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—We report on the use of our DecidArch game to teach software architecture design decision making in two consecutive years. We compare the support of three learning goals for the first version of the game with the second, revised version. Results show how the game has clearly improved. For the remaining issues, we suggest final improvements.

Index Terms—software architecture, decision making, game

I. INTRODUCTION

In 2017 we designed DecidArch, a software architecture card game, and used it in the Software Architecture course at the Vrije Universiteit Amsterdam. The game supports 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 version of this game in [1].

In the meantime, we addressed the lessons learned from the first version in a new version of the game. We used this new version [2] in the 2018 course, and used the same survey as in the previous year to measure the effects of our interventions.

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

II. RELATED WORKS

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 [3] specifically focus on design reasoning and their cards act as triggers for the students to explore the design space by applying the reasoning techniques suggested by the selected card; Cervantes et al [4] focus on teaching the architecture design process by using the Attribute-Driven Design method [5]. 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, consensus, and reasoning among alternatives.

III. THE DECIDARCH GAME

The goal of the DecidArch game is to let players experience the concept of architectural design decision making by playing a simple card game. The players work together as a team of software architects to collaboratively design a software system. They do so by addressing the stakeholder concerns that are gradually revealed. While the design progresses, new concerns and unanticipated events may force the players to reconsider earlier design decisions.

The game consists of four types of cards (see Fig. 1 for an example of each):

(a) The **Project card** contains a brief description of the project for which the players design a software system.

(b) A **Stakeholder card** describes one of the systems stakeholders. There are multiple stakeholder cards, each with their own list of quality attributes. A stakeholder's 'level of importance' for each quality attribute is expressed on the card as a Quality Attribute priority (QA-priority).

(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. Changes introduced by an Event card last the entire game.

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, 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. The time limit represents the time pressure

Fig. 1. Examples of each of the four card types.

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 plus rationale why that 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 group's success is determined by comparing the quality impact of their design decisions (i.e., the pluses and minuses of the chosen options on the Concern cards) with the quality requirements of the stakeholders (i.e., the QA-priorities on the Stakeholder cards, possibly modified by events). If the players ran out of time, the number of remaining concern cards is applied as a penalty.

IV. INTERVENTIONS

For the first version of the 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 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. We created a second version of the game, to which we applied several interventions as discussed below.

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 (see also Fig. 1), 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."* The third design option on the Concern Card depicted in Fig. 1 is an example of an option that is no longer available after this event. When that concern is addressed later in the game, the players can select from the remaining two design options. However, when the concern card has been played before the event occurs, the players need to revisit their previous decision and change it to one of the

remaining options in case they selected the option to use a single local database.

B. Revision of QA-priorities and QA 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 initial QA-priorities set to higher numbers, which can still be affected by 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 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. Improved 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 structure of the rulebook 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 2 shows a comparison of the survey results from both years. The 'past version' of the game is the version that has been used in 2017, 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. In both cases, participation in the game was compulsory but the outcome of the game did not affect the students' course grades in any way. The current version of the game contains the changes discussed in Section IV.

In Figure 2, the statements in rows RQ1, RQ2 and RQ3 test for the satisfaction of LO1-Reasoning, LO2-Differences, and LO3-Reconsideration respectively. 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.

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 can be seen for statement S1.2. This appears to be an effect 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. Especially for the current version, however, the overall agreement with S1.3 is considerably lower than agreement with the other three statements. Reasons provided for disagreeing with this statement include a lack of time to properly document the design decision rationale, and 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 "-" 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

The overall agreement with statements related to LO2-Differences has diminished. The overall disagreement has increased except for S2.1, which can be explained by the increase of 'Neutral' answers. 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".

Remarks for S2.2 include 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.

For S2.3, 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 S3.1 and S3.6. Most notably, the scores for statement S3.3 have strongly increased. There's a substantial decrease in (strong) disagreement. This is an important result, since having to change past decisions is an

RQ	Statement	Past version	Current version
RQ1	S1.1: *I did reflect on how a design option could impact the stakeholder's concerns.*	1% 71% 28%	3% 45% 52%
	S1.2: *To make design decisions, I had to consider trading a quality attribute for another.*	1% 7% 6% 51% 28%	3% 34% 63%
	S1.3: *The templates motivated me to document the reasons behind the chosen design options.*	2% 2% 34% 53% 9%	3% 8% 23% 49% 17%
	S1.4 *The quality impacts on the concern cards motivated me to explicitly reason about trade-offs in quality.*	12% 13% 53% 22%	1% 3% 10% 51% 35%
RQ2	S2.1: *To decide on the best design option, I discussed design ideas with the other players.*	2% 2% 34% 62%	5% 31% 64%
	S2.2: *The differences (e.g. in experience) between the players led to discussions about different design ideas.*	1% 9% 17% 45% 28%	5% 15% 28% 32% 20%
	S2.3: *The use of the decision preparation template gave me an opportunity to contribute new design ideas.*	2% 11% 37% 41% 9%	3% 14% 34% 43% 5%
RQ3	S3.1: *When changing design decisions, I maintained the consistency among design decisions.*	2% 10% 20% 54% 14%	2% 17% 41% 32% 8%
	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.*	5% 16% 34% 37% 8%	1% 13% 25% 35% 25%
	S3.3: *During the game, we had to change past decisions.*	16% 34% 16% 26% 8%	3% 23% 13% 34% 27%
	S3.4: *The templates helped me to revisit design decisions to maintain consistency among changed design decisions.*	4% 18% 33% 43% 2%	2% 8% 31% 46% 14%
	S3.5: *When addressing a new concern, the templates motivated me to revisit earlier design decisions.*	4% 26% 31% 28% 11%	1% 19% 19% 42% 18%
	S3.6: *The game helped me maintain the consistency among the design decisions.*	1% 12% 29% 48% 10%	1% 19% 23% 45% 10%
RQ4	S4.1: *The game was fun.*	6% 23% 52% 19%	12% 47% 41%
	S4.2: *The cards were easy to understand.*	1% 5% 65% 29%	3% 42% 56%
	S4.3: *The rules were clear.*	5% 17% 53% 25%	6% 40% 53%
	S4.4: *The scoring sheet was clear*	4% 7% 13% 51% 25%	1% 4% 43% 51%

Legend: ■ Strongly Disagree ■ Disagree ■ Neutral ■ Agree ■ Strongly Agree
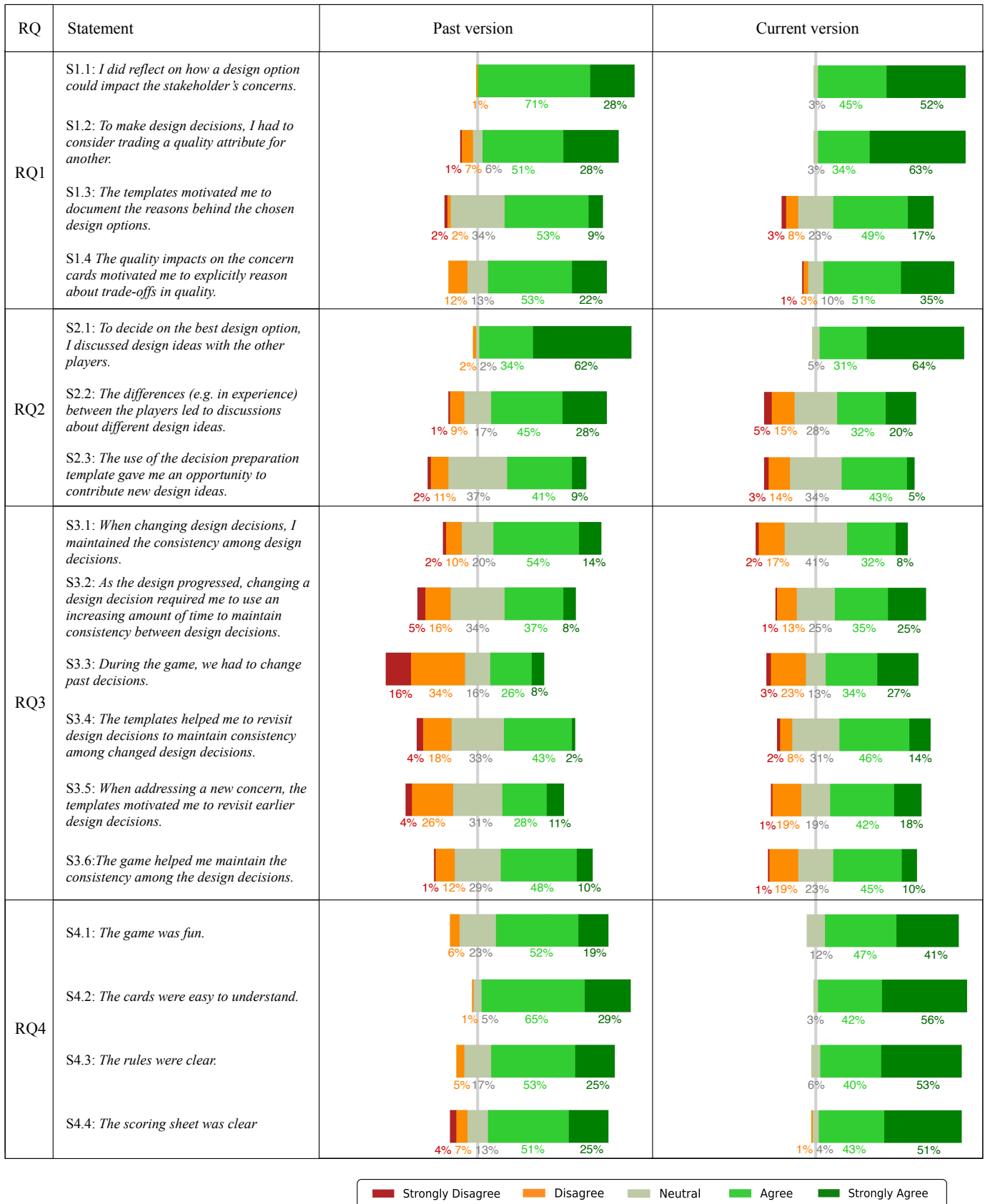
Fig. 2. Comparison of survey results for the two versions of the game

essential factor in creating awareness about relations between design decisions and ripple effects.

Remarks for S3.1, unfortunately, provide little insight into the reason for the higher overall disagreement. With respect to 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 required quality attributes.

D. Effects on playability

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

VI. LESSONS LEARNED AND FUTURE WORK

We have now used our game in two consecutive years, and intend to continue the use of DecidArch as part of the VU Software Architecture course. The overall feedback we receive from students is positive: they like the game. Moreover, the survey results indicate that the game helps to get across aspects of design decision making that are difficult to learn only from theory. While it will always remain a proxy, the game allows students to experience the design decision making process first-hand.

The survey results also point to areas of further improvement. A lack of time is a recurring theme throughout the feedback we received in the survey. 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 30 minutes limit. In an earlier test run of the game with professional architects (cf. [1]) this was more than enough time. For novices / students, clearly not. We will update the rules and extend the limit 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

We are confident that the DecidArch game, especially when the improvements above have been applied, has sufficiently proven itself to be a worthwhile addition to a Software Architecture course. The game materials are readily available,

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 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. so the game could also be used at other universities. Moreover, we see potential for the game to be used in-company awareness training. Practicing software architects will have experienced most of the game's learning objectives in practice; they are not our primary target audience. Their non-architect colleagues, however, could use the game to gain a better understanding and awareness of the intricacies of the architecture decision making process.

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 plan to apply the discussed improvements and use DecidArch v3 in a series of sessions in both university courses and in-company awareness training for non-architects.

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] R. C. de Boer, P. Lago, R. Verdecchia, and P. Kruchten, "DecidArch V.2 Replication Package," https://github.com/DecidArch/DecidarchV2.

[3] 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.

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

[5] 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.