DecidArch v2: An improved Game to teach Architecture Design Decision Making

Remco C. de Boer
ArchiXL
The Netherlands
rdeboer@archixl.nl

Patricia Lago, Roberto Verdecchia
Vrije Universiteit Amsterdam
The Netherlands
{p.lago, r.verdecchia}@vu.nl

Philippe Kruchten
University of British Columbia
Canada
pbk@ece.ubc.ca

***Abstract*—TBA.**

***Index Terms*—software architecture, design decision making, game**

I. Introduction

In 2017 we have designed DecidArch, a software architecture card game, and used it in the 2017 Software Architecture course at the Vrije Universiteit Amsterdam. The game has been designed to support three learning objectives related to the design decision making process: (1) creating awareness about the rationale involved in making trade-offs and choosing design options among alternatives, (2) enabling the appreciation of the design decisions proposed by others, and (3) creating awareness about the dynamics of software architecture design. We have reported our experiences with the first iteration of this game in [1].

In the meantime, we have been working on a new version of DecidArch in which we have addressed the lessons learned from the first iteration. We have used the new version of the game in our 2018 Software Architecture course (i.e., with a different student population), and executed the same survey as in the previous year to measure the effects of our interventions.

In this paper, we describe how DecidArch supports the teaching of design decision making processes, the interventions we applied to the first version of the game and the effects thereof. We also present further lessons learned and an outlook into future work.

II. Related Works

As discussed in [1], the past 15 years witnessed a lot of interest in games to support software engineering education. These span from a large variety of digital games [2], to simpler board games or card games (e.g. [3], [4]). The latter proved to be a better fit by harnessing their social characteristics (e.g. face-to-face interaction) especially in teaching complex subjects that involve social aspects and that are hard to teach without simulation [2]. Such games, where the students play against or with each other rather than against or with a computer, have been used a.o., to illustrate architectural technical debt [5], to teach software project management [6], agile development [7], and process issues in software engineering [4].

To the best of our knowledge, there are two games that, similar to ours, target architecture design reasoning but have very different learning objectives: Tang et al [8] specifically focus on design reasoning and their cards acts as triggers for the students to explore the design space by applying the reasoning techniques suggested by the selected card; Cervantes et al [9] focus on teaching the architecture design process by using the Attribute-Driven Design method [10]. While their cards are much more sophisticated than ours (including technologies, tactics and patterns), they are specific to big data systems. Also, the authors aim to lead the players to make "smart decisions". We rather do not enter the merit of the quality of the decisions themselves, but focus on making students appreciate the dynamics of architecture design, which entails collaboration and consensus, reasoning and among alternative options.

It must be noted that the literature already reports on courses designed to at least teach some aspects of such dynamics. Lago and van Vliet [11], for instance, focused on aspects of software architecture where communication is key, like defining views that frame stakeholder concerns, trade-off analysis and architecture assessment. De Boer et al. [12] emphasized collaboration and built a course that would let the students act as architects and experience the multiple and often conflicting variables influencing architecture design decisions. The game we discuss here, DecidArch, differs by zooming into one single course topic, namely *architecture design decision making*, and leveraging the examples cited above by emphasizing collaboration (the students play our game as teams of architects) and bringing together various skills and competences (by applying theory to practice).

III. The DecidArch Game

The DecidArch game has been developed for the education of software architecture to master students [1]. Teaching software architecture - and especially the software architecture decision making process - is challenging, since it is difficult to let students experience meaningful design scenarios. The goal of the DecidArch game is to let players experience the concept of architectural design decision making by playing a simple card game.

During the game, the players collaboratively design a software system. They do so by addressing the stakeholder con-

cerns that are gradually revealed. While the design progresses, new concerns and unanticipated events may prompt the players to reconsider earlier design decisions. The players need to keep their stakeholders happy by addressing the required quality attributes of the system.

The game consists of four types of cards:

(a) The **Project card** contains a brief description of the project for which the players are designing a software system. It represents the context in which the decision making process takes place.

(b) A **Stakeholder card** describes one of the systems stakeholders. It lists the stakeholder's goal with respect to the system, and the quality attributes the stakeholder is concerned with. There are multiple stakeholder cards, each with their own list of quality attributes. This captures the fact that different stakeholders may be concerned with different quality attributes. Moreover, the individual stakeholder's 'level of importance' for each quality attribute is expressed on the stakeholder card as a QA-priority (where QA stands for 'quality attribute').

(c) A **Concern card** describes a concern and accompanying design options. Each Concern card contains several design options that can be used to address the concern. Each option has a different impact on quality attributes of the system: very negatively (–), negatively (-) neutral (=), positively (+), very positively (++).

(d) An **Event card** describes an event with lasting effects on the project. It describes some change for the project, that may lead to necessary changes in previous design decisions, or may influence decisions yet to be taken. The changes introduced by an Event card last the entire game.

At the start of the game, the Project card is revealed. Then, the players take clockwise turns until the end of the game. In every turn:

1) The current player draws a Concern card
2) Each player, independently, suggests a design option to address the concern on the Concern card.
3) The group, collaboratively, decides which design option is chosen to address the concern.

When every player has played one turn, the round ends and a new round starts. At the end of a round, the group draws an Event card and assesses the effect of the event on their current design. The players may need to revise some previous design decisions, and/or the event may put a constraint on future design options. The game ends after 30 minutes or when all Concern cards have been played, whichever occurs sooner. The game's time limit represents the time pressure under which architectural decision making usually has to operate.

To ensure that all players can equally contribute to the collaborative decision making, each player writes their personal suggestion (i.e., suggested option plus rationale why the option is suggested) on an individual *decision preparation template*. Only when all players have done so, the group collaboratively discusses the suggestions and decides upon the group position, which is then recorded on a collective *decision taking template*.

At the end of the game, the success of the group is determined by comparing the quality impact of their design decisions (expressed by the pluses and minuses of the chosen options on the Concern cards) with the quality requirements of the stakeholders (expressed by the QA-priorities on the Stakeholder cards). Only when all quality attributes have been sufficiently addressed and all stakeholders are satisfied with the quality impact of the design decisions, do the players win the game. Their final score is determined by the 'satisfaction level' of the stakeholders, which is based on the delta between the design decisions' quality impact and the stakeholders' quality requirements. If the players ran out of time, the number of remaining concern cards is applied as a penalty.

IV. Interventions

When we used the first version of our DecidArch game, we asked all participating students to fill out a survey, consisting of closed (Likert scale) questions with the option to add textual remarks. This allowed us to determine to what extent the game contributed to each of our learning objectives:

LO1-Reasoning Create awareness about the rationale involved in making trade-offs and choosing design options among alternatives.

LO2-Differences Enable the appreciation of the design decisions proposed by others.

LO3-Reconsideration Create awareness about the dynamics of software architecture design, especially about the relations between design decisions and the ripple effect of changes.

Our conclusion, reported in [1], was that the first version of DecidArch supported all three learning objectives, but that the support for LO3-Reconsideration depended too much on chance. We also elicited a need for a more balanced use of the decision templates, and obtained some remarks on the rules' clarity - especially the scoring sheet.

This prompted us to design a second version of the game, to which we applied several interventions as discussed in the following.

A. Redesign of event cards

Our main lesson learned from the previous game iteration was that we need players to run into enough events with enough potential impact, so that the chance they have to revisit earlier design decisions increases. To this end, we redesigned the event cards to make the impact of the event much more explicit. We separated (and in some cases, introduced) the event's *consequences*, which had previously been included in or implied from the event description.

For example, the first version of the game contained an event card "Local Fire" that had the description *"A local fire has occurred at one of your datacenters (if applicable). Consider what consequences are applicable to your design. Include any necessary changes to mitigate these consequences. (Related quality attribute: Availability)"*. In the new version,

this event card was changed to "Fire!", with as description: *"There has been a small fire in one of the Owner's office buildings. Luckily, the fire could be contained and the data center was not jeopardized. However, because of this incident a new policy is now in effect that prohibits the use of single local databases."*. An explicit description of the consequences (in terms of the game's elements) was added: *"Consequences: If you selected 'single local database' as design option for any of the concerns, those decisions need to be revised. For future decisions, this option is no longer available."*

B. Revision of QA-priorities and QA impacts

In addition to the redesign of the event cards, we revised the stakeholders' QA-Priorities and concern cards' quality impacts.

In the first version of the game, the stakeholders' QA-Priorities were initially set to 0 and could only change through events. Since the amount of event cards played in a game is limited, some groups ended the game with stakeholders that still had all QA priorities set to 0. In the new version the stakeholders have their initial QA-priorities set to higher numbers, fact that can still be affected by the events throughout the game.

We re-balanced the impact on quality of some concerns after remarks from the participating students. For some concerns, for instance, in the first version of the game a particular design option was always a clear winner. To solve this, we made sure that every design option entails a trade-off. We also ensured that the higher QA-Priorities of the stakeholders could be met.

C. Improved practical usefulness of the decision taking template

With the first version of the game, some players remarked that they didn't use the templates since writing down all decisions and rationale is costly without any perceived direct benefit. We increased the direct benefit by adding the QA impact of the chosen design option to the template. Through this change, the template is not only useful during the game but also afterwards when calculating the score.

D. Overall improvement of rulebook, scoring sheet, and card design

We restructured the rulebook and improved the layout of the rulebook and the playing cards. This provides the game material a much more 'professional' appearance and enhances the playability of the game. The addition of icons at the back of the cards, for example, makes it easier to distinguish between different card types. The new version of the rulebook has a structure that is more aligned to rulebooks of store bought games. The scoring of the game is better explained by means of an example, and the accompanying scoring sheet has been simplified.

V. Discussion of Results

For both versions of the game, we presented the game participants with the same survey. Figure 1 shows a comparison of the survey results from both years. The 'past version' of the game is the version that has been used in our 2017 Software Architecture course, and the 'current version' is the version from the 2018 course. The past version of the game has been played by 22 groups with a total of 83 players. The current version has been played by 20 groups with a total of 77 players. For both versions, all players were students of the VU Software Architecture course. The current version contains the changes discussed in Section IV.

Statement	Past version	Current version
S1.1	99%	97%
S1.2	79%	97%
S1.3	62%	66%
S1.4	75%	86%

TABLE I
Overall agreement with statements related to LO1-Reasoning

In Figure 1, the statements in row RQ1 test for the satisfaction of LO1-Reasoning, row RQ2 test for LO2-Differences, and row RQ3 for LO3-Reconsideration. The final row RQ4 contains statements that test the game's playability. The survey results are presented as diverging stacked bar charts. The colors in the bar chart represent the answer options: strongly disagree (dark red), disagree (red), neutral (grey), agree (green), strongly agree (dark green). The size of a colored segment corresponds to the percentage of participants that provided that answer. The percentages are reported below the bar.

Overall, a cursory glance at Figure 1 shows that the current version of the game receives more positive responses than the past version.

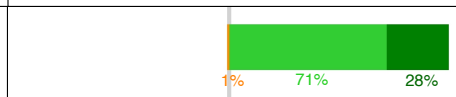
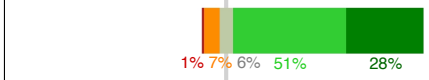
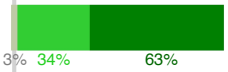
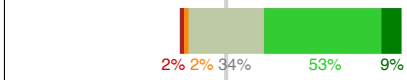
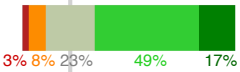
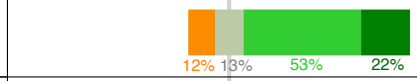
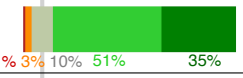
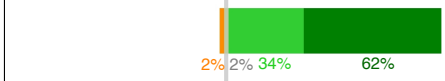
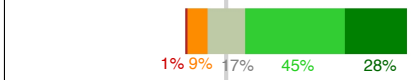
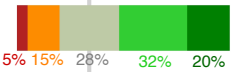
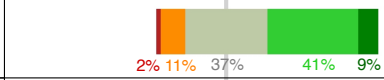
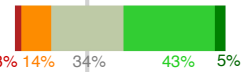
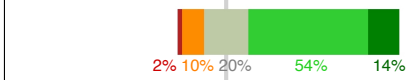
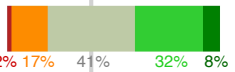
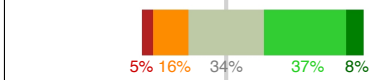
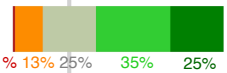
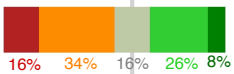
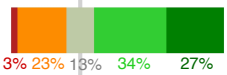
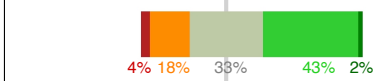
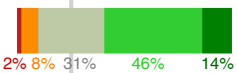
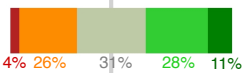
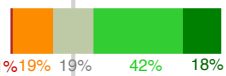
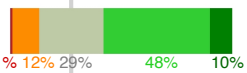
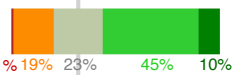
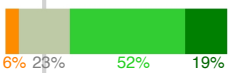
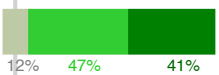
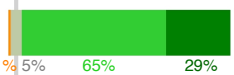
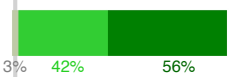
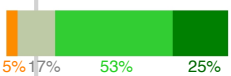
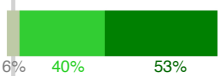
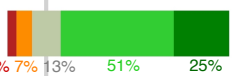
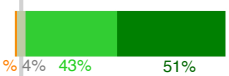
A. Effects on LO1-Reasoning

For LO1-Reasoning, the responses were already quite positive for the past version of the game. For the current version, we see a clear increase in 'Strong Agreement' with the survey statements. The most significant change in Strong Agreement can be seen for statement S1.2: *"To make design decisions, I had to consider trading a quality attribute for another"*. This appears to be one of the effects of our intervention in the quality attribute impacts of concern cards (cf. Section IV-B).

When we look at overall agreement (Strongly Agree and Agree) for the four statements, there's a clear positive shift as well (see Table I). Especially for the current version, however, the overall agreement with S1.3 is considerably lower than agreement with the other three statements. Reasons that participants gave for disagreeing with this statement include:

- A lack of time to properly document the design decision rationale;
- Being fully focused on optimizing the "+" and "-" impact of design options on quality attributes.

The latter may be an unanticipated consequence of the intervention described in Section IV-B; now that the QA-impacts and trade-offs are made more explicit, the game's model is more apparent to the players. It is interesting that these players apparently did not consider the "+" and "-"

RQ	Statement	Past version	Current version
RQ1	S1.1: *I did reflect on how a design option could impact the stakeholder's concerns.*		
	S1.2: *To make design decisions, I had to consider trading a quality attribute for another.*		
	S1.3: *The templates motivated me to document the reasons behind the chosen design options.*		
	S1.4 *The quality impacts on the concern cards motivated me to explicitly reason about trade-offs in quality.*		
RQ2	S2.1: *To decide on the best design option, I discussed design ideas with the other players.*		
	S2.2: *The differences (e.g. in experience) between the players led to discussions about different design ideas.*		
	S2.3: *The use of the decision preparation template gave me an opportunity to contribute new design ideas.*		
RQ3	S3.1: *When changing design decisions, I maintained the consistency among design decisions.*		
	S3.2: *As the design progressed, changing a design decision required me to use an increasing amount of time to maintain consistency between design decisions.*		
	S3.3: *During the game, we had to change past decisions.*		
	S3.4: *The templates helped me to revisit design decisions to maintain consistency among changed design decisions.*		
	S3.5: *When addressing a new concern, the templates motivated me to revisit earlier design decisions.*		
	S3.6:*The game helped me maintain the consistency among the design decisions.*		
RQ4	S4.1: *The game was fun.*		
	S4.2: *The cards were easy to understand.*		
	S4.3: *The rules were clear.*		
	S4.4: *The scoring sheet was clear*		

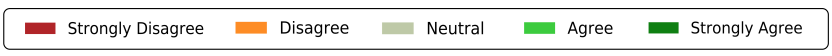


Fig. 1. Comparison of survey results for the two versions of the game

Statement	Past version	Current version
S2.1	96%	95%
S2.2	73%	52%
S2.3	50%	48%

TABLE II
OVERALL AGREEMENT WITH STATEMENTS RELATED TO LO2-DIFFERENCES

Statement	Past version	Current version
S2.1	2%	0%
S2.2	10%	20%
S2.3	13%	17%

TABLE III
OVERALL DISAGREEMENT WITH STATEMENTS RELATED TO LO2-DIFFERENCES

Statement	Past version	Current version
S3.1	68%	40%
S3.2	45%	60%
S3.3	34%	61%
S3.4	45%	60%
S3.5	39%	60%
S3.6	58%	55%

TABLE IV
OVERALL AGREEMENT WITH STATEMENTS RELATED TO LO3-RECONSIDERATION

Statement	Past version	Current version
S4.1	71%	88%
S4.2	94%	98%
S4.3	78%	93%
S4.4	76%	94%

TABLE V
OVERALL AGREEMENT WITH STATEMENTS RELATED TO PLAYABILITY

impact of the design options part of their design rationale. To them, it seems, it was more a part of 'winning the game'.

B. Effects on LO2-Differences

A closer look at the statements related to LO2-Differences shows that the overall agreement with these statements has diminished (see Table II). The overall disagreement (Table III) has increased, except for statement S2.1. For this statement, the differences can be explained by the increased amount of respondents that answered 'Neutral' (from 2% to 5%). Two participants clarified their choice for 'Neutral'. One of them remarked that "We always agreed". The other indicated that "Limited time makes it about the pluses and minuses without discussing".

Clarifications for disagreement with or neutral scoring of statement S2.2 include, as for S1.3, a lack of time to have extensive discussions and being primarily focused on the game result. As for S2.1, some players remarked that in their team there was little disagreement about the option to choose. Finally, some respondents indicated they did not know the background of their fellow players, which led them to disagree with this statement.

For S2.3, finally, time pressure was again a major contributing factor for disagreeing. For efficiency reasons, these players bypassed the template and immediately discussed their design ideas without first writing them down.

C. Effects on LO3-Reconsideration

Most of the statements related to LO3-Reconsiderations received more positive feedback for the current version of the game. Exceptions are statements S3.1 and S3.6 (see Table IV). Most notably, the scores for statement S3.3 *"During the game, we had to change past decisions"* have strongly increased. There's a substantial decrease in (strong) disagreement. This is an important result, since having to change past decisions is an essential factor in creating awareness about relations between design decisions and ripple effects.

Remarks for statement S3.1, unfortunately, provide little insight into the reason of the higher overall disagreement. With respect to statement S3.6, players remarked that the game did not provide any restrictions on choosing conflicting design options. On the contrary, some teams felt the need to select options they knew or suspected to be conflicting, simply because that combination had the highest impact on the quality attributes they had to achieve.

D. Effects on playability

The overall agreement with statements related to the game's playability has increased to an almost perfect level (see Table V). We can only conclude that our interventions had the desired effect.

VI. LESSONS LEARNED AND FUTURE WORK

A lack of time is a recurring theme throughout the feedback we received in the survey. Note, by the way, that for the past version of the game the time limit was also 30 minutes, but while students were playing we decided on-the-fly to increase the play time to 45 minutes as we noticed that many teams did not finish their first round and hence had not even played their first event card. For the current version, the improved rulebook and game design meant players could be up to speed sooner, so we decided *a priori* to stick to the game's time limit of 30 minutes. In an earlier test run of the game with professional architects (cf. [1]) such time resulted more than enough. For novice / student architects, clearly not. We will update the game rules and extend the game time to 60 minutes. We expect this to solve many of the current version's remaining issues.

What is also apparent from the survey results is that if you ask people to optimize for certain parameters, that's what they will do. Even though we ask the players to maintain design decision consistency, the game's final score does not take consistency into account. Consequently, players choose to ignore this aspect: they want to win the game, and the final score is all that counts. In a next (and probably final) version of the game, we aim to make the dependencies between design options for different concerns explicit, in particular relations that express conflicts and prerequisites. In this way, inconsistencies between chosen design options will become a more apparent part of the game, which can stimulate the players to address and remove them.

Even though there are still some improvements possible, we are confident that the current version game has significantly improved over the previous version.

VII. Conclusions

Our interventions had a clear positive effect on the playability of the game, and the game's support for LO1-Reasoning. The overall effect on LO3-Reconsideration was also positive. Remaining issues (for LO2-Differences and LO3-Reconsideration) have to be attributed to the allotted time for the game, the lack of explicit relations between design options for different concerns, and the game's inherent focus on optimizing individual design decisions for their impact on quality attributes. We intend to alleviate these issues by extending the game's playing time and by introducing explicit dependency relations between design options.

We expect the game in its current form to be playable in other contexts like other universities' software architecture courses, and in-company awareness training for non-architects. We will, however, apply the discussed improvements and use DecidArch v3 in a series of sessions in both context types.

Acknowledgment

We thank Jia F. Cai for contributing to the first version of the DecidArch game during his Master Thesis.

References

[1] P. Lago, J. Cai, R. de Boer, P. Kruchten, and R. Verdecchia, "Decidarch: Playing cards as software architects," in *52nd Hawaii International Conference on System Sciences (HICSS)*, 2019.

[2] M. Kosa, M. Yilmaz, R. V. O'Connor, and P. M. Clarke, "Software engineering education and games: A systematic literature review," *Journal of Universal Computer Science*, vol. 22, no. 12, pp. 1558–1574, 2016.

[3] G. Taran, "Using games in software engineering education to teach risk management," in *20th Conference on Software Engineering Education Training*, pp. 211–220, 2007.

[4] A. Baker, E. Oh Navarro, and A. van der Hoek, "An experimental card game for teaching software engineering processes," *Journal of Systems and Software*, vol. 75, no. 1-2, pp. 3–16, 2005.

[5] T. Grant, "Dice of debt game." www.agilealliance.org/dice-of-debt-game. Accessed on: 2018-03-13.

[6] P. Kruchten and J. King, "Mission to Mars: An agile release planning game," in *24th Conference on Software Engineering Education and Training (CSEET)*, pp. 552–552, 2011.

[7] Agile42, "Kanban pizza game." www.agile42.com/en/training/kanban-pizza-game. Accessed: 2018-03-13.

[8] A. Tang, F. Bex, C. Schriek, and J. M. E. M. van der Werf, "Improving software design reasoning–a reminder card approach," *The Journal of systems and software*, vol. 144, pp. 22–40, Oct. 2018.

[9] H. Cervantes, S. Haziyev, O. Hrytsay, and R. Kazman, "Smart decisions: An architectural design game," in *Proceedings of the 38th International Conference on Software Engineering Companion (SEET Track)*, ICSE, pp. 327–335, ACM, 2016.

[10] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord, and B. Wood, "Attribute-driven design (ADD), version 2.0," tech. rep., 2006. CMU/SEI-2006-TR-023.

[11] P. Lago and H. van Vliet, "Teaching a course on software architecture," in *18th Conference on Software Engineering Education Training (CSEET)*, pp. 35–42, 2005.

[12] R. de Boer, R. Farenhorst, and H. van Vliet, "A community of learners approach to software architecture education," in *Conf. on Software Engineering Education & Training*, pp. 190–197, 2009.