Empirical Evaluation of a Process to Increase Consensus in Group Architectural Decision Making

Dan Tofan[a], Matthias Galster[b], Ioanna Lytra[c], Paris Avgeriou[a], Uwe Zdun[c], Mark-Anthony Fouche[d], Remco de Boer[e], Fritz Solms[d]

[a] Department of Mathematics and Computing Science, University of Groningen, Netherlands
[b] Department of Computer Science and Software Engineering, University of Canterbury, New Zealand
[c] Faculty of Computer Science, University of Vienna, Austria
[d] Department of Computer Science, University of Pretoria, South Africa
[e] ArchiXL, XL&Knowledge, Netherlands

Context: Many software architectural decisions are group decisions rather than decisions made by individuals. Consensus in a group of decision makers increases the acceptance of a decision among decision makers and their confidence in that decision. Furthermore, going through the process of reaching consensus means that decision makers understand better the decision (including the decision topic, decision options, rationales, and potential outcomes). Little guidance exists on how to increase consensus in group architectural decision making.

Objective: We evaluate how a newly proposed process (named GADGET) helps architects increase consensus when making group architectural decisions. Specifically, we investigate how well GADGET increases consensus in group architectural decision making, by understanding its practical applicability, and by comparing GADGET against group architectural decision making without using any prescribed approach.

Method: We conducted two empirical studies. First, we conducted an exploratory case study to understand the practical applicability of GADGET in industry. We investigated whether there is a need to increase consensus, the effort and benefits of GADGET, and potential improvements for GADGET. Second, we conducted an experiment with 113 students from three universities to compare GADGET against group architectural decision making without using any prescribed approach.

Results: GADGET helps decision makers increase their consensus, captures knowledge on architectural decisions, clarifies the different points of view of different decision makers on the decision, and increases the focus of the group discussions about a decision. From the experiment, we obtained causal evidence that GADGET increases consensus better than group architectural decision making without using any prescribed approach.

Conclusions: There is a need to increase consensus in group architectural decisions. GADGET helps inexperienced architects increase consensus in group architectural decision making, and provides additional benefits, such as capturing rationale of decisions. Future work is needed to understand and improve other aspects of group architectural decision making.

Keywords: software architecture; group architecture decisions; decision making

Contents

1. Introduction

Designing the software architecture for a system involves making many architectural decisions [1]. Typical examples of architectural decisions are choosing development platforms (e.g. Java EE, .NET), database systems (e.g. Oracle, MongoDB), frameworks (e.g. object-relational mapping frameworks), or architectural patterns. Architectural decisions involve trade-offs (e.g. one decision may increase usability, but reduce security), are hard to make due to necessary trade-offs, and expensive to change (e.g. changing from the Java EE to the .NET platform) [2].

1.1. Problem Description

In practice, most software architecture decisions are made in groups (and involve different stakeholders), rather than by individual architects [3, 4]. Unfortunately, little is known about group architectural decisions, and how to improve group architectural decision making. In a recent mapping study on architectural decisions [5], we found that not much research exists on group architectural decisions. Group architectural decision making entails substantial challenges, such as communication among decision makers and the need to reach a certain degree of consensus between decision makers and other stakeholders [6].

Increasing consensus among decision makers is a critical factor of group decision making. On the one hand, low consensus in early architectural decisions may lead to misunderstandings within the group of decision makers [6]. Such misunderstandings may cause problems. For example if a stakeholder feels that her point of view about a decision was not taken seriously, that stakeholder might not accept the final software system. On the other hand, benefits of consensus include higher acceptance and better understanding of the architectural decision by all involved stakeholders. Furthermore, consensus increases confidence in the correctness of the architectural decision [6]. Therefore, consensus needs to be addressed explicitly as part of group architectural decision making. However, as mentioned before, no approach from software architecture literature targets explicitly the increase of consensus in group architectural decision making.

Regarding the scope of this paper, we focus on *consensus* (i.e. 'we have some general agreement and we understand each other's perspectives') instead of *unanimity* (i.e. 'all of us have the same perspectives'). Furthermore, in our work, *consensus* has two main components: *general agreement* and *mutual understanding* among stakeholders involved in making a decision [7]. Therefore, in this paper, we focus on how to increase *general agreement* and *mutual understanding* amongst inexperienced architects.

1.2. Contributions

In this paper, we propose and evaluate GADGET (*Group Architectural Decisions with repertory Grid Technique*), which is a group decision making process for helping architectural decision makers (e.g. architects and other stakeholders who have a decision-making role) increase consensus about their decisions. GADGET aims at helping groups that are recently formed and which do not have common procedures and processes in place, and therefore may benefit from a standardized way of interaction. The process offers guidance for increasing consensus incrementally, making explicit the knowledge of the decision makers, and helping them structure their group interactions.

This paper contributes with the GADGET process and empirical evidence of how GADGET increases consensus in group architectural decision making. The validation has two parts:

- a case study with seven students and thirteen practitioners
- an experiment with 113 students to answer research questions that emerged from the case study

1.3. Paper Structure

Figure 1 shows an overview of the research presented in this paper. Phase 1 consists of previous work that motivated the research in this paper. While investigating how architectural decisions are made in practice [3], we found out that most architectural decisions are group decisions, similar to [4]. Furthermore, one of the outcomes of a systematic mapping study on architectural decisions literature was that there is little research on group architectural decisions [5]. These outcomes motivated us to propose an approach to improve consensus in group architectural decisions in phase 2. The resulting approach (GADGET) is presented in Section 2. In phase 3, we conduct a case study to collect initial evidence on the practical applicability of GADGET. As reported in Section 3, case study results also suggested that no systematic approach is used in practice for reaching consensus (we term any ad-hoc approach used as ADHOC). In phase 4, we conduct an experiment to compare GADGET vs. ADHOC, and obtain causal evidence on how GADGET increases consensus compared to ADHOC (see Section 4). Furthermore, we discuss validity threats of the case study and the experiment in Section 5, and related work in Section 6. Finally, Section 7 presents conclusions and future work.

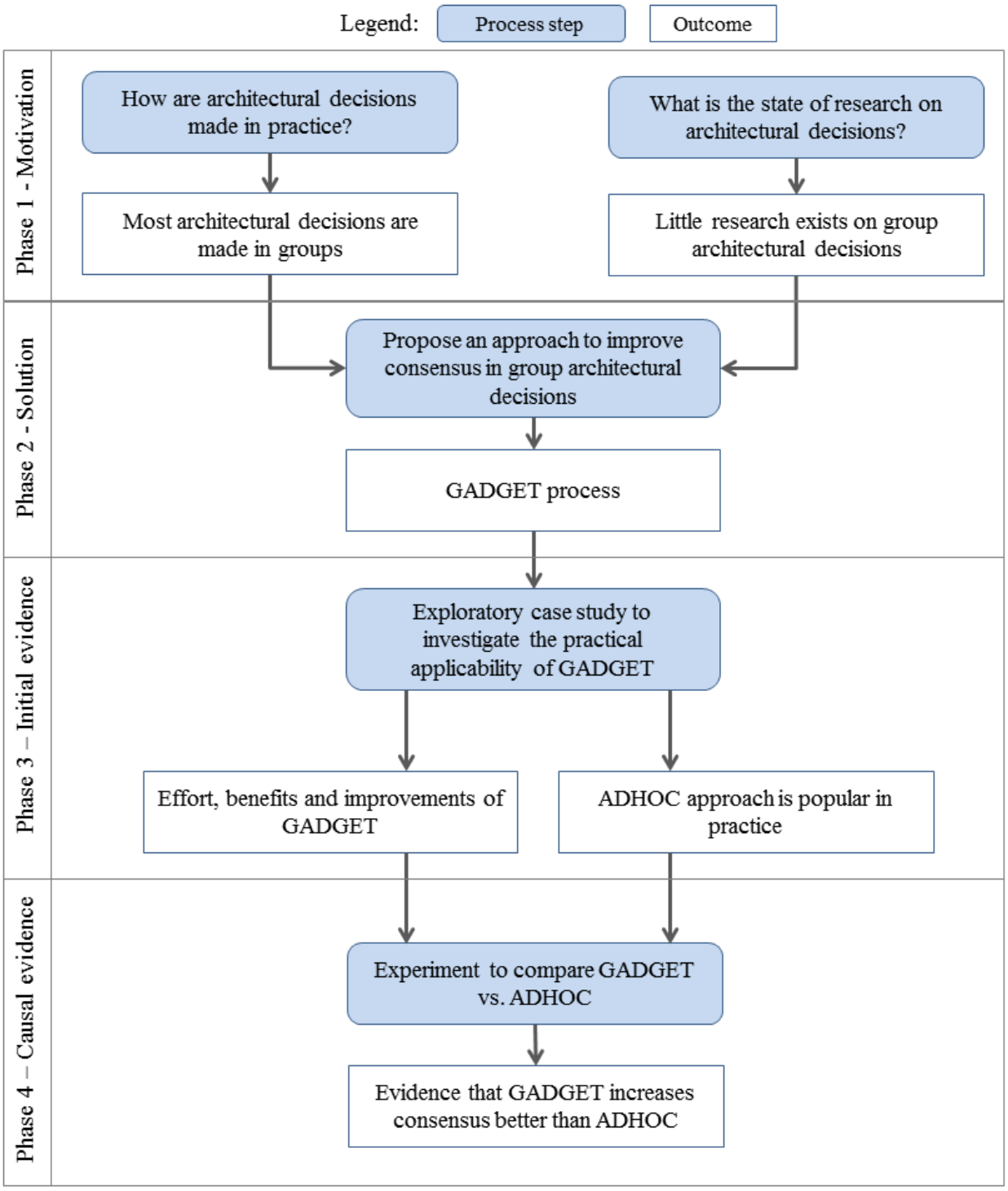

Figure 1. Overview of the research presented in this paper. Phase 1 is reported in previous work. Phases 2 to 4 are reported in this paper.

2. The GADGET Process

To describe the GADGET process, we present its roots (section 2.1) and concrete steps (section 2.2).

2.1. GADGET Roots

GADGET extends our previous work on making and capturing architectural decisions with the **Repertory Grid technique** [8-10], with the idea of group evaluations and feedback from the **Delphi technique** [11].

The **Repertory Grid technique** [12] is a structured technique for knowledge acquisition [13]. In our previous work, we adapted the Repertory Grid technique for architectural knowledge acquisition [8-10], and presented evidence about advantages and disadvantages of using the Repertory Grid technique for making and capturing architectural decisions. For example, the Repertory Grid technique provides systematic architectural decision making support, concise documentation, and reduces architectural knowledge vaporization. The Repertory Grid technique adapted for architectural knowledge acquisition consists of the following steps:

1. Indicate a decision topic.
2. Indicate decision alternatives.
3. Get concerns that characterize decision alternatives (e.g. through repeated comparisons among alternatives); the output of steps 2 and 3 is a matrix (or grid) with concerns as rows and alternatives as columns.
4. Prioritize concerns (e.g. using the *hundred-dollar approach*: assign a priority to each concern from 0 to 100, so that the sum of priorities is 100 [8]).
5. Rate alternatives against each concern using a one-to-five Likert scale, which fills the matrix of alternatives and concerns with ratings.
6. Analyze the matrix of alternatives, concerns, and ratings to indicate the most preferable alternative (for detailed examples, see [8-10, 12]).

The **Delphi technique** is a '*method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem*' [11]. In Delphi, participants answer questions on a complex problem in several iterations, receive a summary of answers from all other participants, and are given the opportunity to revise their answers for the next iteration. After several iterations, the answers converge and determine the solution to the complex problem.

In addition to Delphi, we also considered other techniques to be included in GADGET, namely *brainstorming* [14] and *nominal group* [15]. However, we preferred Delphi for the following reasons. *Brainstorming* is strong at generating new, creative ideas, while performing evaluations. Since our goal was to increase consensus, these characteristics were not high priority for GADGET. The *nominal group* technique has similar steps as Delphi, but the evaluation step is anonymous. We preferred that GADGET has an open evaluation step, so that participants can communicate and understand faster each other's perspectives.

2.2. GADGET Steps

Figure 2 shows the five steps of GADGET. The input of GADGET is an architectural decision topic (e.g. choice of database, architectural patterns, JavaScript framework, or platform technologies). The decision topic can be proposed from inside the group (e.g. one or more decision makers), or from outside the group (e.g. a stakeholder). Identifying decision makers can be supported by using established architectural frameworks such as TOGAF, since TOGAF offers explicit steps for stakeholder management, such as the identification of decision makers. Furthermore, in our previous study [3] we found out that typical size of a group of architectural decision makers in the industry is three. There are also stakeholders that influence the decision, but who are not directly involved in making this decision. Our previous study found out that typically there are three such stakeholders for a decision.

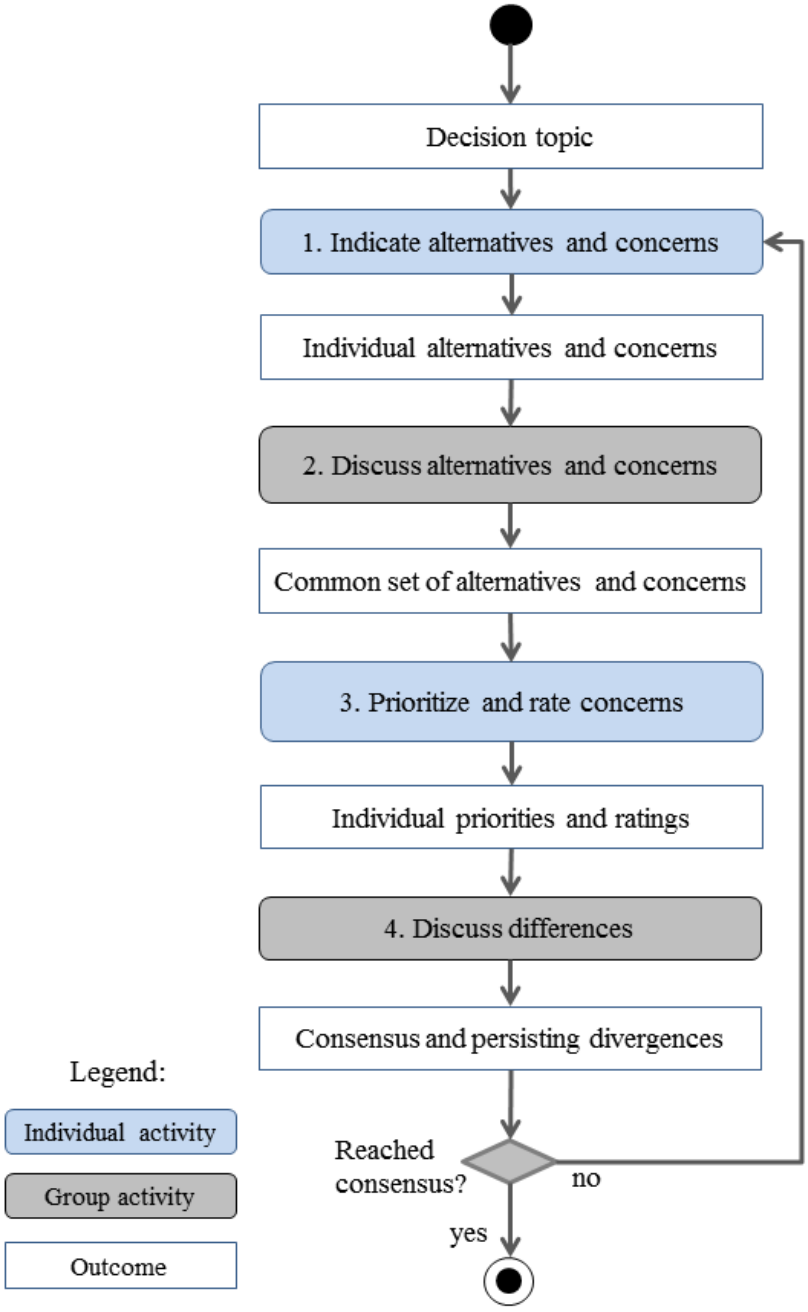

Figure 2. GADGET process steps and outcomes.

Each step consists of the following.

1. *Indicate alternatives and concerns:* Decision-makers indicate individually their alternatives and concerns for the decision topic. To support this step, decision-makers can reuse relevant alternatives and concerns that were identified previously using an architectural framework (e.g. concerns indicate the why in Zachman's framework). In addition, decision-makers can reuse relevant concerns that were captured using ISO 42010 compliant viewpoints which might be used in the organization. Decision-makers can indicate what alternatives or concerns to remove from previous iterations (see Step 5). The rationale for this step is to ensure that any potentially

relevant alternative and concern is considered in the decision making process. The output of this step is a set of alternatives and concerns from each decision-maker. For example, for making a decision about the JavaScript framework, one of the decision-makers indicates three alternatives (e.g. Angular, Ember, and Backbone), and four concerns (e.g. testability, performance, learning curve, and existing skillsets).

2. *Discuss alternatives and concerns:* Decision-makers have a group discussion on the alternatives and concerns, with the purpose of consolidating them in a common set of agreed alternatives and concerns. The rationale for this step is to clarify and potentially add or remove alternatives and concerns that are included in the decision making process. For example, more alternatives can be added and some concerns can be clarified (e.g. what is minimum acceptable performance of a JavaScript framework).
3. *Prioritize concerns and rate concerns against alternatives*: Decision-makers individually prioritize the common set of concerns using the hundred-dollar approach (i.e. assign a priority to each concern from 0 to 100, so that the sum of priorities is 100). Even though other prioritization techniques could be used in this step, our previous research indicates that the hundred-dollar approach is most suitable in this context [8]. In addition, decision-makers individually rate each of the common alternatives against every concern, using a five-level Likert scale, with values ranging from '1-strongly disagree' to '5-strongly agree'. Decision-makers may use supplementary values such as 'not applicable' and 'don't know'. The rationale for this step is to ensure that alternatives and the importance of concerns are considered when making the decision (some stakeholders may consider alternatives and concerns more or less important than others). The output of this step is the set of ratings and priorities from each decision-maker.
4. *Discuss differences*: Based on the ratings and priorities of concerns from Step 3 metrics are calculated for priorities and ratings. For ease of interpretation and usability of GADGET, only four metrics are used for the ratings and priorities indicated by participants in Step 3:
    a) average of ratings of alternatives based on concerns
    b) average priorities of concerns
    c) range of ratings of alternatives based on concerns (i.e. difference between highest and smallest ratings)
    d) range of priorities (i.e. difference between highest and smallest priorities).

    These metrics help decision-makers understand how their own perspectives compare to the perspectives of the other decision-makers. This generates a 'soft' pressure towards convergence. If differences in ranges are small enough, then there is an acceptable degree of consensus among decision makers. Otherwise, the decision-makers with highest differences present their rationales to stimulate focused discussions about the differences in perceptions. During these discussions, participants are either willing to modify their priorities and ratings, or they 'agree to disagree'. The expected output of this step is increased consensus, and/or explicit list of persisting divergences, which, if too big (i.e. range bigger than 2 for ratings, range bigger than 20 for priorities), suggest the need for an additional iteration. The discussions in this step may modify the perspectives of the decision-makers, which could lead to new alternatives and concerns, or different priorities and ratings of concerns.
5. *Iterate from Step 1*: Consensus is visible when none of the decision-makers is willing to modify his or her earlier input (i.e. alternatives, concerns, ratings or priorities). If consensus is reached, then GADGET finishes. Otherwise, iterating from Step 1 is needed to allow decision-makers to

modify their earlier input. Typically, as discussed further in section 3.3.1, one or two iterations should be enough.

GADGET allows decision makers to iterate as necessary, since there is no constraint on the minimum time to be spent in any of the steps. However, the first iteration provides most alternatives and concerns, while subsequent iterations adjust the alternatives and concerns. For example, if – while working at step 3 - some new concerns appear, the decision makers can move through steps 4 and 5, towards step 1, so that the new concerns can be included in the process.

3. GADGET Case Study

We conducted an exploratory case study to explore the practical applicability of GADGET for the purpose of evaluating GADGET with respect to its impact on consensus among decision makers from the viewpoint of a group of decision makers, in the context of architectural decisions. Case studies are very well suited for exploratory research questions [16], since case studies offer flexibility to study a phenomenon (e.g. group decision making) in its real-world context. Case studies rely on observations to form tentative hypotheses and confirmatory research questions, which can be further investigated in subsequent studies. Next, we report the case study using the guidelines from [17].

3.1. Case Study Design

We defined the following three case study research questions:

RQ1. Is there a practical need for increasing consensus in group architectural decision making?

As discussed in Section 1, there is very little work on consensus in group architectural decision making. Therefore, before investing efforts into developing approaches for increasing consensus, we investigated whether such approaches are needed. If there is a practical need to increase consensus in group architectural decision making, then an approach such as GADGET may satisfy this need.

RQ2. What are the effort and benefits offered by GADGET?

The rationale for RQ2 is that practitioners are usually interested in the actual benefits of a new approach (or GADGET in our case) and effort (i.e. time) involved in using it. If an approach has low benefits and requires high effort, then practitioners are unlikely to use such approach. Researchers need to pay attention to effort and benefits of a new approach, to avoid proposing approaches that practitioners are unlikely to use.

RQ3. What are potential improvements to GADGET?

The rationale for RQ3 is that we wanted to improve GADGET to ensure it satisfies the needs of its potential users. In particular, we were interested in getting feedback on GADGET drawbacks, so that we could use such feedback to improve GADGET.

To recruit participants, we invited practitioners from the local community of architects in Groningen. In addition, to obtain more data, we invited graduate students with practical experience, who took the software architecture course given by one of the authors at the University of Groningen.

The case study used groups of three to four participants. Each case study session for each group consisted of three steps:

1. Participants received an overview of the case study session in which they participated, the GADGET process, and an example to illustrate the GADGET process.
2. Participants used GADGET on an architectural decision topic they had been involved with in their recent activity. Participants entered alternatives, concerns, and ratings into a shared online spreadsheet that we had prepared in advance.
3. Participants provided feedback on GADGET in a group discussion. To focus the group discussions, we prepared the set of discussion items in Table 1. We used the discussion items for RQ1 only during the sessions with practitioners, and skipped these questions in the sessions with students, since we were interested in identifying the real-world need for GADGET, as indicated by practitioners.

Table 1. Discussion items for obtaining feedback from participants.

ID	Discussion Item	Research Question
1	Do conflicting perspectives occur in group architectural decision making?	RQ1
2	What is the impact of conflicting perspectives in group architectural decision making?	RQ1
3	What approaches have you used so far in consensus building? (If any)	RQ1
4	What did you like/dislike about the proposed process?	RQ2, RQ3
5	Would you use this process in your practice?	RQ2, RQ3
6	Did you change your opinion about alternatives? Why (not)?	RQ2
7	How did the process help?	RQ2
8	How can the process be improved?	RQ3
9	In which situations would you apply the process?	RQ2, RQ3

We made audio recordings of the sessions, with the prior permission of the participants. For analyzing the feedback from participants, two researchers independently performed content analysis on the transcriptions of the recordings and observer's notes, to identify codes corresponding to sentences, phrases or paragraphs, as recommended by [18]. Then, in case of differences in interpretation, researchers discussed and resolved the differences. We grouped the codes from the content analysis to answer the three research questions: on need for consensus in group architectural decision making (RQ1), effort/benefits of GADGET (RQ2), and possible improvement for GADGET (RQ3). The content analysis results are available online at [19].

3.2. Results

3.2.1 Case Study Participants and Execution

Table 2 summarizes the groups of students and practitioners that participated in the case study, and the decision topics that were addressed during the sessions. Years of experience refer to practical experience in software engineering. Groups S1, S2 and P2 opted to use topics that we prepared in advance, and all other groups used decision topics from their recent activity.

Table 2. Groups of decision makers that participated in the case study.

Group id	Group size	Group type	Average years of experience	Decision topic	Number of GADGET iterations
S1	4	Students with industry experience	4.62	Enterprise Resource Planning system	1
S2	4		4.50	JavaScript framework	1
P1	3	Practitioners	9	Buy or build critical component	2
P2	3		9	Communication system	1
P3	4		3.66	Operating system	2
P4	3		6	Programming language	2

As an example on the execution of the sessions, participants in S1 indicated concerns such as '*low price*', '*high security*', '*high level of customer service*', and '*low learning curve*'. For S1, step two of GADGET resulted in seven alternatives (e.g. SAP Business One, Microsoft Dynamics, NetSuite) and eleven concerns for the first session. In Step three of GADGET, members of S1 prioritized concerns using the hundred-dollar approach. In addition, participants rated each alternative against each concern on a one-to-five scale, indicating how well an alternative satisfies a concern. Participants were familiar with some of the consolidated alternatives, but needed more time to learn about the others. During the session, they searched for information on the alternatives on the internet, and used the results for the ratings. In Step four of GADGET, members of S1 discussed the differences between the values they assigned, starting with the ratings that had the highest ranges. Participants discussed 14 ratings during the only iteration of the process. Participants reached consensus for eleven ratings.

Finally, we spent 20 minutes to obtain feedback on GADGET through a group discussion. We encouraged participants to provide feedback on their experiences, using the questions in Table 1.

3.2.2 Analysis Results

Next, we present the results of the content analysis, for the three categories corresponding to RQ1, RQ2, and RQ3.

RQ1 - Need for consensus in group architectural decision making

Regarding occurrences of conflicting perspectives (item 1 in Table 1), two architects indicated that conflicting perspectives related to a decision do not occur very often, and four architects indicated that they occur very often. Increasing the number of decision makers increases the number of conflicting perspectives, since decision makers have different priorities for concerns, and tradeoffs need to be found.

From the content analysis, we identified a positive and a negative impact of conflicting perspectives (item 2 in Table 1). On the one hand, participants indicated that conflicting perspectives is often time consuming (as one architect phrased it: '*long and often almost endless discussions*'). On the other hand, participants indicated that the outcome of the decision is better if there are conflicting perspectives, because it encourages decision makers to address concerns of more stakeholders.

Regarding approaches for increasing consensus (item 3 in Table 1), from the content analysis we learnt architects lack structured approaches. Instead, architects use unstructured group discussions to increase consensus.

Overall, there is a need for increasing consensus in group architectural decision making in a systematic way, since 1) conflicting perspectives occur in practice, 2) conflicting perspectives help make better decisions, and 3) architects lack structured approaches for increasing consensus.

RQ2 - Effort and benefits

Regarding effort, we observed that GADGET requires one to three hours per group, for a decision topic with three to six alternatives. Regarding benefits, the main benefit that emerged from the content analysis was increasing consensus among decision makers on the architectural decision. This benefit was indicated by five participants. A participant in the first session expressed this: '*that's what I really liked about the process: not focusing on the decision making in the first place, but on agreeing on a viewpoint*.' Additionally, a participant stated: '*we learnt from it, you see other points of view, you also see your own gaps and misconceptions*'. The overall message from participants was that GADGET helped them increase consensus, by developing an increased shared understanding of each other's perspectives, as a result of discussing the differences between them in a structured manner.

Several other additional benefits emerged from the content analysis:

a. **Increased focus** of the group discussions (appearing three times in the content analysis). According to a participant, decision makers are '*less likely to run off-topic*'. Moreover, participants considered that the process offered a structured way of increasing consensus, with prioritization of items for discussion, allowing them to '*focus on stuff that is important.*'
b. **Rationale** – participants appreciated that GADGET helps them capture the rationale for the decision, in addition to making the decision. Specifically, GADGET provides the rationale through its metrics, and maps concerns to participants. Therefore, architects can see not only the outcome of the group decision, but also the intermediary steps that lead to the outcome.
c. **Reusability** – participants indicated that GADGET output (i.e. alternatives, concerns, and ratings) has high potential for reusability. For example, after making a group decision with GADGET, if a decision on the same topic needs to be made in the future, then alternatives, concerns, and ratings may be reused. In addition, some concerns may be reused across different decisions, especially across decisions that have strong dependencies (e.g. security-related concerns are reusable across most decisions for architecting a security-intensive system).
d. **Clarity of problem** – architects indicated that GADGET helped them clarify their point of view on the decision, by forcing architects to make explicit what matters to them in the decision.

RQ3 – Improvements

During the case study with the first group of participants, they indicated the need for increasing consensus on the priorities of concerns. Therefore, we updated GADGET to include prioritization of concerns (i.e. step three of GADGET), and we used the updated version of GADGET with the rest of the groups.

Here are the additional improvements suggested by participants throughout the sessions, and what we did about them:

- Participants suggested to optimize the time needed to use GADGET, by avoiding idle time in a face-to-face meeting, which happens when participants need different amounts of time to finish a step. For example, step three of GADGET (i.e. prioritize and rate concerns, see section 2) can

take place outside of a face-to-face meeting. Based on this suggestion, we removed time constraints (in section 2) on using GADGET in face-to-face meetings.

- Allow decision makers to eliminate less promising alternatives in later iterations. Based on this suggestion, we made explicit in the GADGET description (see step one in section 2) that decision makers can also indicate what alternatives and concerns to remove when iterating.
- Participants considered that spreadsheets lacked dedicated features, such as the ability to trace divergent perspectives among decision makers. One of the architects indicated that he '*wants to spend most of the time on discussions, instead of working with the tool*.' We used this feedback for developing dedicated, user-friendly tool support for GADGET [20].

3.3. Discussion

The exploratory case study offered us insights on GADGET. The increase in consensus from using GADGET was visible not only in the input from participants (e.g. ratings), but also in the feedback from participants. For example, a participant mentioned: '*I trust the knowledge my teammates have from their respective fields. After noting they are more informed than I am, I would gladly accept their vision of the alternative, and I would concede to their rating*.' Additionally, other participants mentioned that strong arguments from peers in their groups convinced them to adjust their ratings.

Overall, the benefits of GADGET include: increased focus of the discussions, captured rationale of the decisions, potential for reusability of captured knowledge on decisions, and time savings. Still, there is further room for improving GADGET: offering additional prioritization approaches for concerns and adding confidence levels to ratings. Also, tool support for GADGET needs to be user-friendly (i.e. low learning curve, and reducing the time required to learn and use GADGET).

3.3.1 Recommendations for Practitioners

From our experience with using GADGET, we recommend the following:

- Regarding threshold values for step four of GADGET (i.e. discuss differences), the recommended thresholds guideline values for differences are one for ratings and ten for priorities
- Regarding the number of iterations, two iterations for GADGET provide sufficient opportunities for decision-makers to reach consensus (i.e. general agreement on the decision, and mutual understanding of each other's perspectives)
- GADGET is particularly useful when the following conditions are met:
    - The topic of the architectural decision is important enough for a group decision.
    - The architectural decision has several promising alternatives, so that spending time to evaluate them systematically is worthy.
    - The decision makers have the maturity and openness to adopt and apply a systematic approach for their decision.

3.3.2 Implications for Research

Although there is a need for consensus in group architectural decision making, when making group architectural decisions, decision makers typically do not use any structured approach for increasing consensus. This means that decision makers use an 'as-is' or 'natural' approach which occurs when decision makers increase consensus without using any predefined approach. We call this approach ADHOC - the approach of increasing consensus in group architectural decisions without using any structured approach. Overall, the ADHOC approach seems to be popular in practice.

Exploratory case studies, such as the one we reported in this section, are useful for obtaining insights and generating hypotheses for further research [21]. This case study brought initial evidence that GADGET increases consensus. Moreover, this case study helped us generate research questions and hypotheses for comparing GADGET with ADHOC, which we report in Section 4. Validity threats are reported in sub-section 5.1.

4. GADGET Experiment

The exploratory case study offered insights and initial evidence into the need for increasing consensus in group architectural decisions, as well as the effort and benefits offered by GADGET. One of the insights was that, in practice, consensus is often increased without using any structured approach (i.e. ADHOC). Therefore, we conducted an experiment to compare GADGET (i.e. a new approach) with ADHOC (i.e. the existing frequently used approach). This comparison allows drawing conclusions whether GADGET improves the current state of practice. Next, we report the experiment using the guidelines from [22].

In this experiment, we used ADHOC (as motivated in the previous section) for the **control** groups, and GADGET for the **treatment** groups. By comparing GADGET with ADHOC, we could better understand if GADGET increases consensus, compared to ADHOC. This was a further research step compared to the exploratory case study in Section 3, in which we brought initial evidence that GADGET increases consensus, but we did not compare GADGET with another approach.

We chose to compare GADGET with ADHOC, instead of another process, for two reasons:

1. **Practical relevance.** Since ADHOC is popular in practice (as found in the case study in section 3), the comparison with ADHOC helps practitioners understand what they can expect from adopting GADGET.
2. **Lack of a reference process.** As we found out in previous research [5], there is no reference process in the literature for group architectural decision making to use as a baseline for comparison.

4.2. Research Goal and Questions

The goal of the experiment was to *compare* GADGET with ADHOC *for the purpose of* understanding them *with respect to* their impact on consensus among decision makers *from the viewpoint of* decision makers, *in the context of* group decision making for software architecture.

From our research goal, we derive the following two research questions.

RQ1. Compared to ADHOC, what is the impact of GADGET on increasing consensus among group architectural decision makers?

Rationale: This research question aims at offering evidence on how GADGET compares against ADHOC at increasing consensus among decision makers. In the case study in Section 3, we found that GADGET has the potential to increase consensus. However, an ad-hoc and unsystematic approach (i.e., ADHOC) can also help achieve consensus. If ADHOC has the same effect as GADGET, then it makes little sense for decision makers to use GADGET, since ADHOC has less overhead than GADGET.

RQ2. How do perceptions on GADGET and ADHOC differ among decision makers?

Rationale: The perception of an approach influences strongly the actual intention to use that approach [23]. A positive perception of an approach likely leads to a higher intention to use the approach, which, in turn, results in actual usage of the approach. For example, if some architects perceive that GADGET brings benefits such as capturing rationale and correctness, without significant extra effort, then these architects are likely to use GADGET in their future activity. Therefore, understanding the perceptions on GADGET helps us understand the actual potential future usage of GADGET.

We present the metrics for answering RQ1 and RQ2 in sub-sections 4.5 and 4.6.

4.3. Participants

There are certain constraints when selecting participants for experiments. If the experiment has insufficient participants, then it is difficult to obtain relevant results. Also, if the sample is not representative enough, then the results of the experiment can be debated. However, a trade-off needs to be made between the number of participants and their representativeness. Kubickova and Ro [24] indicate that students are used as research subjects in an increasingly large number of scientific studies in various disciplines (e.g. in 80% of consumer research studies), despite continuous debates which have been going on for several decades on the scientific value of using students as research subjects [24].

Such debates also exist in software engineering research. A study on freshmen, graduate students, and industry people found no conclusive results on differences between these types of participants [25]. Another study suggests that students "may work well" as subjects for software engineering studies [26].

We chose to use a high number of participants with a good-enough representativeness for inexperienced software architects, who can benefit much from a structured approach for increasing consensus in their group architectural decisions. Furthermore, since we aim at establishing causal relationships, using students is preferable than using practitioners: students help reducing variations and thus confounding factors, so they help increase the internal validity of the study.

Participants in our experiment were graduate and undergraduate software engineering students, who took a Software Architecture course, in which they were presented the concept of architectural decisions. We conducted the experiment with students from three universities: University of Groningen in Netherlands, University of Vienna in Austria, and University of Pretoria in South Africa. To eliminate potential confounding factors such as expertise (graduate/undergraduate, practical experience), and background (different universities), each experimental session followed the same experimental process (see Section 4.4.3), in which we randomized and balanced the distribution of the students across the control and treatment groups. Section 4.7 describes the background of participants, including their practical experience, and their balanced distribution across the control and treatment groups.

For validity and ethical purposes, we ensured that students had commitment for the study, and that the study contributed to participants' education, as recommended by [27]. To this end, we followed a checklist for integrating student empirical studies with our research and teaching goals [28, 29]. Below we present several items from Carver's checklist for our study.

1. **Ensure adequate integration of the study into the course topics**. The course lectures discussed architectural decisions. In the introduction of the experiment, we explained to students how the session helps them improve their architectural decision making skills.

2. **Write up a protocol and have it reviewed.** We prepared the set of steps to follow and discussed them with two other researchers not involved in the study. Furthermore, the ethics committee from the University of Pretoria reviewed the protocol and approved it, with minor modifications. Reviews of ethics committees from the other universities were not required.
3. **Obtain participants' permission for their participation in the study**. We told students about the experiment at least one week in advance. We also told students that the session covers advanced topics in software architecture, and that participation is voluntary, with no influence on their grades. By showing up for the session, students consented to participate. In addition, students from the University of Pretoria signed a consent form to indicate explicitly their consent.
4. **Build or update a lab package**. We developed the lab package at the University of Groningen. Later on, researchers from University of Vienna and University of Pretoria used the same lab package to replicate the experiment.

4.4. Experimental Materials and Process

The lab package (available online at [19]) included the experimental case and other experimental materials. In this section, we describe the experimental case (in 4.4.1), other experimental materials (in 4.4.2), and the experimental process (in 4.4.3).

4.4.1 Experimental Case

We used a predefined experimental case. The case contained the architectural decision and contextual information about the decision. The case was based on an architectural decision that we elicited from interviewing architects in the industry [30]. The case had a five-page description with all the details that students needed in order to engage in the group decision making: a description of the organization for which the decision was made, the decision topic, concerns, alternatives, and decision maker roles. There were three decision maker roles: Department Manager, IT Architect, and Business Analyst. Each student took one of the roles during the experiment.

In summary, the case is about three decision makers from an organization that need to make an architectural decision about its current newsletter system. The case describes four candidate alternatives:

A. Use a Software as a Service solution
B. Develop a new custom system
C. Customize an existing open source system
D. Enhance the current system

The case describes six concerns which are applicable to the candidate alternatives:

1. Delivery time
2. Training time
3. Analytics
4. Cost
5. Scalability
6. Security

As specified in section 4.2, the goal of the experiment was to understand the impact of decision making approaches on consensus among decision makers. In a group decision, if the consensus is trivial to reach

(i.e. there is a clear superior alternative that satisfies all decision makers), then the impact of the decision making approach is very difficult to understand. On the contrary, if reaching consensus is not trivial, then the impact of the group decision making approaches can be understood. Therefore, to reach our experimental goal, we had to design a non-trivial situation for reaching consensus.

To ensure that reaching consensus was not trivial, in the case we specified that each role had different priorities for the concerns, and each alternative satisfied the concerns to various extents. Table 3 summarizes how each alternative satisfied each concern, and the most important concerns for each of the three roles.

Table 3. Summary of the experimental case in terms of how each alternative satisfies each concern, and the most important concerns for the three decision makers. The underlined values indicate the alternatives that satisfy best the most important concerns for each of the decision makers.

Alternative	**Delivery time** (most important for the Business Analyst)	**Training time** (important for the Business Analyst)	**Analytics** (most important for the Department Manager)	**Cost** (important for the Department Manager)	**Security** (most important for the IT Architect)	**Scalability** (important for the IT Architect)
A	1 month	easy online guides	basic	hundreds	basic	up to 75k
B	5 months	limited	advanced	45K + extras	basic	up to 70k
C	4 months	5 months, little docs	basic, better than A	25k + 7k/year	some, better than A,B	100K
D	6 months	2 weeks	same as A	28k + 7k/year	most secure	80k

Based on Table 3, each of the decision makers had the following alternatives, which satisfied best their top concerns:

- The Business Analyst preferred first A, then C
- The Department Manager preferred first B, then C
- The IT Architect preferred first D, then C

4.4.2 Overall, the alternatives in the experimental case included alternatives preferable to one of decision makers, and Other Experimental Materials

The other experimental materials are:

1. **Tasks descriptions.** Each student received descriptions of the experimental tasks to perform during the experiment with detailed instructions.
2. **Shared spreadsheet.** Students who used GADGET received access to a shared Google spreadsheet. Each group received a separate spreadsheet. Each spreadsheet included GADGET-specific fields (e.g. ratings, priorities) for each decision maker, and instructions on how to use the spreadsheet.
3. **Post-questionnaire.** At the end of session, students filled out a post-questionnaire about their educational background and experience, as well as their perceptions on various aspects of the group decision making process (detailed in sub-section 4.6). In addition, given the size and

challenges of designing the experimental case (as detailed in sub-section 4.4.1), we included seven questions on the experimental case itself (detailed in sub-section 4.7.3) in the post-questionnaire, so that students could give us feedback.

4. **Post-questionnaire to measure consensus.** This questionnaire included questions about prioritizing concerns and rating how well the alternatives satisfy concerns. Students filled them out from their role's point of view, but also from the perspective of the other two group members and how they would fill them out. For example, a student could indicate a set of concerns' priorities for her role, a different set of concerns' priorities for one of her colleagues, and a totally different set of concerns' priorities for the other colleague. We explain further the rationale for these measurements in sub-sections 4.5.2 and 4.5.3.

Table 4 shows an example of an item from the post-questionnaire on consensus for capturing an IT Architect's point of view. The topic of the architectural decision described in Table 4 is choosing the newsletter system that an organization is using for communicating with its customers. Alternative A is to replace the current legacy system with a third-party software-as-a-service solution. Alternative B is to pay a partner to develop a new, modern system. Alternative C is to use an open source platform and various plugins. Alternative D is to enhance the current legacy system. Students who had the role of IT Architects filled out this item with their own values for priorities of concerns (whose sum had to be 100). In addition, students filled out ratings from one to five, indicating strong disagreement, disagreement, neutral, agreement, or strong agreement on how well each of the alternatives described in the case (i.e. A, B, C, and D) satisfied each of the concerns.

To help students maintain their focus throughout the experiment, we simplified the post-questionnaire on consensus. We asked students to rate alternatives from the other roles' points of view for the ratings of two concerns, instead of six concerns. Thus, post-questionnaire items for IT Architects' point of view only had the last two rows (i.e. cost-efficient, training time), while the items for the Business Analyst role included only the first two rows, and the items for the Department Manager included only the middle two rows. This simplification helped us reduce the risk of obtaining random data as a potential reaction to being asked to perform a tedious task, by helping students to maintain their focus.

Table 4. Example of post-questionnaire item for capturing an IT Architect's priorities of concerns, and ratings of the four alternatives (i.e. A, B, C, and D) against two concerns.

Concerns	Priorities	A	B	C	D
Better analytics		-			
Higher security					
Better delivery time					
Easily scalable					
More cost-efficient					
Better training time					
Total:	100				

4.4.3 Experimental Process

Figure 3 shows the steps of the experimental process. First, we presented the plan for the session, and an overview of tasks. Second, we selected students randomly to form groups of three students, since architectural decisions involve typically three persons [3]. When the number of students was not divisible by three, we included each extra student. Third, we distributed the groups into two groups: half of the participants remained in the same room (control group), and the other half went to a different room (treatment group). Fourth, students read the experimental case and tasks descriptions. Fifth, students made the group decisions. Finally, students filled out the post-questionnaires on perceptions and consensus. During the session, we were available to answer questions from students, if necessary.

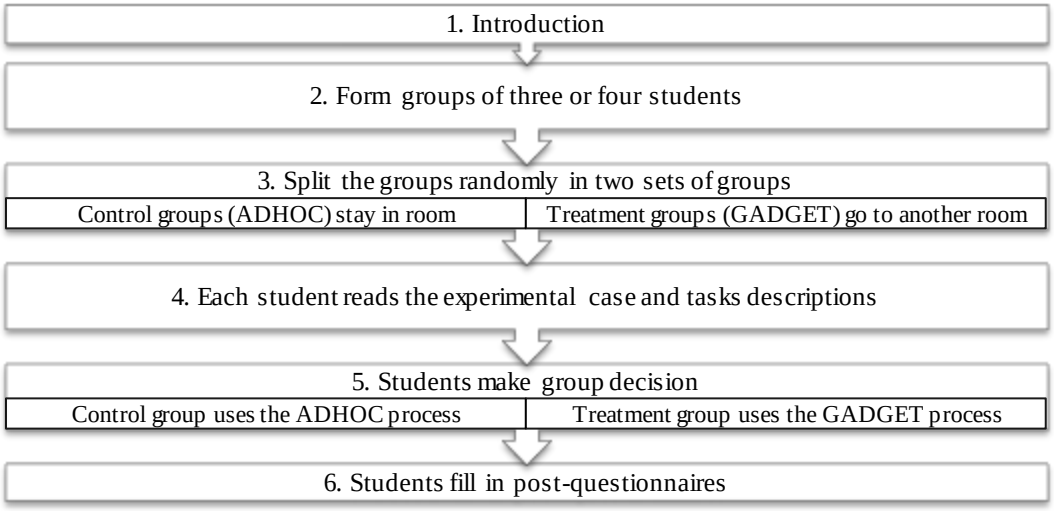

Figure 3. Students followed the above steps for the experimental process.

In general, for an experiment, a null hypothesis (H_0) states that the treatment causes no difference (e.g. using GADGET does not make any difference when compared to an ad-hoc decision making approach). The alternative hypothesis (H_1) states that the treatment makes a difference (e.g. GADGET may help or hinder reach consensus, compared to an ad-hoc approach) [31]. Based on the analysis of the data from the experiment, the null hypothesis can be rejected and the alternative hypothesis can be accepted. The analysis uses statistical tests to determine statistically significant differences between the data from the control group (e.g. ADHOC) and data from the treatment group (e.g. GADGET). Next, we present the hypotheses, including their null and alternative hypotheses, on the differences caused by the treatment in our experiment (i.e. GADGET).

4.5. Hypotheses for RQ1 – Consensus

To answer RQ1, we define metrics for operationalizing consensus among decision makers. As mentioned in Section 1, we consider two components of consensus: *general agreement* and *mutual understanding*. We define hypotheses and metrics on both components of consensus.

4.5.1 Hypothesis on General Agreement

Regarding *general agreement*, we defined a metric that counts how many groups reached agreement on their group architectural decision. For example, if no group reached agreement on their group architectural decisions, then this metric is zero. Using this metric, we define the following hypothesis.

H_{a0}: ADHOC and GADGET result in the same **general agreement** among group decision makers.
H_{a1}: GADGET results in higher **general agreement** than ADHOC**.**

4.5.2 Hypothesis on Mutual Understanding on the Priorities of Concerns

Regarding *mutual understanding* among decision makers, a group has high mutual understanding on a decision, if group members are also able to indicate accurately the perspectives of the other group members on that decision. For example, let us consider three architects (Anne, Bob, and Charlie) who need to make a group architectural decision on which framework (e.g. A, B, C, or D) to use for a new software system. High mutual understanding among the three architects means that, after discussions, each of the three architects is able to estimate accurately what the other two architects think about the performances of each framework. In contrast, low mutual understanding may suggest the input from the other group members was not taken seriously, which resulted in misunderstandings among architects on each other's perspectives (e.g. at the end of the discussion, Charlie has no idea what Anne thinks about the performance of the C framework, although Anne mentioned this during the discussion).

Priorities of concerns are a ratio type of data, which means that calculating differences between priorities is allowed. For the metric related to the mutual understanding on the priorities of concerns, we calculate the sum of absolute differences between the priorities assigned by a student, and the priorities that the student's group colleagues estimated. Based on these assumptions, equation (1) summarizes the metric for calculating mutual understanding on priorities (MUP) of concerns, for a decision with six concerns (see Table 4) in a group of three decision makers. p_{Ai} stands for the priority indicated by architect A for the i concern, from A's point of view. $p_{i,j}$ stands for the priority estimated by colleague j for the i concern, as colleague j estimates that A indicated. MUP ranges from 0 to 100. Lower values for the metric mean higher mutual understanding among group decision makers, due to smaller differences between estimated and actual priorities.

$$MUP_A = \sum_{j=1}^{2} \sum_{i=1}^{6} |p_{Ai} - p_{i,j}| \qquad (1)$$

Using the above metric, we propose the following hypothesis.

H_{b0}: ADHOC and GADGET result in the same level of **mutual understanding on priorities of concerns** among group decision makers.
H_{b1}: GADGET results in higher **mutual understanding on priorities of concerns** than ADHOC**.**

4.5.3 Hypothesis on Mutual Understanding on Ratings

Ratings of alternatives are provided on a 5-point Likert scale, which may be considered an ordinal type of data. This means that summing differences among ratings (similar to eq. (1) in sub-section 4.5.2) is problematic. Instead of summing differences among ratings, we use the standard deviation to measure the variation among ratings. Similar to the metric for priorities, we calculate the standard deviation for one's own ratings, and the ratings that the other decision makers in the group estimated for one's ratings. Lower

values for the standard deviation indicate higher mutual understanding on ratings among group decision makers, due to smaller variation between estimated and actual priorities.

Using the standard deviation metric, we propose the following hypothesis.

H_{c0}: ADHOC and GADGET result in the same level of **mutual understanding on ratings of alternatives against concerns** among group decision makers.
H_{c1}: GADGET results in higher **mutual understanding on ratings of alternatives against concerns** than ADHOC**.**

4.6. Hypotheses for RQ2 - Perceptions

To answer RQ2, we defined metrics to measure the perceptions of the group decision makers about the process they use (i.e. GADGET or ADHOC). Based on existing literature, we propose three categories of perceptions: on benefits of using GADGET, challenges related to the use of GADGET, and satisfaction from using a group decision making process. For each category, we propose several perception items. Each perception item is operationalized by indicating the level of agreement with items in the post-questionnaire, using a five-point Likert scale (i.e. from strong disagreement to strong agreement). The items in the post-questionnaire originate from the initial GADGET evaluation in Section 2, and literature on decisions. Table 5 shows the perception categories, perception and post-questionnaire items, as well as the literature source for the items.

Table 5. Mapping of perception categories, metrics, and post-questionnaire items.

ID	Perception category	Perception metric item	Post-questionnaire item	Source
M1.	Benefits	Reevaluation of initial perspective	After discussing the case with my team I changed my mind regarding the importance of one or more concerns	[32, 33]
M2.		Reveals extra points	The discussion with my team revealed valid points that I would not be able to consider on my own	[32, 33]
M3.		Reusability	The artefacts (documents, notes, tables, spreadsheets, etc.) that my team created during the decision-making session could be reused to examine similar situations in the future.	[34, 35]
M4.		Rationale	The artefacts that my team created during the decision-making session could be used to justify to other people the reasons we made this decision.	[34, 35]
M5.		Clarifies problem	After the decision-making session, my team had a clearer view on ASO's problem	[36]
M6.		Improves decision making skills	The decision-making session improved my decision-making skills	[37]
M7.	Challenges	Low understandability	It was too difficult for me to understand what I was required to do	[37]
M8.		Clarity of instructions	The instructions were clear enough	[37]
M9.		Long time for decision	I believe that the decision-making session required too much time	[9, 36]
M10.		Large effort	I believe that the decision-making session required too much effort	[9, 36]
M11.		Long preparation time	It took me too long to understand what I was required to do in the decision-making session	[9, 36]
M12.	Satisfaction	Willingness for future	I would be willing to work with the same team on other projects in the future	[32]

		collaboration		
M13		Satisfaction on cooperation	Working together with my teammates was an enjoyable experience	[32]
M14		Enjoyment	I enjoyed the decision-making session	[32]
M15		Commitment	I strongly support my group's final decision	[32]
M16		Overall satisfaction	I am satisfied with my group's decision	[32]

Based on the 16 metrics in Table 5, we define 16 hypotheses, as follows. Since the hypotheses are similar and only the metrics vary, we formulate a generic hypothesis, which is adaptable to each of the 16 hypotheses.

H_{Mi0}: ADHOC and GADGET result in similar **perceptions on the M_i metric** (where M_i varies from M1 to M16), among group decision makers.
H_{Mi1}: ADHOC and GADGET result in different **perceptions on the M_i metric** among group decision makers**.**

In summary, the independent variable for this experiment is the group decision making process (i.e. GADGET or ADHOC). The dependent variables for RQ1 and RQ2 are summarized in Table 6.

Table 6. Summary of dependent variables for each research question.

RQ	Hypothesis	Metric description	Scale type	Range
RQ1	H_{a0}	General agreement	Nominal	Yes/no
	H_{b0}	Sum of differences between priorities of concerns	Ratio	Zero or more
	H_{c0}	Standard deviation of ratings	Ratio	Zero or more
RQ2	H_{Mi0}	16 perception metrics	Interval	1 to 5

4.7. Results

The experiment took place in three sessions. The first session took place with 18 students at the University of Groningen. The second session took place with 72 students at the University of Vienna. The third session took place with 23 students at the University of Pretoria. All sessions followed the same experimental process. After performing the experimental sessions, we discarded data from 11 students, due to missing or incomplete values. The valid data from the remaining 102 students was analyzed as described in sub-section 4.7.1. Figure 4 summarizes the number of students and groups from each university across the control (i.e. ADHOC) and treatment (i.e. GADGET) groups, showing the full number of students (i.e. 113) and groups, as well as the numbers for valid data only (i.e. 102 students).

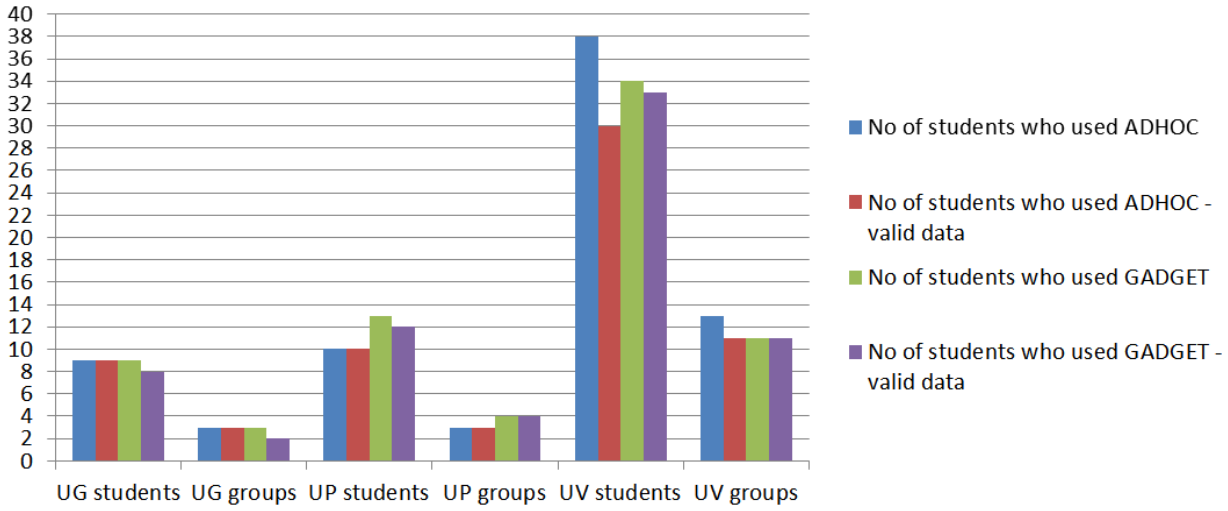

Figure 4. Summary of the number of students and groups from the University of Groningen (UG), the University of Pretoria (UP), and the University of Vienna (UV). Most invalid data came from UV students who used ADHOC.

4.7.1 Analysis Procedure

To analyze the collected data, we defined analysis procedures for investigating the hypotheses in sub-sections 4.5 and 4.6. Table 7 summarizes the analysis procedures for all hypotheses. We used the Mann-Whitney U test because it is well suited for comparing two independents samples (i.e. the treatment/GADGET and control/ADHOC groups). Furthermore, this statistical test is non-parametric (i.e. it makes no assumption regarding the normal distribution of the data), which is suitable to this experiment, since we cannot assume that the data is normally distributed. Still, we checked the normality of the data using the Shapiro-Wilk test, to confirm the validity of using a non-parametric test. We used IBM SPSS for applying statistical tests.

Table 7. Summary of hypotheses and their analysis procedure.

Research question	Hypothesis	Hypothesis number	Analysis procedure
RQ1 Consensus	Agreement	H_{a0} - H_{a1}	Binomial test
	Mutual understanding (priorities of concerns)	H_{b0} - H_{b1}	Mann-Whitney U tests
	Mutual understanding (ratings of alternatives against concerns)	$H_{c0} - H_{c1}$	
RQ2 Perceptions	Benefits, challenges and satisfaction	H_{Mi0} - H_{Mi1} *Mi covers M1 to M16*	

4.7.2 Participants' Background

Regarding background, we asked participants to indicate their number of years of practical experience in software engineering. Figure 5 summarizes the results. Five students declined to respond. One third of the students had more than one year of practical experience. Participants' levels of experience are balanced across the treatment (i.e. GADGET) and control (i.e. ADHOC) groups.

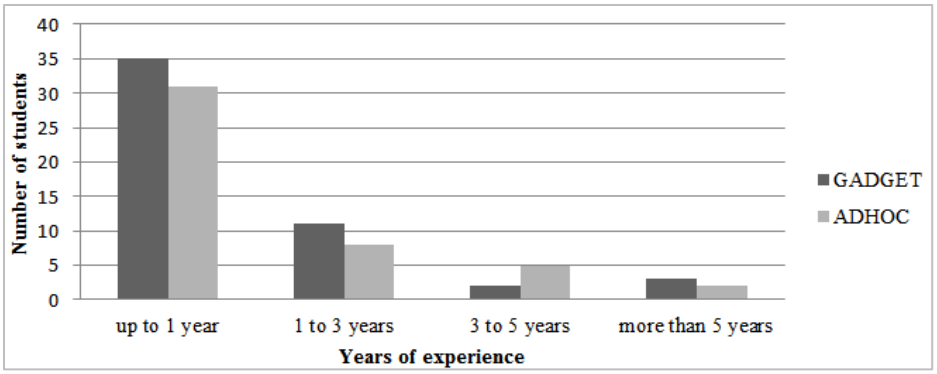

Figure 5. Summary of the years of practical experience in software engineering of the students.

4.7.3 Participants' Feedback on the Experimental Case

Table 8 indicates the seven statements in the post-questionnaire which were rated by participants from one (strongly disagree) to five (strongly agree). Table 8 includes feedback from the 102 students who offered valid data, for both the treatment and control groups. The feedback indicates agreement with the statements one, six, and seven, neutrality on statements four and five, and disagreement with statements two and three.

The results in Table 8 indicate the following. The experimental case included the right amount of information (i.e. the needed information, without too many details) in an easy to understand manner, although the description of the decision alternatives could have been clearer (given the neutral answers on statement three). Students were comfortable with their roles, which was important for us to find out, given that their roles had different preferences on the decision alternatives (as detailed in 4.4.1).

Table 8. The statements on the experimental case were rated by participants. The median and mean for each statement indicate agreement or disagreement with each statement.

Statement number	Statement	Median	Mean
1	The experimental case was well documented	4	3.80
2	The experimental case included too many details	2	2.39
3	I found it difficult to understand the experimental case	2	1.90
4	I enjoyed reading the experimental case	3	3.24
5	The alternative solutions were too vague	3	2.73
6	The alternative solutions' descriptions included all the information my team needed to make the decision	4	3.41
7	I felt comfortable with the role I had to play	4	3.80

4.7.4 Answer to RQ1 - Consensus

To answer RQ1, we tested the three hypotheses on the two components of consensus (i.e. *agreement* and *mutual understanding*) summarized in Table 7. Regarding the hypothesis on *agreement*, we found that all groups from both treatments reached consensus. Therefore, we cannot reject the null hypothesis (H_{a0} – detailed in sub-section 4.5.1), and conclude that both GADGET and ADHOC result in agreement among group decision makers.

Table 9 summarizes the values for the hypotheses on *mutual understanding* on *priorities* of concerns and ratings. For example, the average values for the metrics on priorities (as defined in sub-section 4.5.2) were 133.91 for students in the control group (ADHOC), and 102.69 for students in the treatment group

(GADGET). We checked the normality of the data using the Shapiro-Wilk test, and we found out that the data was not normally distributed (p-value = 0.011). The non-parametric Mann-Whitney U test on H_b returned a statistically significant difference (p-value = 0.0003). Therefore, we reject the null hypothesis (i.e. H_{b0} in sub-section 4.5.2), and conclude that GADGET results in higher consensus for priorities of concerns among group decision makers.

Regarding the hypothesis on *mutual understanding* on *ratings* of alternatives against concerns, we found lower standard deviations of ratings in the GADGET group. The average values for metrics on ratings (as defined in sub-section 4.5.3) was 1.29 for students in the control group (ADHOC), and 1.12 for students in the treatment group (GADGET). The Shapiro-Wilk test for normality indicated the data was not normally distributed (p-value = 0.015). The Mann-Whitney U test returned a statistically significant difference (p-value = 0.00001). Therefore, we reject H_{c0}, and we conclude that GADGET results in higher consensus for ratings among group decision makers.

Table 9. Medians and means for ADHOC and GADGET for the metrics on mutual understanding.

Hypothesis number	Hypothesis	Metric description	Median (mean) ADHOC	Median (mean) GADGET	p-value
H_b	Mutual understanding on priorities of concerns	Sum of differences between priorities of concerns	130 (133.91)	95 (102.69)	0.0003
H_c	Mutual understanding on ratings	Standard deviation of ratings	1.31 (1.29)	1.17 (1.12)	0.00001

4.7.5 Answer to RQ2 - Perceptions

To understand how perceptions on GADGET and ADHOC differ among decision makers (i.e. RQ2), we tested the 16 hypotheses defined in sub-section 4.6 on students' perceptions on the GADGET and ADHOC approaches. The Shapiro-Wilk test for normality indicated that data for all metrics was not normally distributed (p-value = 0.000). After applying Mann-Whitney U tests, we found statistically significant ($p < 0.05$) differences on eight metrics. We rejected H_{M30}, H_{M40}, H_{M70}, H_{M90}, H_{M100}, H_{M120}, H_{M130}, and H_{M160}. We accepted their corresponding alternative hypotheses: H_{M31}, H_{M41}, H_{M71}, H_{M91}, H_{M101}, H_{M121}, H_{M131}, and H_{M161}. Table 10 summarizes the results for the 16 hypotheses corresponding to M1 to M16, including the medians and means for the results on each perception metric for GADGET and ADHOC using a scale from 1 (i.e. strong disagreement) to 5 (i.e. strong agreement).

Table 10. Results for perception metrics on ADHOC and GADGET. Shaded rows indicate statistically significant differences of perceptions.

ID	Perception category	Perception metric item	Median (mean) ADHOC	Median (mean) GADGET	p-value
M1.	Benefits	Reevaluation of initial perspective	3 (2.90)	3 (3.17)	.192
M2.		Reveals extra points	3 (3.33)	3 (3.17)	.375
M3.		**Reusability**	**3 (2.76)**	**4 (3.55)**	**.019**
M4.		**Rationale**	**4 (2.86)**	**4 (3.77)**	**.005**
M5.		Clarifies problem	4 (3.88)	4 (3.83)	.821
M6.		Improves decision making skills	3 (3.27)	3 (3.25)	.641
M7.	Challenges	**Low understandability**	**1 (1.31)**	**2 (1.64)**	**.007**
M8.		Clarity of instructions	4 (4.27)	4 (4.15)	.352
M9.		**Long time for decision**	**2 (2.10)**	**2 (2.58)**	**.007**
M10.		**Large effort**	**2 (1.96)**	**3 (2.57)**	**.0003**
M11.		Long preparation time	2 (1.65)	2 (1.83)	.222
M12.	Satisfaction	**Willingness for future collaboration**	**4 (4.29)**	**4 (3.7)**	**.00006**
M13.		**Satisfaction on cooperation**	**4 (4.29)**	**4 (3.94)**	**.005**

M14.		Enjoyment	4 (4.18)	4 (3.81)	.157
M15.		Commitment	4 (4.14)	4 (3.87)	.155
M16.		**Overall satisfaction**	**4 (4.15)**	**4 (3.87)**	**.037**

4.8. Discussion

Controlled experiments are particularly useful for establishing causal relationships [31]. In this experiment, we compared the impact of the group decision making approach (i.e. GADGET or ADHOC) on two components of consensus: mutual understanding and general agreement. We found out that GADGET performs better than ADHOC at increasing mutual understanding among decision makers, for both priorities of concerns and ratings of alternatives against concerns. We found no difference between GADGET and ADHOC at the general agreement.

Additionally, we found statistically significant differences between perceptions (RQ2) on GADGET vs. ADHOC as follows:

- Regarding perceptions on the **benefits** of GADGET vs. ADHOC approaches, reusability of created artefacts (e.g. alternatives, rationale) while using the approaches was significantly higher for GADGET. In addition, the GADGET approach allowed better capturing of the rationale for the architectural decisions than ADHOC. However, we found no significant differences on reevaluating the initial perspectives, revealing extra points, problem clarification, and improving decision making skills.
- Regarding perceptions on the **challenges** of using GADGET vs. ADHOC, we found the following significant differences. GADGET users had more difficulties understanding the process than ADHOC users, which reflects the learning curve of GADGET. In addition, GADGET users perceived a higher time and effort to make decisions compared to ADHOC, which reflects the effort of using a structured approach for group decision making. However, we found no differences on the clarity of the instructions and the preparation time.
- Regarding perceptions on the **satisfaction** of using GADGET vs. ADHOC, we found significantly higher willingness for future collaboration with the same team members for ADHOC. Also, ADHOC users reported higher satisfaction on cooperation and overall higher satisfaction with their decisions than GADGET users. However, we found no significant differences on enjoying the session, and on one's commitment to one's group final decision.

4.8.1 Interpretation of Results

These findings mean the following:

- Regarding consensus among decision makers, this experiment indicates GADGET's positive effect on increasing consensus. The combined evidence from the case study in Section 3 and the experiment in this section indicates that practitioners can use GADGET to increase consensus in their architectural decisions.
- Regarding the results on the **benefits** of GADGET vs. ADHOC, the results on reusability and capturing rationale in the experiment confirmed the results from the case study. These benefits help practitioners avoid architectural knowledge vaporization, and reduce maintenance costs. For the remaining four items on benefits (i.e. reevaluation of initial perspective, revealing extra points, clarifying problem, and improving decision making skills), the results in Table 10 indicate

no differences between GADGET and ADHOC, which means these four items are not key benefits of GADGET.

- Regarding meaning of results on the **challenges** of GADGET vs. ADHOC, the results indicate there is a higher cost for decision makers in terms of time and effort for using GADGET. These results were obtained in the context of first-time users of GADGET and not-first time users of ADHOC (since participants were very likely to have made other group decisions before the experiment, given their years of experience, as shown in Figure 5). We can expect that the effort of using GADGET would decrease for subsequent uses, after passing its learning curve. Still, the lack of differences on instructions clarity and preparation time (in Table 10) suggests that participants could learn about GADGET from the written instructions they received. Overall, although GADGET has a learning curve, we expect practitioners to progress fast on the learning curve.
- Regarding meaning of results on **satisfaction** on using GADGET vs. ADHOC, we note that ADHOC scored more favorably than GADGET. However, the results on GADGET still show positive satisfaction from using GADGET. Overall, practitioners who use GADGET for the first time can expect positive satisfaction, although lower than ADHOC, which is more familiar to practitioners.

4.8.2 Additional Remarks

From the case study and the experiment, we learnt that GADGET increases consensus among participants. Furthermore, GADGET helps make better decisions, by encouraging decision makers to evaluate systematically alternatives. Finally, GADGET reduces architectural knowledge vaporization by capturing the rationale of the group decision.

As visible in section 2.2, GADGET uses a minimalistic and accessible set of software architecture-specific concepts (e.g. concerns, alternatives), to help involve stakeholders with a diverse background and limited expertise in software architecture. Such stakeholders appreciate a group decision method that is accessible to a wider audience.

Furthermore, GADGET is built on the assumption that group decision making in software architecture is not **fundamentally** different from group decision making in other domains. We have two arguments in favor of this statement. First, there are decision making methods which have proven successful across a variety of domains (e.g. Delphi). This suggests cross-domain commonalities among decision making methods. Second, there is at least one successful architectural decision making method which is based on ideas from another domain: CBAM [43] has roots in economic modeling. Still, different domains have different challenges, so it is important to bring empirical evidence on any proposed method for group architectural decision making, regardless if it was validated in a different domain. We provide such evidence for GADGET in sections 3 and 4.

4.8.3 Limitations of GADGET

There are a few limitations for applying GADGET in practice. GADGET assumes participants in the group decision making are on a similar hierarchy level, and no politics are involved in the decision making. Other factors include social relationships among participants. For example, if the group has high cohesion, then the group decision making process might be easier to adopt and follow. Still, more work is

needed to understand these limitations and their influence on the adoption and results of group decision making processes, such as GADGET.

We regard GADGET as a useful tool in architects' toolbox, but not as the only tool in the toolbox. GADGET does not intend to cover the full architecture design process, or even all types of group architectural decisions (e.g. series of strongly coupled architectural decisions). Section 7 suggests future work to cover more aspects of group architectural decision making. Overall, GADGET provides clear value for its intended use.

5. Validity Threats

Using guidelines from [22] and [31], we present construct, internal, external, and conclusion validity threats for the case study (detailed in Section 3) and experiment (detailed in Section 4).

5.1. Case Study Validity Threats

Construct validity is about the generalization of study results to the theory behind the study [31]. To avoid this threat, we conducted the case study not only with students (two groups), but also with practitioners (four groups). Furthermore, we prevented interviewer (i.e. to please researchers) and response biases (i.e. responses that make participants look good) by encouraging participants to criticize GADGET openly. In turn, this helped us collect areas for improvement, as reported in sub-section 3.2.2. Finally, participants were anonymized and had no incentive (e.g. grades, money) to please researchers.

Internal validity threats refer to the extent to which the independent variable was responsible for the effects on the dependent variables [31] [22]. Internal validity threats were not applicable for the case study, since we did not attempt to show any causality relationship.

External validity threats refer to the ability of generalizing our results to practice [31]. To address this threat, we involved practitioners in the case study. Furthermore, the students who participated in the case study also had practical experience (as presented in Table 2). Still, there are factors that complicate group decision making in practice: different hierarchy levels among participants, hidden agendas, group dynamics, and politics. Such factors were out of scope for this paper.

Conclusion validity threats regard issues affecting the ability to draw accurate study conclusions [31]. The study conclusions were drawn based on the results from the content analysis of interviews with participants, using guidelines from the literature [18]. To ensure accurate conclusions, two researchers were involved in the content analysis of the interviews with participants. The researchers made sure that there was high agreement in their interpretation of the data.

5.2. Experiment Validity Threats

We addressed **construct validity** by operationalizing the constructs in our experiment: we defined metrics for each hypothesis (see sub-sections 4.5 and 4.6). Furthermore, to avoid impact on participants' behavior, we made clear to the participants that the experiment would not have any impact on their grades. Additionally, to avoid hypotheses guessing and evaluation apprehension, we did not tell participants our hypotheses.

To address **internal validity** threats, such as the instrumentation validity threat, we made a pilot for the experiment [30], to increase the clarity of the experimental package. For example, we increased the readability of the questionnaire, so that participants can easily understand their tasks. We addressed the *mortality* validity threat by integrating the study with the software architecture course (see sub-section 4.3), so that participants joined it voluntarily for the educational value. We distributed participants randomly to the groups to avoid selection threats. Furthermore, by using students we increased internal validity, since using practitioners means larger variation in confounding variables such as domains, types of previous projects, or previous experiences.

Another instrumentation validity threat is that students took roles (i.e. department manager, IT architect, or business analyst) for which they had little or no experience. To address this threat and to avoid relying on the experience of participants, we gave each student printouts with the description of their corresponding role. This description contained all the information they needed to make the decision and to participate in the group decision process. Thus, it was not necessary that students required external sources of information during the experiment, or previous experience. Furthermore, as detailed in sub-section 4.7.3, feedback from students indicates the experimental case had the right amount of information, and students were comfortable with their roles.

Regarding **external validity**, Kitchenham et al. regard students as relatively close to the population of interest, because they are the next generation of software professionals [38]. We consider our results as applicable to inexperienced architects, rather than senior architects. Since inexperienced architects need more support than senior architects, it is reasonable to use students in the experiment, instead of senior architects. Moreover, the nature of tasks students had to perform did not require experience levels of senior architects, as students had sufficient knowledge to perform their tasks. To ensure the commitment of the participants, we made sure that the experiment contributes to participants' education (see sub-section 4.3). To check whether or not GADGET is also applicable to more experienced or senior architects, we need to conduct a future similar experiment with practitioners.

Regarding **conclusion validity**, statistical tests have various assumptions, and violating them may lead to poor conclusions. We used non-parametric tests that make fewer assumptions, such as Mann-Whitney. By conducting the experiment with a large sample of students from multiple universities, we aimed at increasing tests' statistical power. Another potential threat is that some metrics (e.g. perceptions) tend to be less reliable than others (e.g. ratings). To address this threat, we piloted our study [30] to clarify wording, and avoid misunderstandings.

6. Related Work

Outside the software architecture domain, there is much interest in group decision making. For example, Herrera-Viedma et al. [40, 41] conceptualize group decision making in two sub-processes: consensus and selection. Consensus focuses on getting a maximum degree of consensus between experts, and selection refers to selecting the actual decision alternative. Herrera-Viedma et al. [40, 41] use two metrics for consensus in group decision making. The first one is a consensus measure, to evaluate the general agreement of all experts. The second one is a proximity measure to evaluate agreement between an individual and the group. By providing a feedback mechanism to the persons in a group, decision makers can re-evaluate their perspectives and increase their proximity to the group perspective, thus increasing

consensus. The group decision making process in [40, 41] allows decision makers to express their preferences in much detail and more formally (e.g. using fuzzy preference relations) than GADGET. In comparison, GADGET offers more simplicity, thus making the process easier to use for architects.

There have been a few approaches and studies on group architectural decisions. Zannier et al. describe real-world architectural decisions, and ask for more work on understanding real-world group architectural decisions [42]. Kazman et al. propose an extension of CBAM [43] that considers explicitly the preferences of group architectural decision makers [44]. Recently, Rekha and Muccini analyze real-world group architectural decision making [45]. Nowak and Pautasso analyze situational awareness in group architectural decision making [46]. Gaubatz et al. propose automatic enforcements of constraints in group architectural decisions [47]. Groher and Weinreich analyze four approaches for group decision making that were proposed by students with practical experience [48]. In this paper, we focus on a particular aspect of group architectural decision making (i.e. increasing consensus), which has not been addressed in previous work.

Related work on processes for group architectural decision making include the following. Babar et al. studied the feasibility of groupware support for architecture evaluation, with applicability on architectural decisions [49]. Al-Naeem et al. propose using the Analytical Hierarchy Process in group architectural decision making [50]. Nakakawa et al. propose a theoretical model on group architectural decision making for enterprise software systems [51]. Sousa et al. present a process for group architectural decision making, in which a facilitator helps the group interactions [52]. In this paper, the proposed GADGET process does not require a facilitator, while our focus is on presenting empirical evidence on the GADGET process.

Related work on approaches that capture architectural knowledge and help group architectural decisions include the following. Falessi et al. reported an experiment with students on documenting the rationale of group architectural decisions [53]. Mohan and Ramesh propose a traceability framework for group architectural decisions [54]. Zimmermann et al. propose a framework for capturing architectural decisions which can help group architectural decisions [55]. In this paper, we provide evidence that the GADGET process reduces architectural knowledge vaporization.

Tang [56] mentions communication issues that may appear in group architectural decision making, but no process improvement is offered. Also, Kazman et al. [57] describe the importance of consensus for the ATAM approach, but without describing how to increase consensus for architectural decisions. Furthermore, the Attribute-Driven Design method [58] does not indicate how to increase consensus in group architectural decisions. In contrast, in this paper we provide evidence on how GADGET increases consensus in group architectural decisions.

7. Conclusions and Future Work

In this paper, we evaluate GADGET, an upfront process for increasing consensus in group architectural decisions. GADGET was motivated by noticing that most architectural decisions are made in groups [3, 4] and that little research exists on group architectural decisions [5]. Consensus is conceptualized in terms of its two main components: general agreement and mutual understanding. GADGET is based on the Delphi technique and our previous work on using the Repertory Grid technique to make and capture architectural decisions [8-10]. GADGET was evaluated with students and practitioners, in a case study

and an experiment. Thirteen practitioners and eight students participated in the case study, and 113 students participated in the experiment.

From the case study, we identified the need for increasing consensus in group architectural decisions. In addition, we found that GADGET helps practitioners increase consensus in group architectural decisions. From the experiment, we found that GADGET and ADHOC resulted in agreement among group decision makers, while GADGET resulted in higher mutual understanding than ADHOC. GADGET provides significantly higher reusability of architectural decisions and more captured rationale than ADHOC. However, GADGET requires more effort than ADHOC.

The results of the two studies in this paper indicate that GADGET helps practitioners, and particularly inexperienced architects to increase consensus in group architectural decisions, and capture the rationale of architectural decisions. Still, group architectural decision making is a multifaceted topic, since in practice group decisions can be influenced by factors such as hierarchy levels, hidden agendas, or politics. Such factors were out of scope for this paper. Overall, for architectural decisions in which such factors do not play a role, GADGET is particularly useful for increasing consensus in group architectural decisions and capturing the rationale of the decisions.

Additionally, more approaches for prioritizing concerns can be used, such as ranking or pairwise comparisons. Currently, we only used ratings from one to five, but in future work we consider adding other types of ratings, such as specific categories. Additionally, uncertainties in the decision need to be addressed explicitly.

This paper opens several directions for future work:

1. **GADGET refinements** - since there is a need for treating uncertainty in architectural decision making [5], we will update GADGET to include support for uncertainty in group architectural decisions. Also, we will investigate and collect evidence on the value of using more fine-grained iterations among GADGET steps than the current iteration in step 5 of GADGET.
2. **GADGET for senior architects** – since this paper focuses on inexperienced architects, we will analyze GADGET for senior architects.
3. **Understanding group architectural decision making** - there is a need to further understand group architectural decision making in practice, as also noticed in previous work [5]. For example, there is a need to define criteria (e.g. extending the criteria in Table 5) for evaluating various group decision making processes. One such criterion can be the influence of the group decision making processes on the quality of the architectural decisions. Finally, further research is needed on the influence of hierarchy levels, hidden agendas, or politics on group architectural decision making.
4. **Supporting group architectural decision making** – once our understanding of group architectural decision making in practice is mature, we should be able to support practitioners tackle other challenges of group architectural decision making, besides consensus.

8. Acknowledgments

We thank Konstantinos Tselios for his help in conducting the studies, and participants for their efforts.

9. References

[1] A. Jansen, J. Bosch, Software architecture as a set of architectural design decisions, in: 5th Working IEEE/IFIP Conference on Software Architecture, 2005, pp. 109-120.
[2] O. Zimmermann, Architectural Decisions as Reusable Design Assets, IEEE Software, 28 (2011) 64-69.
[3] D. Tofan, M. Galster, P. Avgeriou, Difficulty of Architectural Decisions – a Survey with Professional Architects, in: 7th European Conference on Software Architecture, Springer, 2013, pp. 192-199.
[4] C. Miesbauer, R. Weinreich, Classification of Design Decisions – An Expert Survey in Practice, in: K. Drira (Ed.) Software Architecture, Springer Berlin Heidelberg, 2013, pp. 130-145.
[5] D. Tofan, M. Galster, P. Avgeriou, W. Schuitema, Past and future of software architectural decisions – A systematic mapping study, Information and Software Technology, 56 (2014) 850-872.
[6] M. Svahnberg, An industrial study on building consensus around software architectures and quality attributes, Information and Software Technology, 46 (2004) 805-818.
[7] W.J. Tastle, M.J. Wierman, Consensus and dissention: A measure of ordinal dispersion, International Journal of Approximate Reasoning, 45 (2007) 531-545.
[8] D. Tofan, P. Avgeriou, M. Galster, Validating and Improving a Knowledge Acquisition Approach for Architectural Decisions, International Journal of Software Engineering and Knowledge Engineering, 24 (2014) 553-589.
[9] D. Tofan, M. Galster, P. Avgeriou, Capturing Tacit Architectural Knowledge Using the Repertory Grid Technique (NIER Track), in: 33rd International Conference on Software Engineering, 2011, pp. 916-919.
[10] D. Tofan, M. Galster, P. Avgeriou, Reducing Architectural Knowledge Vaporization by Applying the Repertory Grid Technique, in: 5th European Conference on Software Architecture, Springer, Germany, 2011, pp. 244-251.
[11] H. Linstone, M. Turoff, The Delphi method: Techniques and applications, 2002.
[12] D. Jankowicz, The easy guide to repertory grids, Wiley, 2003.
[13] M. Hassenzahl, R. Wessler, Capturing Design Space From a User Perspective: The Repertory Grid Technique Revisited, International Journal of Human-Computer Interaction, 12 (2000) 441-459.
[14] A.F. Osborn, Applied Imagination; Principles and Procedures of Creative Problem-solving: Principles and Procedures of Creative Problem-solving, Scribner, 1963.
[15] A.L. Delbecq, A.H. Van de Ven, A group process model for problem identification and program planning, The Journal of Applied Behavioral Science, 7 (1971) 466-492.
[16] P. Runeson, M. Host, A. Rainer, B. Regnell, Case study research in software engineering: Guidelines and examples, John Wiley & Sons, 2012.
[17] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Softw. Engg., 14 (2009) 131-164.
[18] K. Krippendorff, Content analysis: An introduction to its methodology, Sage Publications, Inc, 2004.
[19] D. Tofan, Appendix, http://www.cs.rug.nl/~dan/GADGETexperiment/, last accessed on September, 2015.
[20] D. Tofan, M. Galster, Capturing and Making Architectural Decisions: an Open Source Online Tool, in: Proceedings of the 2014 European Conference on Software Architecture Workshops, ACM, Vienna, Austria, 2014, pp. 1-4.
[21] R.K. Yin, Case study research: Design and methods, Sage Publications, 2003.
[22] A. Jedlitschka, M. Ciolkowski, D. Pfahl, Reporting experiments in software engineering, in: F. Shull, J. Singer, D. Sjøberg (Eds.) Guide to advanced empirical software engineering, Springer, 2008, pp. 201-228.
[23] V. Venkatesh, F.D. Davis, A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, Management Science, 46 (2000) 186-204.
[24] M. Kubickova, H. Ro, Are students "Real People"? The Use of Student Subjects in Hospitality Research, (2011).

[25] P. Runeson, Using students as experiment subjects–an analysis on graduate and freshmen student data, Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering.– Keele University, UK, (2003) 95-102.
[26] M. Svahnberg, A. Aurum, C. Wohlin, Using students as subjects - an empirical evaluation, in: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, ACM, Kaiserslautern, Germany, 2008, pp. 288-290.
[27] P. Berander, Using students as subjects in requirements prioritization, in: Proceedings of the International Symposium on Empirical Software Engineering, 2004, pp. 167-176.
[28] J.C. Carver, L. Jaccheri, S. Morasca, F. Shull, A checklist for integrating student empirical studies with research and teaching goals, Empirical Software Engineering, 15 (2009) 35-59.
[29] M. Galster, D. Tofan, P. Avgeriou, On Integrating Student Empirical Software Engineering Studies with Research and Teaching Goals, in: Proceeding of the Evaluation and Assessment in Software Engineering, 2012.
[30] K. Tselios, P. Avgeriou, D. Tofan, Two empirical studies on decision-making processes in software architecture (Master's Thesis), in, University of Groningen, 2012.
[31] C. Wohlin, P. Runeson, M. Hst, M.C. Ohlsson, B. Regnell, A. Wessln, Experimentation in Software Engineering, Springer Publishing Company, Incorporated, 2012.
[32] D.M. Schweiger, W.R. Sandberg, J.W. Ragan, Group Approaches for Improving Strategic Decision Making: a Comparative Analysis of Dialectical Inquiry, Devil'S Advocacy, and Consensus, Academy of Management Journal, 29 (1986) 51-71.
[33] R.T. Hartwig, Facilitating Problem Solving : A Case Study Using the Devil's Advocacy Technique, Group Facilitation: A Research & Applications Journal, (2010) 17.
[34] A. Jansen, J. van der Ven, P. Avgeriou, D.K. Hammer, Tool Support for Architectural Decisions, in: Proc. Working IEEE/IFIP Conference on Software Architecture WICSA '07, 2007, pp. 4-4.
[35] A. Tang, H. van Vliet, Software Architecture Design Reasoning, in: Software Architecture Knowledge Management, Springer, 2009, pp. 155-174.
[36] V.S. Lai, B.K. Wong, W. Cheung, Group decision making in a multiple criteria environment: A case using the AHP in software selection, European Journal of Operational Research, 137 (2002) 134-144.
[37] B.C. Hardgrave, F.D. Davis, C.K. Riemenschneider, Investigating Determinants of Software Developers' Intentions to Follow Methodologies, J. Manage. Inf. Syst., 20 (2003) 123-151.
[38] B. Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. El Emam, J. Rosenberg, Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering, 28 (2002) 721-734.
[39] E. Triantaphyllou, Multi-criteria decision making methods: a comparative study, Springer Science & Business Media, 2013.
[40] E. Herrera-Viedma, F. Herrera, F. Chiclana, A consensus model for multiperson decision making with different preference structures, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 32 (2002) 394-402.
[41] E. Herrera-Viedma, L. Martinez, F. Mata, F. Chiclana, A Consensus Support System Model for Group Decision-Making Problems With Multigranular Linguistic Preference Relations, IEEE Transactions on Fuzzy Systems, 13 (2005) 644-658.
[42] C. Zannier, M. Chiasson, F. Maurer, A model of design decision making based on empirical results of interviews with software designers, Information and Software Technology, 49 (2007) 637-653.
[43] R. Kazman, M. Klein, Quantifying the costs and benefits of architectural decisions, in: Proceedings of the International Conference on Software Engineering, IEEE, 2001, pp. 297-306.
[44] R. Kazman, H.P. In, H.-M. Chen, From requirements negotiation to software architecture decisions, Information and Software Technology, 47 (2005) 511-520.
[45] S. Rekha, H. Muccini, A Study on Group Decision-Making in Software Architecture, in: 11th Working IEEE/IFIP Conference on Software Architecture (WICSA 2014), 2014.

[46] M. Nowak, C. Pautasso, Team Situational Awareness and Architectural Decision Making with the Software Architecture Warehouse, in: K. Drira (Ed.) Software Architecture, Springer Berlin Heidelberg, 2013, pp. 146-161.
[47] P. Gaubatz, I. Lytra, U. Zdun, Automatic enforcement of constraints in real-time collaborative architectural decision making, Journal of Systems and Software, 103 (2015) 128-149.
[48] I. Groher, R. Weinreich, Collecting Requirements and Ideas for Architectural Group Decision-Making Based on Four Approaches, in: Proceedings of the 9th European Conference on Software Architecture, 2015, pp. 181-192.
[49] M.A. Babar, B. Kitchenham, L. Zhu, I. Gorton, R. Jeffery, An empirical study of groupware support for distributed software architecture evaluation process, Journal of Systems and Software, 79 (2006) 912-925.
[50] T. Al-Naeem, I. Gorton, M.A. Babar, F. Rabhi, B. Benatallah, A quality-driven systematic approach for architecting distributed software applications, in: Proceedings of the International Conference on Software Engineering, ACM, New York, USA, 2005, pp. 244-253.
[51] A. Nakakawa, P. Bommel, E. Proper, Towards a Theory on Collaborative Decision Making in Enterprise Architecture, in: R. Winter, J.L. Zhao, S. Aier (Eds.) Global Perspectives on Design Science Research, Springer Berlin Heidelberg, 2010, pp. 538-541.
[52] K. Sousa, H. Mendonça, E. Furtado, Applying a multi-criteria approach for the selection of usability patterns in the development of DTV applications, in: Proceedings of VII Brazilian symposium on Human factors in computing systems, ACM, Natal, RN, Brazil, 2006, pp. 91-100.
[53] D. Falessi, G. Cantone, M. Becker, Documenting design decision rationale to improve individual and team design decision making: an experimental evaluation, in: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, ACM, 2006, pp. 134-143.
[54] K. Mohan, B. Ramesh, Traceability-based knowledge integration in group decision and negotiation activities, Decis. Support Syst., 43 (2007) 968-989.
[55] O. Zimmermann, J. Koehler, F. Leymann, R. Polley, N. Schuster, Managing architectural decision models with dependency relations, integrity constraints, and production rules, Journal of Systems and Software, 82 (2009) 1249-1267.
[56] A. Tang, Software designers, are you biased?, Proceeding of the 6th international workshop on, (2011) 1-8.
[57] R. Kazman, M. Barbacci, M. Klein, S. Jeromy Carriere, S.G. Woods, Experience with performing architecture tradeoff analysis, in: 21st International Conference on Software Engineering, IEEE, 1999, pp. 54-63.
[58] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord, B. Wood, Attribute-Driven Design (ADD), Version 2.0, (2006).