Cloud-native Application Quality Model Validation Survey Report

Robin Lichtenthäler, Jonas Fritzsch

Contents

1 Introduction

This report analyzes the answers from a survey conducted through an online questionnaire in the period from October 2022 until December 2022. The aim of the survey was to empirically validate elements of a previously created quality model for cloud-native application architectures. More specifically, the quality model consists of so-called *factors* which represent architectural characteristics of software systems and so-called *quality aspects* which represent higher-level quality attributes mainly adopted from the ISO 25010 standard. Factors and quality aspects are connected through *impacts* which describe how the presence of a factor in a software architecture impacts the respective quality aspect. Through this, a hierarchical quality model can be created. The survey focused on these impacts with the goals of:

1. Confirming the impacts stated in the initially formulated, literature-based quality model
2. Identifying additional impacts not considered yet in the quality model
3. Evaluating the strength of impacts

In the following, the setup of the survey is described in section 2 and some general statistics on how the survey was answered by participants are reported in section 3 together with demographic information in section 4. The main results of the survey regarding the validation of the quality model are presented in section 5 and a conclusion with a short discussion of these results is given in section 6.

2 Survey setup

2.1 Survey Design and Interface

Because the quality model is intended to cover the breadth of the topic of cloud-native applications and considers multiple quality aspects in combination, it has a large number of factors, currently 76, which are arranged in an hierarchical graph with the quality aspects they impact at the top. To validate all these factors in a survey is challenging, therefore we:

1. considered only the impacts from factors on quality aspects, excluding impacts from factors on other factors within the hierarchy,
2. excluded factors merely serving as mediating factors between factors and quality aspects,
3. excluded factors which are less specific to cloud-native applications.

However, this still led to 45 factors for the survey together with 24 quality aspects which can be impacted. To enable participants to state any impact without being biased, all potential impacts, that means 45 * 24 = 1080, need to be considered, leading to the problem of how to design such a survey.

To tackle this, we developed a custom survey frontend which aims to make the rating of impacts from factors on quality aspects simple and intuitive. At the core, a participant is presented with a factor explained by a short description and the quality aspects grouped by their high-level quality aspects. By clicking on a quality aspect, it is possible to rate the impact from that factor on that quality aspect. Therefore, in each question a participant has to consider only one factor. Because the question page is structurally the same for each factor, the participant should get used to the format reasonably quickly.

In addition, we decided that participants could freely choose for how many factors they wanted to provide an answer. In an ideal case, each participant would provide an answer to each factor (45), but the time and effort one is willing to invest in such a survey is limited which limits the number of factors for which a participant is willing to provide an answer. Then again, there are differences between participants: some interested participants might be willing to provide more answers, others are not and might even abort the survey if the number of questions is too high. Thus, instead of asking for a random number of factors, we designed the survey flexible so that each participant could choose the factors to be answered based on their topics of interest and also their available time. Also, for each factor it was possible to choose any number of quality aspects that would be impacted by this factor.

2.2 Survey distribution & participation

The intended target audience for the survey were IT professionals who have practical experience with implementing and deploying web applications on cloud infrastructures. This covers a rather broad range of professionals which is why we also distributed the survey in a broader approach. To ask for participation, we:

- contacted personally known professionals via email
- contacted professionals who had published articles on a fitting topic online
- contacted professionals who held a talk with a fitting topic at industry conferences
- contacted professionals who had published with a fitting topic at scientific conferences
- posted the request for participation on social media, including community groups specifically considering cloud computing

On the welcome page for the survey we stated the purpose of the survey and who would be eligible to participate so that interested persons could decide for themselves whether they wanted to fill out the survey. During the period in which the survey was available online, we received 42 complete submissions. The resulting data from these submissions as well as this report can be found online: https://github.com/r0light/qmsurvey-results

3 General statistics

3.1 Amount and distribution of answers

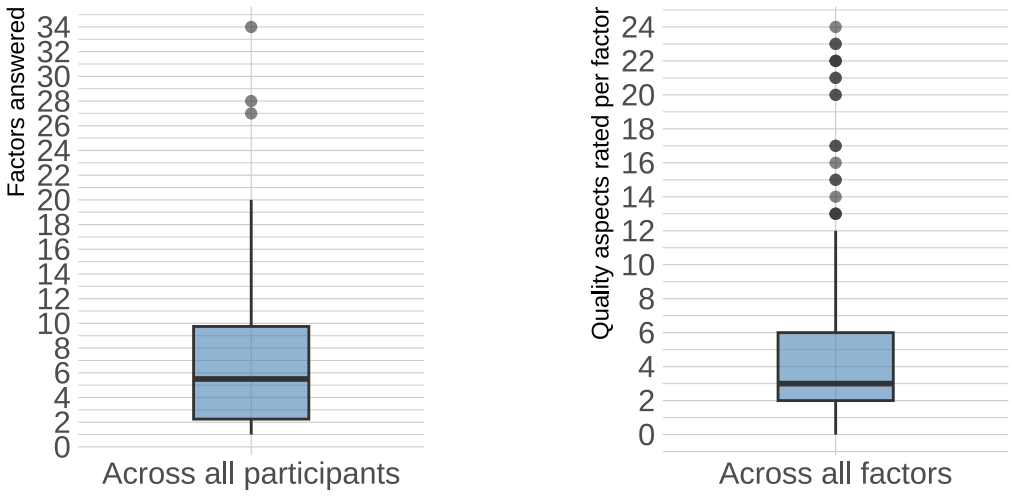


Figure 1: No. of factors answered by participants and No. of ratings stated across all factors

Because of the flexible survey design, it is interesting to analyze how participants made use of this flexibility, that means for how many factors each participant provided an answer and how many quality aspects they selected to rate for each factor. This can be seen in Figure 1. In the left boxplot we can see that the median for how many factors were answered per participant is 5.5 which is quite low compared to the number of 45 factors in total. It shows the difficulty of validating larger quality models using such an online questionnaire. Nevertheless, there were also participants who provided answers to significantly more factors which is helpful and would not have been possible with a non-flexible survey design in which the number of factors to answer was fixed beforehand. The right boxplot shows for how many quality aspects ratings were stated across all factors. For a better understanding of this data it has to be noted that for each quality aspect participants could specify the impact based on the following scheme:

- the factor has a **positive impact (++)** on the quality aspect
- the factor has a **slightly positive impact (+)** on the quality aspect
- the factor has **no impact (0)** on the quality aspect
- the factor has a **slightly negative impact (-)** on the quality aspect
- the factor has a **negative impact (–)** on the quality aspect

The impact type *no impact* was selected as a default for each quality aspect, if no other impact was explicitly stated. The boxplot on the right of Figure 1 shows per factor, for how many of the 24 quality aspects a participant stated impacts explicitly. In the instructions of the survey we told participants that we would expect "*typically between one and five*" quality aspects for which impacts are stated. As the median of 3 shows, most participants stayed within this range, but there are numerous outliers where participants stated impacts for many quality aspects. In one case, a participant stated impacts for all 24 quality aspects which might be the result of a misunderstanding of how to fill out the survey.

In addition, it is interesting to analyze which factors were selected by participants. Figure 2 shows for each factor, by how many participants it was selected to state impacts on quality aspects. While it is difficult to interpret why certain factors where chosen more often than others, it can be seen that there are significant differences between the factors.

Potential reasons might be that certain factors are easier to understand, such as *Service replication* while others such as *Communication partner abstraction* are more abstract and therefore more difficult to evaluate. The number of times answers have been provided for a factor will also be important for the interpretation of the results in section 5, because the results are only interpretable based on the set of participants who provided an answer which is nor representative for the whole set of participants. Additionally, only a larger number of answers leads to more confidence in the interpretation of results.

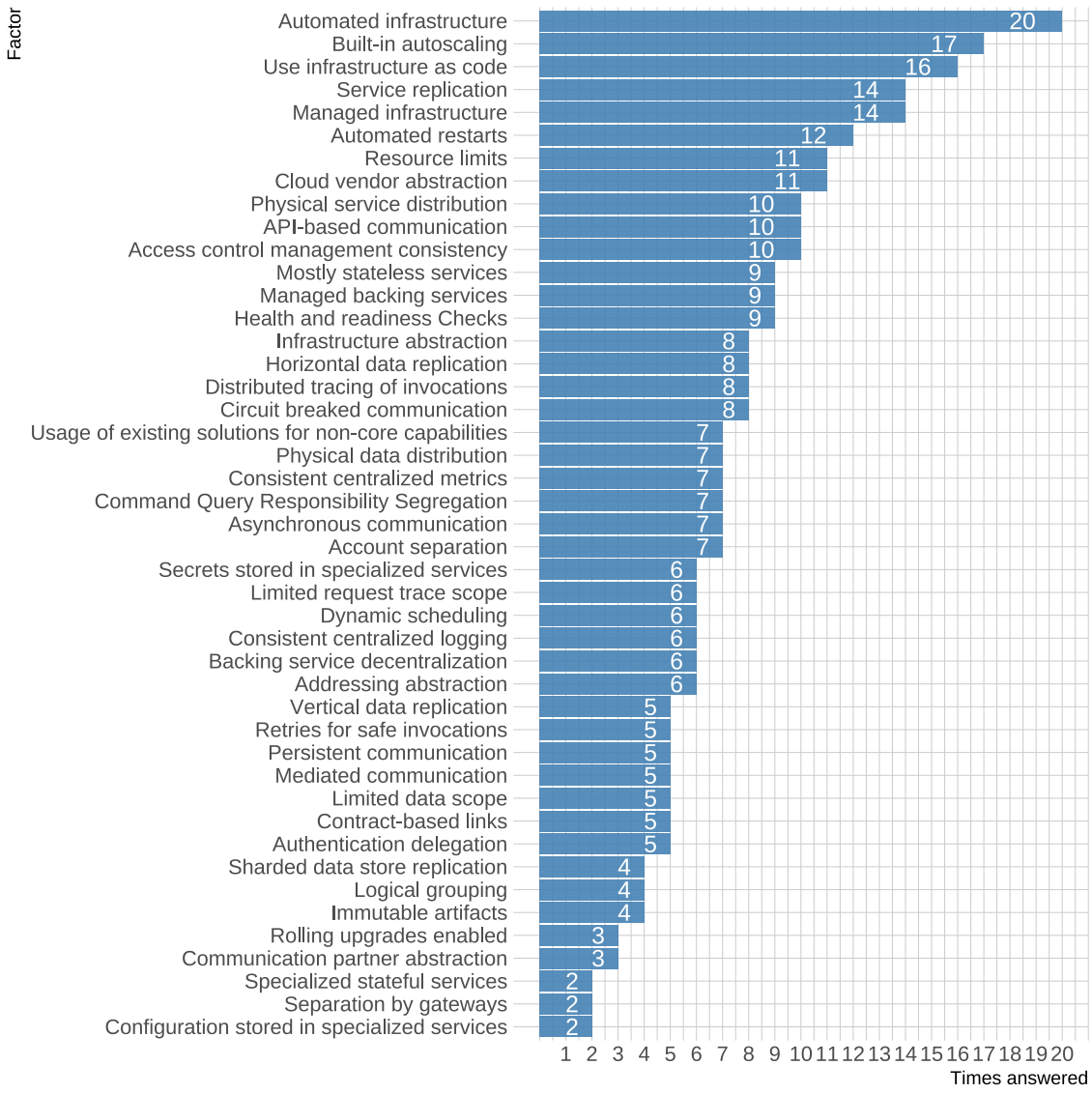


Figure 2: Number of times the individual factors have been answered

3.2 Answer times

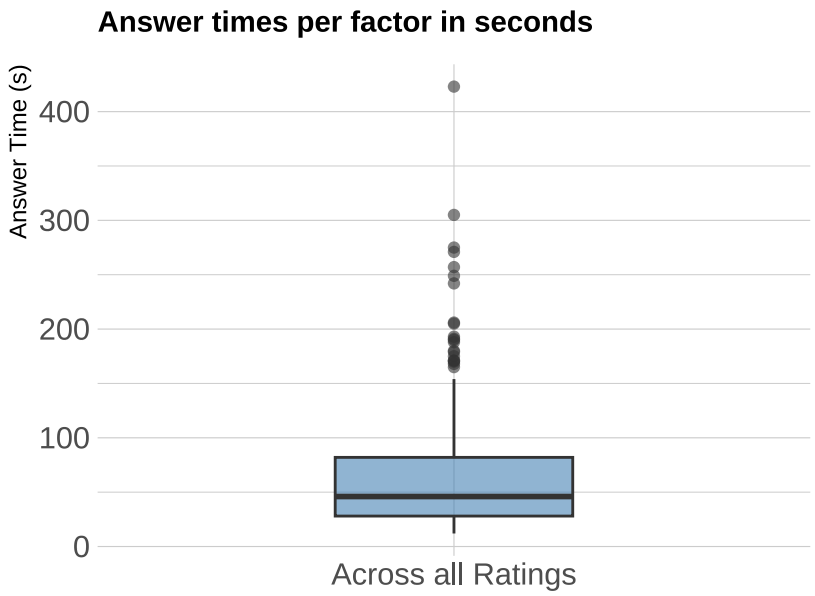
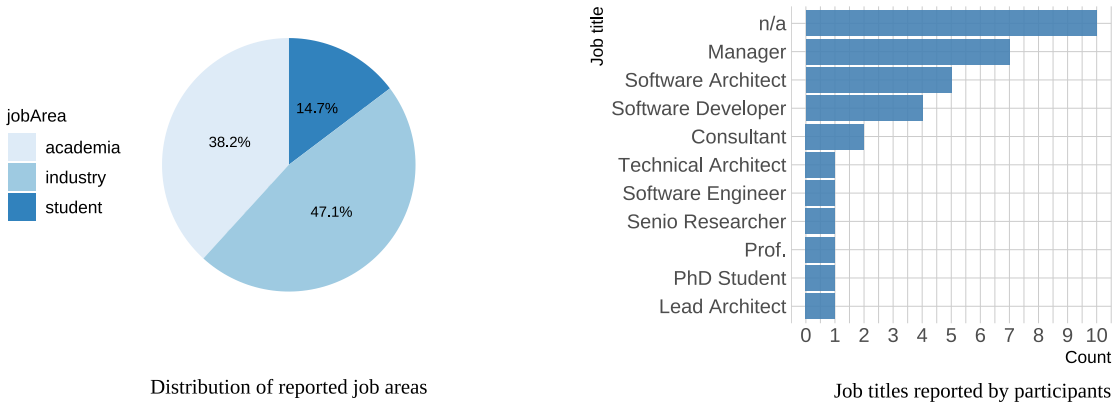


Figure 3: Time spent for answering individual factors

Finally, the survey tool also captured a timestamp each time a factor was answered by a participant. This way, we can to some extent evaluate how much time participants spent on thinking about factors. However, an exact evaluation of such answer times would only be possible in a controlled experiment while with the online distribution of the survey we do not know how participants filled out the survey. For example, it was also possible to start the survey and continue at a different time, because the progress was stored locally in the participant's browser. Therefore, some answer times that were captured showed values of 188262 or 10861 seconds which are the result of such behaviour and therefore need to be excluded. As a countermeasure we firstly excluded obviously erroneous answer time values greater than 3600 seconds and based on the remaining data points, we secondly removed outliers which were greater than the doubled standard deviation. The remaining data is plotted in Figure 3. The median is at 46 seconds, so slightly below one minute. The lower quartile is at 28 seconds and the upper quartile is at 82 seconds. The measured times seem reasonable, although there is also a significant amount of outliers which might be the result of factors that are difficult to answer. This data could be useful for planning future surveys with similar types of questions.

4 Demographics

In addition to the main part of the survey, we also asked some final demographics questions including the primary area to which participants would assign their current job to, their current job title, as well as the years of experience they have with *software development in general* on the one hand and *experience with cloud platforms* on the other hand. It has to be noted that answering these questions was optional and only 34 participants reported demographic data at least partly. Nevertheless, in Figure 4 it can be seen that nearly half of the participants work in the industry area, in contrast to the other half stemming from academia, which provides a diversified field of participants. And as Figure 5 shows, the participants also have significant experience. An expectable, but also interesting fact is that the experience with cloud computing is consistently lower than general software development experience, due to the relative novelty of the technology.



Distribution of reported job areas

Job titles reported by participants

Figure 4: Job areas and job titles stated by participants

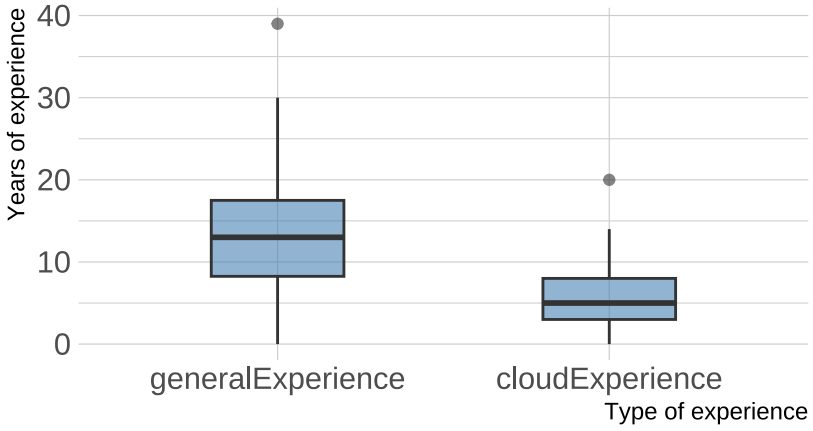


Figure 5: Years of experience reported by participants

5 Results

5.1 Data characteristics

The main results of our survey are the aggregated ratings stated by the participants per **factor-quality aspect** combination. Overall, there is a quite broad distribution of impacts stated by participants. For 629 out of the 1080 potential factor-quality aspect combinations there was at least one impact explicitly stated by participants.

To illustrate how the results for these combinations look like, some exemplary combinations with the aggregated data from the survey are shown in the Figures 6-11. The graphs in these figures show for a single combination, how many times each possible impact rating has been stated by participants. As already mentioned, the impact type `0` was set as a default for all ratings. This means that there is a certain probability that a quality aspect has not been considered for a factor by a participant, just because he or she prioritized other quality aspects, but the rating was nevertheless captured as `0`.

Based on these combinations, we want to derive which impacts exist between the different factors and quality aspects. As stated in the introduction, we want to use these results to validate the initially stated impacts of the quality model for cloud-native application architectures as well as to include additional impacts. To prepare the analysis, we calculated the following measures for each factor-quality aspect combination:

- **weightedMean** (`numeric`): The mean value of ratings weighted by the number of times they have been stated. To enable this calculation we assigned values to impact types in the following way: **--** is interpreted as the value `-2`, **-** as the value `-1`, **0** as the value `0`, **+** as the value +1, and **++** as the value +2.
- **aboveThreshold** (`logical`): Is TRUE if the total number of ratings provided that factor-quality aspect combination is at least the threshold we defined to show some significance. For the analysis we set this threshold to 5.
- **pValue** (`numeric`): Probability of the observed distribution under the null hypothesis. See the following explanation.

To have an indicator for the significance of a result, we used the Exact multinomial test of goodness-of-fit which is suitable when there are multiple values of one nominal variable (the different types of impact) and the sample size is small. Additionally, the individual observations (answers) need to be independent which we can assume, because participants did not see answers from other participants and did the survey presumably alone. For this test we need to formulate a theoretically expected distribution which represents the null hypothesis. We can then calculate the probability of getting the actually observed distribution under the null hypothesis to assess whether the null hypothesis can be rejected. As a distribution for the null hypothesis, we assume that the impact of a factor on a quality aspect is undecidable: That means that all possible ratings are equally likely with an exception of the rating **no impact (0)**. We consider the rating **0** as being twice as likely as the other ratings, because it is selected as the default if the participant did not explicitly state a different rating. For the ratings **--**:**-**:**0**:**+**:**++**, we thus assume a ratio of 1:1:2:1:1 under the null hypothesis. For each factor-quality aspect combination, we can therefore calculate the p value using this test, which we added as **pValue** to each combination. Again, it has to be noted, that these values depend on the number of participants who decided to provide an answer for the respective factor and are therefore not representative for the sample as a whole.

In section 5.3, we describe how we validated the initially stated impacts from the quality model using these measures. And in section 5.4, we explore additionally stated impacts to integrate them in the quality model.

5.2 Examples

Exemplary factor-quality aspect combinations with the results from the survey to illustrate the data.

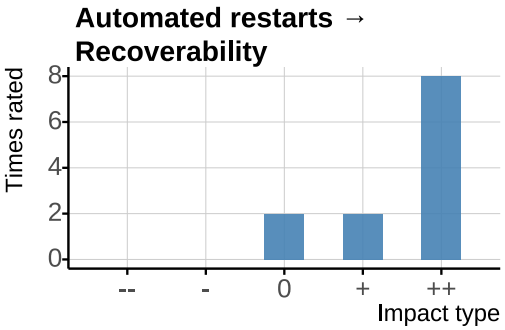


Figure 6: Ratings for the impact from Automated restarts on Recoverability

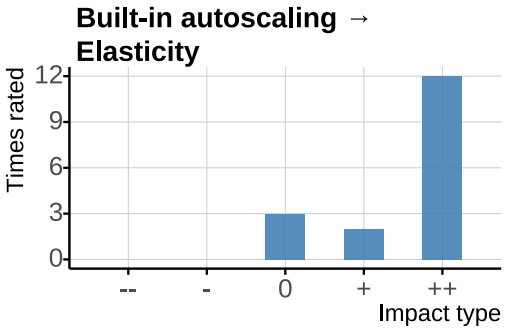


Figure 7: Ratings for the impact from Built-in autoscaling on Elasticity

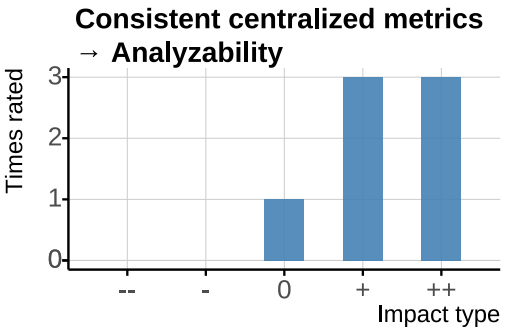


Figure 8: Ratings for the impact from Consistent centralized metrics on Analyzability

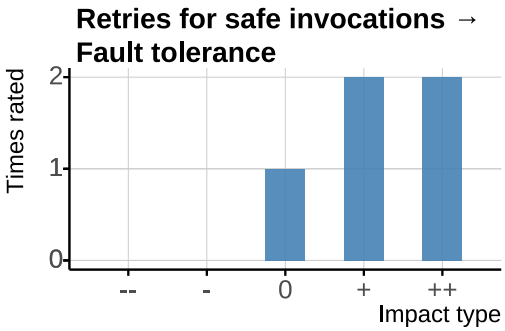


Figure 9: Ratings for the impact from Retries for safe invocations on Fault tolerance

Figure 10: Ratings for the impact from Service replication on Time-behaviour

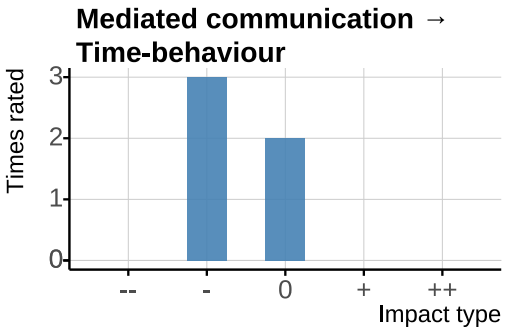


Figure 11: Ratings for the impact from Mediated communication on Time-behaviour

5.3 Validation of previously stated impacts

To assess whether the impacts stated in our initial quality model are confirmed and therefore validated by the survey, we mapped the results for each factor-quality aspect combination to the factor-quality aspect combination with a hypothesized impact from the initial quality model. For each combination, we then used the following R function to decide on whether the hypothesized impact can be validated or not:

```
validateImpact <- function(statedImpact, weightedMean,
                           aboveThreshold, pValue) {

  pThreshold <- 0.1
  validationResult <- ""

  if(weightedMean < 0.5 & weightedMean > -0.5) {
    validationResult <- if(aboveThreshold & pValue < pThreshold) "✗" else "(✗)"
  } else {
    impactTrend <- if(weightedMean > 0) "positive" else "negative"
    if (aboveThreshold & pValue < pThreshold) {
      validationResult <- if(impactTrend == statedImpact) "✓" else "⇄"
    } else {
      validationResult <- if(impactTrend == statedImpact) "(✓)" else "(⇄)"
    }
  }
  return(validationResult)
}
```

The function receives the following input parameters:

- **statedImpact** (`character`): The impact stated in the initial quality model. Is either `positive` or `negative`
- **weightedMean**, **aboveThreshold**, and **pValue** as described in section 5.1

and returns a `character` value describing the validation result.

In essence, an impact is considered valid if the **weightedMean** value is beyond a limit of |0.5| and has the same leading sign as expected based on the hypothesized impact. Additionally, we used a significance level of $\alpha = 0.1$. If the p Value is not below α or the number of answers for that factor is below the set threshold, results are marked as only probable, using brackets. The p Value additionally works as an indicator for how probable a result is. Finally, it is also possible that the analysis of the results shows an impact inverse to the hypothesized one. The possible outcomes are also listed in Table 1.

Table 1: Possible results for the validation of impacts.

Result	Explanation
✓	The impact is valid.
(✓)	The impact is potentially valid.
✗	The impact is invalid.
(✗)	The impact is potentially invalid.
⇄	The impact is inverse.
(⇄)	The impact is potentially inverse.

In the following, we have grouped the outcomes by the individual factors for a better overview and have further separated the factors based on the validation results.

Table 2 includes all factors for which at least one hypothesized impact could be confirmed. Table 3 lists remaining factors with at least potentially valid impacts and finally Table 4 lists those factors for which no hypothesized impacts could be validated or where the result is unclear. Especially the results which are controversial, such as the impact from *Resource limits* on *Resource utilization* in Table 4, indicate that these combinations need further investigation. Possibly, the descriptions provided for these factors and quality aspects can be interpreted in different ways or the factors cover different characteristics which impact quality aspects differently. In any case these factors need to be revised either by describing more clearly what characteristics they cover or by exchanging them with separate factors that cover individual characteristics which have differing impacts on quality aspects.

Overall, several impacts stated in the initial quality model can be validated with the survey and for many there is at least a tendency towards their validation. By tendency we mean that for these impacts individual ratings supported the initially stated impacts, but there were simply not enough answers to rate the impact as validated based on our evaluation.

However, many impacts also could not be validated and might need to be removed from the quality model. One reason why impacts that were stated in the initial quality model could now not be validated might be that we did not consider mediating factors in the survey. Mediating factors are those factors in a hierarchical quality model which impact quality aspects, but are also impacted by other factors. In a sense they "explain" the relationship between lower-level factors and higher-level quality aspects. But when they are missing, an indirect impact from a lower-level factor on a quality aspect might be less obvious. Furthermore, especially the factors in Table 4 consider more abstract characteristics which might be less intuitive to understand and therefore more difficult to rate. Nevertheless, it can make the quality model simpler and therefore better understandable if impacts that could not be validated are removed.

Table 2: Factors with a successful validation of initially stated impacts

Factor	Quality Aspect	hypothesized	- -	-	0	+	++	Impact	Validation	p-Value
Authentication delegation	Authenticity	positive	0	0	0	2	3	++	✓	0.0288066
Automated infrastructure	Modifiability	positive	0	2	12	1	5	+	✗	0.0224132
	Recoverability	positive	0	0	7	3	10	+	✓	0.0005842
Automated restarts	Recoverability	positive	0	0	2	2	8	++	✓	0.0006659
Built-in autoscaling	Elasticity	positive	0	0	3	2	12	++	✓	0.0000051
	Resource utilization	positive	0	0	3	3	11	+	✓	0.0000271
Cloud vendor abstraction	Adaptability	positive	0	0	4	1	6	+	✓	0.0175511
Consistent centralized metrics	Analyzability	positive	0	0	1	3	3	+	✓	0.0559842
	Recoverability	positive	0	0	6	0	1	+	(✗)	0.1163980
Distributed tracing of invocations	Analyzability	positive	0	0	1	2	5	++	✓	0.0121694
	Recoverability	positive	0	0	7	0	1	+	✗	0.0422811
Health and readiness Checks	Analyzability	positive	0	0	5	1	3	+	(✓)	0.2329247
	Recoverability	positive	0	0	5	0	4	+	✓	0.0414333
Infrastructure abstraction	Adaptability	positive	0	0	2	4	2	+	✓	0.0969603
	Modifiability	positive	0	0	5	2	1	+	(✓)	0.3182442
	Recoverability	positive	0	0	6	2	0	+	(✗)	0.1054955
Managed infrastructure	Resource utilization	positive	0	0	7	4	3	+	✓	0.0765035
	Simplicity	positive	0	0	11	1	2	+	✗	0.0092780
Physical service distribution	Availability	positive	0	0	1	1	8	++	✓	0.0001016
Service replication	Time-behaviour	positive	0	0	4	4	6	+	✓	0.0140215
Use infrastructure as code	Modifiability	positive	0	0	12	3	1	+	✗	0.0047745
	Recoverability	positive	0	0	8	3	5	+	✓	0.0378335
	Installability	positive	0	0	8	1	7	+	✓	0.0039382
Vertical data replication	Time-behaviour	positive	0	0	1	0	4	++	✓	0.0288066

Table 3: Factors with potentially valid initially stated impacts

Factor	Quality Aspect	hypothesized	- -	-	0	+	++	Impact	Validation	p-Value
Access control management consistency	Integrity	positive	0	0	4	4	2	+	(✓)	0.1548783
Account separation	Accountability	positive	0	0	4	0	3	+	(✓)	0.1243999
API-based communication	Interoperability	positive	0	0	6	1	3	+	(✓)	0.1655474
	Testability	positive	0	0	4	4	2	+	(✓)	0.1548783
Circuit breaked communication	Fault tolerance	positive	0	0	3	1	4	+	(✓)	0.1455047
Configuration stored in specialized services	Adaptability	positive	0	0	1	0	1	+	(✓)	1.0000000
	Availability	positive	0	1	1	0	0	-	(⇄)	1.0000000
Consistent centralized logging	Analyzability	positive	0	0	1	2	3	+	(✓)	0.1184842
	Recoverability	positive	0	0	5	0	1	+	(✗)	0.1718107
Dynamic scheduling	Modifiability	positive	0	0	6	0	0	0	✗	0.0696159
	Recoverability	positive	0	0	6	0	0	0	✗	0.0696159
	Resource utilization	positive	0	0	2	1	3	+	(✓)	0.3158436
Immutable artifacts	Replaceability	positive	0	0	3	0	1	+	(✓)	0.6296296
Limited data scope	Modularity	positive	0	0	2	2	1	+	(✓)	0.7222222
Logical grouping	Modifiability	positive	0	0	3	0	1	+	(✓)	0.6296296
	Co-existence	positive	0	0	4	0	0	0	(✗)	0.2160494
Managed backing services	Resource utilization	positive	0	0	7	2	0	+	✗	0.0432623
	Simplicity	positive	0	0	4	3	2	+	(✓)	0.3169439
Mostly stateless services	Modularity	positive	0	0	5	1	3	+	(✓)	0.2329247
	Replaceability	positive	0	0	6	2	1	+	(✗)	0.2329247
	Elasticity	positive	0	0	4	1	4	+	(✓)	0.1325017
Physical data distribution	Availability	positive	0	0	2	1	4	+	(✓)	0.1099966
Retries for safe invocations	Fault tolerance	positive	0	0	1	2	2	+	(✓)	0.2695473
Rolling upgrades enabled	Availability	positive	0	0	0	0	3	++	(✓)	0.0185185
Secrets stored in specialized services	Confidentiality	positive	0	0	2	2	2	+	(✓)	0.4547325
	Availability	positive	1	1	4	0	0	-	(⇄)	0.6399177
Separation by gateways	Reusability	positive	0	0	2	0	0	0	(✗)	1.0000000
	Availability	positive	0	0	0	2	0	+	(✓)	0.1111111
	Integrity	positive	0	0	2	0	0	0	(✗)	1.0000000
Sharded data store replication	Time-behaviour	positive	0	0	3	0	1	+	(✓)	0.6296296

Table 4: Factors with (potentially) invalid and unclear initially stated impacts

Factor	Quality Aspect	hypothesized	- -	-	0	+	++	Impact	Validation	p-Value
Addressing abstraction	Interoperability	positive	0	0	5	0	1	+	(✗)	0.1718107
	Modularity	positive	0	0	6	0	0	0	✗	0.0696159
Asynchronous communication	Modularity	positive	0	0	5	1	1	+	(✗)	0.3458505
Backing service decentralization	Modifiability	positive	0	0	5	0	1	+	(✗)	0.1718107
	Co-existence	positive	0	0	6	0	0	0	✗	0.0696159
Command Query Responsibility Segregation	Modularity	positive	0	0	5	2	0	+	(✗)	0.1430041
Communication partner abstraction	Interoperability	positive	0	0	3	0	0	0	(✗)	0.4444444
	Modularity	positive	0	0	2	1	0	+	(✗)	1.0000000
	Analyzability	negative	0	0	3	0	0	0	(✗)	0.4444444
Contract-based links	Interoperability	positive	0	0	5	0	0	0	(✗)	0.1100823
	Testability	positive	0	0	5	0	0	0	(✗)	0.1100823
Horizontal data replication	Time-behaviour	positive	0	0	7	0	1	+	✗	0.0422811
Limited request trace scope	Modifiability	positive	0	0	6	0	0	0	✗	0.0696159
	Co-existence	positive	0	0	5	1	0	+	(✗)	0.1718107
Mediated communication	Interoperability	positive	0	0	4	1	0	+	(✗)	0.3518519
	Modularity	positive	0	0	4	1	0	+	(✗)	0.3518519
Persistent communication	Modularity	positive	0	0	4	1	0	+	(✗)	0.3518519
	Recoverability	positive	0	0	3	2	0	+	(✗)	0.3518519
Resource limits	Resource utilization	positive	2	2	2	3	2	+	(✗)	0.7243259
Specialized stateful services	Modularity	positive	0	0	2	0	0	0	(✗)	1.0000000
	Replaceability	positive	0	0	2	0	0	0	(✗)	1.0000000
	Elasticity	positive	1	0	1	0	0	-	(⇄)	1.0000000
Usage of existing solutions for non-core capabilities	Reusability	positive	0	0	4	3	0	+	(✗)	0.1243999

5.4 Additionally found impacts

Next, we analyzed the survey results for impacts that were not previously considered in the quality model. As written before, there is a broad range of factor-quality aspect combinations for which ratings were stated by the participants. The challenge therefore is to filter out the significant ones which can potentially be included in the quality model.

To determine the type and strength of each impact, we again used the metrics described in section 5.1. We categorized the different impacts according to their type and probability by comparing the **weightedMean** value with the same limits as in the previous validation analysis and the p-Value with the same significance level as in the previous analysis.

The resulting factor-quality aspect combinations can be seen in the Tables 5 and 6. However we only included impacts which were not already considered in the quality model (and therefore the validation step) and from those only impacts showing a positive or negative trend.

The impacts are grouped by the factors as well as the high-level quality aspects of the quality aspects impacted by factors. Therefore, it can be seen that for several factors, impacts on quality aspects of the same high-level quality aspect have been rated. This shows a potential difficulty of using the quality aspects from the ISO 25000 standard: Because the quality aspects which we used for the survey are strongly related based on their common high-level quality aspect, it is more difficult to clearly distinguish between them in terms of how they are affected by software characteristics. For example, the factor *Managed infrastructure* in Table 6 has been rated as positively impacting all quality aspects grouped under the *Reliability* high-level aspect. In a hierarchical quality model, however, it would be ideal if there is only one path from a factor to a single top-level quality aspect. For the quality model this case could thus mean that either the quality aspects as considered in the survey should be used as the top-level aspects, excluding the high-level aspects or that the factor *Managed infrastructure* might need to be separated into separate other factors which then exclusively impact the different quality aspects. The survey results therefore show aspects which need to reconsidered in the quality model and can guide the future work on it to develop a conceptually sound quality model.

Table 5: Factors with new impacts (potentially) considerable for the quality model

Factor	Quality Aspect	High-level Aspect	- -	-	0	+	++	Impact	p-Value
Access control management consistency	Authenticity	Security	0	0	6	2	2	(+)	0.2302288
	Confidentiality		0	0	4	2	4	(+)	0.1548783
Account separation	Confidentiality	Security	0	0	4	2	1	(+)	0.5498971
	Integrity		0	0	2	1	4	(+)	0.1099966
Addressing abstraction	Modifiability	Maintainability	0	0	4	1	1	(+)	0.6399177
	Simplicity		0	0	3	3	0	(+)	0.1322016
	Replaceability	Portability	0	0	3	2	1	(+)	0.6399177
API-based communication	Modularity	Maintainability	0	1	2	0	7	+	0.0022373
	Reusability		0	0	4	1	5	+	0.0480364
	Simplicity		0	1	5	2	2	(+)	0.6684242
	Replaceability	Portability	0	0	7	0	3	+	0.0376729
Asynchronous communication	Elasticity	Performance efficiency	0	0	5	0	2	(+)	0.1430041
	Resource utilization		0	0	3	2	2	(+)	0.5498971
	Fault tolerance	Reliability	0	0	4	2	1	(+)	0.5498971
Automated infrastructure	Modularity	Maintainability	0	0	13	2	5	+	0.0031067
	Reusability		0	0	11	2	7	+	0.0028087
	Testability		0	2	9	3	6	+	0.0996570
	Capability	Performance efficiency	0	0	14	1	5	+	0.0007953
	Elasticity		0	0	11	2	7	+	0.0028087
	Resource utilization		0	0	13	2	5	+	0.0031067
	Time-behaviour		0	0	13	2	5	+	0.0031067
	Installability	Portability	0	1	8	2	9	+	0.0050431
	Availability	Reliability	0	0	7	3	10	+	0.0005842
	Fault tolerance		0	0	11	1	8	+	0.0005992
Automated restarts	Availability	Reliability	0	1	0	3	8	++	0.0000560
	Fault tolerance		0	0	3	3	6	+	0.0137452
Backing service decentralization	Elasticity	Performance efficiency	0	0	4	1	1	(+)	0.6399177
	Availability	Reliability	0	0	3	1	2	(+)	0.6399177
	Fault tolerance		0	0	3	0	3	(+)	0.1322016

Table 6: Factors with new impacts (potentially) considerable for the quality model continued

Factor	Quality Aspect	High-level Aspect	- -	-	0	+	++	Impact	p-Value
Built-in autoscaling	Capability	Performance efficiency	0	0	7	3	7	+	0.0095213
	Time-behaviour		0	0	8	4	5	+	0.0291854
	Availability	Reliability	0	0	8	5	4	+	0.0291854
	Fault tolerance		0	0	8	5	4	+	0.0291854
Circuit breaked communication	Recoverability	Reliability	0	0	5	1	2	(+)	0.3182442
Cloud vendor abstraction	Interoperability	Compatibility	0	0	6	3	2	(+)	0.1908058
	Reusability	Maintainability	0	0	6	1	4	+	0.0804314
	Installability	Portability	0	1	3	3	4	(+)	0.2344083
	Replaceability		0	0	6	1	4	+	0.0804314
Command Query Responsibility Segregation	Simplicity	Maintainability	0	4	3	0	0	-	0.0559842
	Elasticity	Performance efficiency	0	0	4	2	1	(+)	0.5498971
	Availability	Reliability	0	0	4	1	2	(+)	0.5498971
	Fault tolerance		0	0	4	2	1	(+)	0.5498971
Consistent centralized logging	Testability	Maintainability	0	0	4	0	2	(+)	0.3158436
	Accountability	Security	0	0	3	2	1	(+)	0.6399177
Contract-based links	Adaptability	Portability	0	0	3	0	2	(+)	0.3518519
Dynamic scheduling	Testability	Maintainability	1	1	4	0	0	(-)	0.6399177
	Capability	Performance efficiency	0	0	1	0	5	++	0.0026578
	Elasticity		0	0	3	0	3	(+)	0.1322016
	Time-behaviour		0	0	4	0	2	(+)	0.3158436
	Adaptability	Portability	0	0	4	1	1	(+)	0.6399177
Health and readiness Checks	Testability	Maintainability	0	0	6	1	2	(+)	0.2329247
	Availability	Reliability	0	0	3	1	5	+	0.0414333
	Fault tolerance		0	0	3	4	2	(+)	0.1325017
Horizontal data replication	Elasticity	Performance efficiency	0	0	5	0	3	(+)	0.1054955
	Availability	Reliability	0	0	2	3	3	(+)	0.1455047
	Fault tolerance		0	0	4	2	2	(+)	0.4862826
Infrastructure abstraction	Interoperability	Compatibility	0	0	5	2	1	(+)	0.3182442
	Simplicity	Maintainability	0	0	4	3	1	(+)	0.2798354
	Installability	Portability	0	0	5	1	2	(+)	0.3182442
	Replaceability		0	0	4	2	2	(+)	0.4862826
Limited data scope	Analyzability	Maintainability	0	0	2	3	0	(+)	0.1923868
	Testability		0	0	3	1	1	(+)	0.8456790
Limited request trace scope	Analyzability	Maintainability	0	1	2	1	2	(+)	0.8868313
	Time-behaviour	Performance efficiency	0	0	3	3	0	(+)	0.1322016
Managed backing services	Availability	Reliability	0	1	4	2	2	(+)	0.8023548
Managed infrastructure	Co-existence	Compatibility	0	1	8	2	3	(+)	0.2434166
	Elasticity	Performance efficiency	0	0	7	4	3	+	0.0765035
	Installability	Portability	0	1	8	2	3	(+)	0.2434166
	Availability	Reliability	0	0	4	3	7	+	0.0073683
	Fault tolerance		0	0	9	2	3	+	0.0498688
	Maturity		0	0	8	3	3	+	0.0765035
	Recoverability		0	0	8	3	3	+	0.0765035
Mediated communication	Time-behaviour	Performance efficiency	0	3	2	0	0	(-)	0.1923868
	Replaceability	Portability	0	0	3	1	1	(+)	0.8456790
Mostly stateless services	Analyzability	Maintainability	0	1	4	2	2	(+)	0.8023548
	Reusability		0	0	4	3	2	(+)	0.3169439
	Testability		0	0	2	0	7	++	0.0006255
	Resource utilization	Performance efficiency	0	0	4	2	3	(+)	0.3169439
	Recoverability	Reliability	0	0	6	1	2	(+)	0.2329247
Persistent communication	Elasticity	Performance efficiency	0	0	3	1	1	(+)	0.8456790
	Availability	Reliability	0	0	2	2	1	(+)	0.7222222
	Fault tolerance		0	0	1	2	2	(+)	0.2695473
Physical data distribution	Resource utilization	Performance efficiency	1	4	2	0	0	(-)	0.1099966
	Fault tolerance	Reliability	0	0	1	1	5	++	0.0095165
	Recoverability		0	1	3	1	2	(+)	0.9039780
Physical service distribution	Fault tolerance	Reliability	0	0	2	3	5	+	0.0232529
	Maturity		0	0	6	1	3	(+)	0.1655474
	Recoverability		0	2	3	2	3	(+)	0.6204132
Retries for safe invocations	Availability	Reliability	0	0	2	1	2	(+)	0.7222222
	Recoverability		0	0	3	0	2	(+)	0.3518519
Secrets stored in specialized services	Accountability	Security	0	0	4	0	2	(+)	0.3158436
	Authenticity		0	0	3	1	2	(+)	0.6399177
	Integrity		0	0	1	2	3	(+)	0.1184842
Service replication	Availability	Reliability	0	0	1	4	9	++	0.0000276
	Fault tolerance		0	0	5	3	6	+	0.0238423
Usage of existing solutions for non-core capabilities	Interoperability	Compatibility	0	0	4	2	1	(+)	0.5498971
	Simplicity	Maintainability	0	0	3	4	0	+	0.0559842
Use infrastructure as code	Reusability	Maintainability	0	0	9	2	5	+	0.0241678
	Testability		0	0	9	3	4	+	0.0404223
	Adaptability	Portability	0	0	8	6	2	+	0.0163453
	Replaceability		0	0	10	3	3	+	0.0285723
	Availability	Reliability	0	0	9	3	4	+	0.0404223
	Fault tolerance		0	0	12	0	4	+	0.0009318
Vertical data replication	Analyzability	Maintainability	0	3	2	0	0	(-)	0.1923868
	Availability	Reliability	0	0	3	0	2	(+)	0.3518519
	Fault tolerance		0	0	3	1	1	(+)	0.8456790

6 Conclusion

The results of this survey will be used to continue the work on the quality model for cloud-native application software architectures. While parts of the quality model could be validated by the results of the survey, others need reconsideration. Our plan for future work is to apply the quality model to software architectures functioning as use cases based on which we want to further validate the elements of the quality model. The insights from this survey can guide this work by showing which factors to focus on or for example which hypotheses to formulate for experiments with such software architectures.

Although we received a lower number of submissions than we had initially hoped for, the general approach for the survey showed feasible. But the lower number of submission also means that the interpretation of the results needs to be done with caution which we did by considering the significance of results and marking results as uncertain if applicable.