Kuhn, Eds., 2014, pp. 33–43.

[33] M. Warnier, "How can corpus linguistics help improve requirements writing? specifications of a space project as a case study," in *23rd IEEE International Requirements Engineering Conference (RE'15)*, 2015, pp. 388–392.

[34] H. Femmer, D. Méndez Fernández, E. Juergens, M. Klose, I. Zimmer, and J. Zimmer, "Rapid requirements checks with requirements smells: Two case studies," in *1st International Workshop on Rapid Continuous Software Engineering (RCoSE'14)*, 2014, pp. 10–19.

[35] M. Warnier and A. Condamines, "A case study on evaluating the relevance of some rules for writing requirements through an online survey," in *25th IEEE International Requirements Engineering Conference (RE'17)*, 2017, pp. 243–252.

[36] K. Winter, H. Femmer, and A. Vogelsang, "How do quantifiers affect the quality of requirements?" in *26th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'20)*, N. Madhavji, L. Pasquale, A. Ferrari, and S. Gnesi, Eds., 2020, pp. 3–18.

[37] J. Fischbach, J. Frattini, D. Mendez, M. Unterkalmsteiner, H. Femmer, and A. Vogelsang, "How do practitioners interpret conditionals in requirements?" in *Product-Focused Software Process Improvement*, L. Ardito, A. Jedlitschka, M. Morisio, and M. Torchiano, Eds. Springer International Publishing, 2021, pp. 85–102.

[38] *Certification Specifications for Engines*, European Aviation Safety Agency (EASA) Std. CS-E, Amd 1, Annex to ED Decision 2007/015/R, 2007. [Online]. Available: https://www.easa.europa.eu

[39] S. Williams, R. Power, and A. Third, "How easy is it to learn a controlled natural language for building a knowledge base?" in *4th International Workshop Controlled Natural Language (CNL)*, B. Davis, K. Kaljurand, and T. Kuhn, Eds., 2014, pp. 20–32.

[40] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*. Springer, New York, NY, 1992.

[41] V. Johannessen, "CESAR - text vs. boilerplates: What is more effcient– requirements written as free text or using boilerplates (templates)?" Master's thesis, Norwegian University of Science and Technology, 2012.

[42] C. Rupp and SOPHIST GmbH, *Requirements-Engineering und -Management*, 4th ed. Carl Hanser Verlag München, 2007.

[43] *IEEE Standard for a Software Quality Metrics Methodology*, IEEE Std. IEEE 1061-1998, 12 1998.

[44] C. Rupp and A. Günther, "Das SOPHIST-REgelwerk," in *Requirements-Engineering und -Management*, 6th ed. Carl Hanser Verlag München, 2014, pp. 123–164.

[45] ECSS Secretariat and ESA-ESTEC Requirements & Standards Division, *Space engineering - Technical requirements specification*, ECSS Std. ECSS-E-ST-10-06C, 3 2009.

[46] ——, *ECSS - Draft rules and template for ECSS Standards*, ECSS Std. ECSS-D-00-01C, 5 2014.

[47] M. Alexander, M. Allen, E. Baumann, C. Bixby, B. Boland, T. Brady, L. Bromley, M. Brown, M. Brumfield, P. Campbell, D. Carek, and R. Cox, "NASA SYSTEMS ENGINEERING HANDBOOK," NASA, Tech. Rep. NASA SP-2016-6105 Rev2, 2016. [Online]. Available: https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook

[48] W. H. DuBay. (2004, 8) The principles of readability. [Online]. Available: https://eric.ed.gov/?id=ed490073

[49] R. Plösch, A. Dautovic, and M. Saft, "The value of software documentation quality," in *14th International Conference on Quality Software*, 2014, pp. 333–342.

[50] F. Heylighen and J.-M. Dewaele, "Variation in the contextuality of language: an empirical measure," *Foundations of Science*, vol. 7, no. 3, pp. 293–340, 2002.

[51] ——, "Formality of language: definition, measurement and behavioral determinants," Center "Leo Apostel", Free University of Brussels, Internal Report, 1999.

[52] K. Großer, M. Rukavitsyna, and J. Jürjens. (2023) Evaluation of templates for requirements documentation: Data-set and sources. [Online]. Available: https://doi.org/10.5281/zenodo.8020672

[53] L. K. Alexander, B. Lopes, K. Ricchetti-Masterson, and K. B. Yeatts, "Common measures and statistics in epidemiological literature," in *ERIC notebook*, 2nd ed. Chapel Hill-NC: Epidemiologic Research and Information Center (ERIC), 2015.

[54] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Lawrence Erlbaum Associates, 1988.

[55] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908. [Online]. Available: http://www.jstor.org/stable/2331554

[56] S. S. Sawilowsky, "New effect size rules of thumb," *Journal of Modern Applied Statistical Methods*, vol. 8, no. 2, pp. 597–599, 2009.

[57] C. Spearman, "The Proof and Measurement of Association between Two Things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[58] J. C. F. de Winter, S. D. Gosling, and J. Potter, "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes," *Psychological Methods*, vol. 21, no. 3, pp. 273–290, 2016.

[59] M. C. Whitlock and D. Schluter, *The Analysis of Biological Data*. Roberts & Company Publishers, 2008. [Online]. Available: https://www.zoology.ubc.ca/~bio300/StatTables.pdf

[60] ECSS Secretariat and ESA-ESTEC Requirements & Standards Division, *Space engineering - Satellite attitude and orbit control system (AOCS) requirements*, ECSS Std. ECSS-E-ST-60-30C, 8 2013.

[61] B. Scott. (2023) Readability formulas. [Online]. Available: https://readabilityformulas.com

[62] V. Cook, "Standard punctuation and the punctuation of the street," in *Essential Topics in Applied Linguistics and Multilingualism*, M. Pawlak and L. Aronin, Eds. Springer, 2014, pp. 267–290. [Online]. Available: http://www.viviancook.uk/Punctuation/PunctFigs.htm

[63] J. P. Kincaid, R. P. J. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel," U.S. Naval Technical Training Command, Naval Air Station Memphis - Millington, TN, Tech. Rep. Research Branch Report 8-75, 1975.

[64] M. Coleman and T. L. Liau, "A computer readability formula designed for machine scoring." *Journal of Applied Psychology*, vol. 60, no. 2, pp. 283–284, 1975.

[65] S. Boyd, D. Zowghi, and A. Farroukh, "Measuring the expressiveness of a constrained natural language: an empirical study," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005, pp. 339–349.

[66] A. C. Graesser, D. S. McNamara, Z. Cai, M. Conley, H. Li, and J. Pennebaker, "Coh-metrix measures text characteristics at multiple levels of language and discourse," *The Elementary School Journal*, vol. 115, no. 2, pp. 210–229, 2014.

[67] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaram, A. Ta, and M. Theofanos, "Identifying and measuring quality in a software requirements specification," in *1st International Software Metrics Symposium*, 1993, pp. 141–152.

[68] A. Mavin, P. Wilksinson, S. Gregory, and E. Uusitalo, "Listens Learned (8 Lessons Learned Applying EARS)," in *24th IEEE International Requirements Engineering Conference (RE)*, 2016, pp. 276–282.

[69] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, "Automated checking of conformance to requirements templates using natural language processing," *IEEE Transactions on Software Engineering*, vol. 41, no. 10, pp. 944–968, 2015.

[70] A. Ferrari, G. Gori, B. Rosadini, I. Trotta, S. Bacherini, A. Fantechi, and S. Gnesi, "Detecting requirements defects with NLP patterns: an industrial experience in the railway domain," *Empirical Software Engineering*, vol. 23, no. 6, pp. 3684–3733, 12 2018.

[71] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications," *Requirements engineering*, vol. 13, no. 3, pp. 207–239, 2008.

[72] G. Lami, R. Ferguson, D. Goldenson, M. Fusani, F. Fabbrini, and S. Gnesi, "QuARS: Automated Natural Language Analysis of Requirements and Specifications," *INCOSE International Symposium*, vol. 15, no. 1, pp. 344–353, 2005.

[73] Y. Wang, I. L. Manotas Gutiérrez, K. Winbladh, and H. Fang, "Automatic detection of ambiguous terminology for software requirements," in *18th International Conference on Application of Natural Language to Information Systems (NLDB'13)*. Springer, 2013, pp. 25–37.

[74] J. Sayyad Shirabad and T. J. Menzies. (2005) PROMISE software engineering repository. School of Information Technology and Engineering, University of Ottawa, Canada. [Online]. Available: http://promise.site.uottawa.ca/SERepository/

[75] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements Engineering*, vol. 12, no. 2, pp. 103–120, 2007. [Online]. Available: http://ctp.di.fct.unl.pt/RE2017//downloads/datasets/nfr.arff

[76] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "PURE: A Dataset of Public Requirements Documents," in *25th IEEE International Requirements Engineering Conference (RE'17)*, 2017, pp. 502–505.

[77] J. Dick and J. Llorens, "Using statement-level templates to improve the quality of requirements," in *24th International Conference on Software & Systems Engineering and their Applications (ICSSEA)*, 2012.