

Modifiers for Quality Assurance in Group Facilitation

G. L. Kolfshoten · P. Grünbacher · R. O. Briggs

Published online: 23 February 2011
© Springer Science+Business Media B.V. 2011

Abstract A key task of a professional facilitator is to assure the quality of the knowledge products created through collaborative effort. To manage the quality of the knowledge a group generates, facilitators attend to, judge, and question the quality of the contributions a group makes, the decisions it makes and the commitments its members make toward achieving the group's goals. When facilitators or group members detect deficiencies in ideas, decisions, agreements, or artifacts, facilitators may need to intervene to support the group in improving the quality of their output, without disrupting the flow of the group process. In this paper we present a framework for quality assessment and a toolbox with flexible interventions that can be added to a collaborative process on the fly as soon as quality deficiencies are detected. The toolbox is a set of conditional adjustment interventions that a facilitator can add to other facilitation techniques so as to guard the quality of the group's intellectual products. The toolbox can also be used as a starting point for designing intelligent agents that support the automatic detection of quality deficiencies.

G. L. Kolfshoten

Department of System Engineering, Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands
e-mail: G.L.Kolfshoten@tudelft.nl

P. Grünbacher

Johannes Kepler University, Linz, Austria
e-mail: paul.gruenbacher@jku.at

R. O. Briggs (✉)

Department of Business administration, Institute for Collaboration Science, University of Nebraska at Omaha, Roskens Hall Room 512B, Omaha, NE 68182, USA
e-mail: RBriggs@mail.unomaha.edu

Keywords Facilitation · Quality assurance · ThinkLets · Collaboration engineering · Modifiers

1 Introduction

Collaboration is critical to the ongoing success of many organizations in the knowledge economy (Leibold et al. 2005), because many organizational tasks are now sufficiently complex that cross-functional teams with diverse expertise outperform conventional organizational forms based on functional homogeneity (Higgs et al. 2005; Troy et al. 2008). Such collaboration, however, can be challenging. Ambiguous goals, poor communication, inefficient group processes, conflicts of interest, and a host of other issues can limit group effectiveness (Levi 2007). For high-value tasks, therefore, groups sometimes turn to collaboration technology and professional facilitators to improve the effectiveness and efficiency of their joint efforts (Fjermestad and Hiltz 2001; Thomas et al. 2007). A facilitator is a group process expert who designs and conducts collaborative work practices on behalf of a group. While leading groups, facilitators monitor for and intervene to improve issues of communication, reasoning, information access, goal congruence, and distraction.

In order to help a group achieve its goals, a facilitator usually designs a sequence of activities, sometimes supported by collaboration tools such as Group Support Systems (GSS), shared document editors, audio-video conferencing, or shared document repositories. Facilitators instruct the group on how perform each activity. They guide discussions, and capture artifacts to document the group's ideas, decisions, and agreements. They must continuously evaluate the collaboration process to assure it moves the group toward its goals. Group goal attainment can be characterized in various ways, among them the quantity and quality of the products a group creates (e.g. Reinig et al. 2007) and the satisfaction of its members with group processes and outcomes (e.g. Reinig 2003). One of the key tasks of a facilitator is, therefore, to assure the quality of the knowledge products created by the groups they support (Kolfschoten and Rouwette 2006).

Grünbacher et al. (2004a) argue that quality assurance can be performed during execution of a group process. That, however, presents a facilitator with additional challenges. For instance, the dynamic nature of group processes can make it difficult to perform rigorous quality analyses on the fly. Also, to assure the quality of group products, the facilitator must attend to and track the group's communication, and judge the quality of the ideas group members generate and the decisions they take. This will impose additional cognitive load on the facilitator, whose job already places high demands on attention resources. Further, facilitators are often not experts in the domains of the groups they lead, and so it may be difficult for them to distinguish a more-useful from a less-useful contribution. It would therefore be useful during the execution of a group process if a facilitator could deploy interventions that bring the attention resources and expertise of the group members themselves to bear on the challenge of quality assurance. In this paper, we present a framework for quality assessment and a toolbox with flexible interventions that can be added to existing process designs on the fly as soon as quality deficiencies are detected, without disrupting

the flow of the process. The toolbox is a set of conditional adjustment interventions, which can be used to extend existing facilitation techniques to guard the quality of group products. Grünbacher et al. (2004a,b) also argue that quality assurance can be addressed at the time a facilitator designs a group process. In this paper, we present specific quality assurance techniques that facilitators can add to their existing facilitation at design time, so that quality assurance will be built into the group process from the beginning.

We proceed with presenting background information on facilitation techniques and quality of knowledge or information. From this literature we propose a framework for quality assessment and a framework for interventions. Next, we present the toolbox for quality assurance. Finally, we discuss implementation and implications, and offer suggestions for future research.

2 Quality Goals of Collaboratively Generated Knowledge

Depending on the domain of interest different quality dimensions need to be considered. For instance, Wand and Wang (1996) and Wang et al. (1995) performed an analysis on data quality dimensions and proposed the following quality dimensions: Accuracy and Precision, Reliability, Timeliness and Currency, Completeness, Consistency and Relevance. Another frequently used set of quality dimensions is called the 3C's = correctness, completeness, consistency. Both examples show some overlap, and we will discuss each of these dimensions to determine which ones are most appropriate to qualify collaborative outcomes. The relative importance of different quality dimensions will always depend on the specific goals of a collaborative effort. As described in the Goal-Question-Metric approach (Basili et al. 1994), quality dimensions should be operationalized by defining a set of metrics that are established based on a goal and precise questions to derive performance indicators for that goal.

Correctness means the correspondence of a group's output with fact or truth. Correctness shows some overlap with both accuracy and precision. They indicate the quality of the information in terms of truthfulness, or the degree to which data actually reflect the reality they model. Precision means that the information is formulated in sufficient detail to ensure that it is well understood. Accuracy and Correctness can be seen as synonyms of truthfulness from a rationale perspective. Reliability has to do with the probability of errors in the data. When knowledge is created in collaborative settings it has to do with the trust in the expertise, ability or sincerity of the contributors. Reliability can be seen as a sub-dimension of correctness from a more subjective perspective. Timeliness and Currency are less relevant in a collaborative setting when the knowledge is assessed at the moment it is created. However, as for reliability, people can question whether the knowledge brought into the meeting is accurate and up-to-date.

Correctness is an important quality factor in collaborative outcomes. In order for knowledge to be regarded as a group result it should be created in shared understanding. When the knowledge created is formulated correctly, it is more likely that it is understood by the group members and subsequent consumers of the group's output. When information is incorrect, or incorrectly interpreted it might cause misunder-

ing and incorrect decisions. We define correctness as conforming to fact or common understanding or in accordance with an accepted standard.

Completeness, i.e., having everything that is needed, is an important factor. In complex, knowledge intensive tasks, however, completeness is also linked to parsimoniousness as the quest for completeness can lead to information overload which requires increased cognitive effort to converge into meaningful knowledge. In some cases completeness is less relevant and time-consuming. Completeness can be assessed at two levels, completeness of individual contributions and completeness of the resulting group outcome, a set of contributions. We define completeness as containing all necessary elements. We define parsimoniousness as containing only necessary, non superfluous elements.

Consistency refers to an absence of contradictions and conflicting information in the data set or with respect to information quality requirements. Consistency is especially important when concepts are to be compared or evaluated. For instance, when group members need to compare the feasibility of alternative solutions they need to be described at the same level of detail. In case problems are to be assessed no solutions should appear in the list of problems. We define consistency as the correspondence and compatibility among related aspects.

Relevance is the pertinence of the group's contributions to the matter of the collaborative effort. It is important as a collaborative effort focuses on a specific scope and resources should not be wasted on contributions that are outside this scope. For instance, a team should not waste time solving problems that are beyond the responsibility of the team. We define relevance as instrumental to achieving the goal.

Given the analysis above we consider the following quality dimensions as essential for characterizing the quality of the outcomes of knowledge intensive collaboration:

- Correctness
- Consistency
- Completeness
- Parsimoniousness
- Relevance

The set of dimensions is not necessarily complete and the emphasis may be put on different quality dimensions in specific domains. However, this set offers a basis to identify a set of key facilitation interventions for quality assurance. We list these quality dimensions and their definitions in Table 1.

3 Facilitation Interventions

While facilitation has long been regarded as an art and skill learned through experience and apprenticeship, recently facilitation skills and methods have become more transferable in the field of Collaboration Engineering, an approach to design and deploy collaboration processes in organization (Briggs et al. 2003; Vreede and Briggs 2005; Vreede et al. 2009). In Collaboration Engineering, processes are designed as a sequence of activities consisting of specific predictable facilitation interventions to guide the group through this process and to structure and focus their effort to goal

Table 1 Definitions of quality dimensions

Quality dimension	Definition
Correctness	The extent to which contributions are conforming to fact or common understanding or in accordance with an accepted standard
Consistency	The correspondence and compatibility among related aspects
Completeness	The extent to which contributions are covering or containing all necessary elements
Parsimoniousness	The extent to which contributions are containing only necessary, non superfluous elements
Relevance	The extent to which contributions are instrumental to achieving the goal

achievement. Facilitation techniques in Collaboration Engineering are documented as design patterns called thinkLets.

In this section we will first explain the ThinkLet concept, which provides a framework for instruction in group facilitation. ThinkLets allow capturing facilitation interventions. We will explain in detail how a facilitation intervention works as a way of communicating instructions, and we will present a communication model to describe this instruction.

3.1 ThinkLets

ThinkLets are reusable facilitation techniques realized as named, scripted procedures that invoke known patterns of collaboration (e.g. generate, reduce, clarify, organize, evaluate, build consensus) among people working together toward a goal (Kolfschoten et al. 2006; Vreede et al. 2006). ThinkLets are logical design elements for collaborative work practices. The essence of a ThinkLet is embodied in a set of rules that specify *actions* that people in particular *roles* should take under certain *constraints*, using certain *capabilities* (Kolfschoten et al. 2006; Vreede et al. 2006). Actions may include, for example, adding, editing, associating, judging, or deleting ideas. ThinkLets typically include a moderator role and a participant role, but a given ThinkLet may also include more specific roles such as recorder or devil's advocate. Osborn (1953) famous brainstorming technique could be expressed in ThinkLet rules as follows:

1. *Participants* (role) shall *add* (action) ideas to an *oral channel* (capability) under the following constraints: (a) ideas shall respond to the brainstorming question; (b) participants shall not withhold, but shall contribute bad, outlandish, unconventional ideas along with ideas they believe to be good; (c) participants shall refrain from criticizing ideas contributed by others;
2. *Moderator* (role) shall *add* (action) ideas from the *oral channel* (capability) to a *page that all participants can see* (capability) under the following constraints: (a) moderator shall assure that all participants gain equal access to the oral channel; (b) moderator shall not screen any contributions, but shall record all of them on the publicly viewable page.

The documentation of a ThinkLet includes a generic script that realizes the ThinkLet's rules as a set of things a facilitator should say and do to lead a group through the

activity. A facilitator typically modifies that script at process design time, tailoring it to the task at hand. In this way a facilitator can communicate interventions to the group based on a predictable script. An example of a ThinkLet can be found in (Kolfschoten et al. 2006; Vreede et al. 2006).

3.2 Intervention as a Way of Communicating

ThinkLets and other facilitation techniques are, in essence, a structured form of communication. It consists of a message that is transmitted with use of a channel/vehicle from a sender (facilitator) who can encode the message and send it to a receiver (participant/group) who can decode this message. From this decoding effort, a meaning can be distilled and in response an action can be performed. A response can be either in terms of feedback, i.e. the transmission of a return message, or a communication effect, i.e. any other action in response to the message (Krone et al. 1987) which will constitute a pattern of collaboration, and a specific result. In such a generic system there is no predictability of the action as an effect of the message or its content, to increase predictability several conditions should be met.

A basic set of pre-conditions and circumstances must exist in order for such communication to invoke actions that are consistent with the intention of the facilitator. For example, an absence of noise that could diminish understanding of a transmission, clear, unambiguous instructions that participants can readily understand, and participants must have the capacity to execute the action instructed by the facilitator. Further, there is a need for some level of goal congruence where the intentions of the facilitator do not conflict with the interests of the group members (Briggs 1994; Briggs et al. 2004). There is also a need for some level of trust where the participants believe in the truthfulness and benevolent intent of the facilitator. Lacking these conditions, the group may not accept instructions from a facilitator.

When requisite preconditions are met, however, a facilitator's instruction for a specific action is likely to produce in group members an intention to execute the requested action, and consequently the intended pattern of collaboration would result (Bragge et al. 2005). If, however, instructions are not successfully communicated, the response action may not fully meet the intentions of the facilitator. It is therefore important to find a way to describe requested actions, constraints, and capabilities, with sufficient completeness and specificity to produce desired effects predictably and repeatability.

3.3 The Structure of Facilitation Interventions

To realize an intention by means of intervention, two types of interventions are required (Badler et al. 1998). First, there are static interventions in which one or more commands are given to initiate the key activities of a process. For the balance of this paper we will refer to this kind of communication as an *instruction intervention*, the process which is illustrated in Fig. 1. Second, there are dynamic interventions intended to adjust the actions performed by the group to resolve a discrepancy between the facilitator's intentions and the group's actions. These interventions depend on emergent conditions. Different interventions are required for different types of discrepancies.

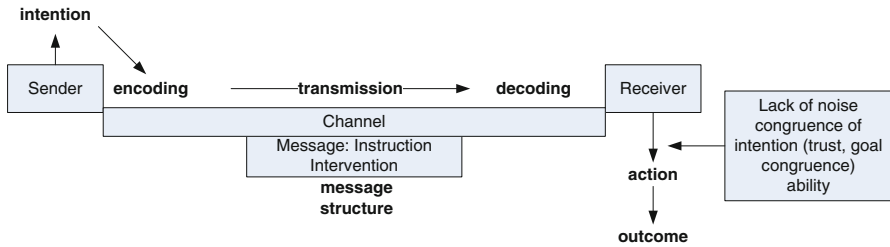


Fig. 1 The process of an instruction intervention

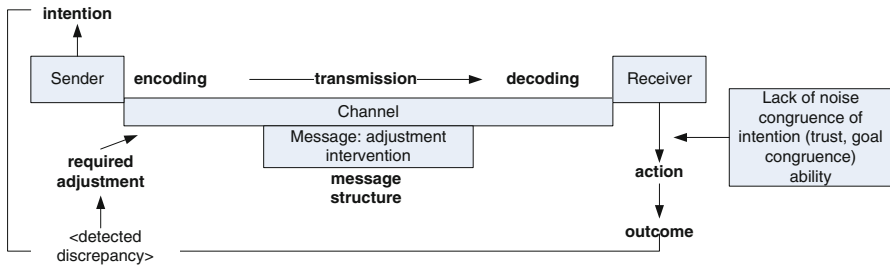


Fig. 2 The process of an adjustment intervention

For instance, if the action of the group is to add solutions, the group can diverge from the intention of the instruction in that the solutions they add are not specific enough or that the solutions do not address the problem at hand. In the first case, an intervention should be executed to evoke more specificity of the solutions. In the second case, a message is required to create more shared understanding of the problem. We will call these messages *adjustment interventions*. Adjustment interventions are thus instructions of the practitioner with the intention to adjust the actions of the group members in order to ensure that the outcome of these actions meet the intention of the process prescription; the intended pattern of collaboration and the intended result. The resulting framework is illustrated in Fig. 2.

To increase the predictability of the facilitation intervention, its messages should be encoded in a way that minimizes the chance of errors in decoding by participants. For this purpose we want the instruction to be precise and complete, but it should not be too complex or cause an information overload. In order to derive a template for the structure of facilitator interventions, we examined the production systems and representation systems from the artificial intelligence literature.

The artificial intelligence literature proposes several ways of mimicking human behavior (Badler et al. 1998; Newell and Simon 1972; Rich 1983; Winston 1984; Zhang 1997). Instructions can be framed as conditional actions (if condition applies, carry out action), as problem solving (if problem occurs, search for action that makes the problem go away), or as goal satisfaction (see how a desired state can be achieved from the current state using the production rules). The structure of ThinkLets as a collection of rules is similar to that of parameterized natural language action representations described by Badler et al. (1998, 1999) to specify the actions of virtual humans

(avatars) in natural language in a parameterized action representation. The conceptual design of a ThinkLet exists in a set of rules (Kolfschoten et al. 2006; Vreede et al. 2006). These rules are similar to rules mimicking human behavior in avatars (Badler et al. 1998). Each rule describes for a role an action that needs to be performed using a capability under some set of constraints to restrict those actions. Further, some ThinkLets include conditional rules for frequently-required adjustment interventions because specific discrepancies manifest predictably during the execution of an activity based on the ThinkLet (Kolfschoten and Houten 2007). These interventions, however, are local and internal to that ThinkLet. Other adjustment interventions however have been codified as reusable modifiers that can be applied across a collection of ThinkLets to address quality assurance issues that commonly manifest in a wide variety of ThinkLets.

4 Modifiers

Modifiers are named, documented, reusable variations that can be applied to a number of ThinkLets to create predictable change or variation in the patterns of collaboration invoked by these techniques. These variations are created by adding rules to a ThinkLet, or by replacing existing ThinkLet rules with different rules. The same modifier can be used to create a variation on ThinkLets that produces different patterns of collaboration. Several authors have shown that such variations can create significant effects on the quality of the outcomes of a collaborative effort (Kolfschoten and Santanen 2007; Santanen and Vreede 2004; Shepherd et al. 1996). By using modifiers, we can add nuance to a set of basic design patterns without suffering a combinatorial explosion of the pattern language. When modifiers are distilled from a set of design patterns, the pattern language can become richer, as more combinations can be made from fewer elements. At the same time, the pattern language becomes more concise as the number of concepts in the pattern language becomes smaller.

In one sense, a modifier is to a ThinkLet as a virus is to a cell. A virus, invokes predictable changes on the way the cell performs. In a like manner, modifiers can be applied to ThinkLets to create predictable *changes* in the patterns of collaboration the ThinkLet invokes. An example of a modifier is the use of anonymity. When brainstorming or voting in a group process, it can be useful to work anonymously. A modifier can thus be used in combination with various ThinkLets to create different patterns of collaboration. Using anonymity creates a predictable change in the process; it removes certain barriers to share information and critique, resulting in more open discussion and higher participation (Valacich et al. 1992).

Since quality assurance interventions are used when the quality of the output shows deficiencies, they are likely to exist mainly in adjustment interventions. This means that based on a quality deficiency, a modifier is chosen to assess and resolve the quality deficiency. Quality deficiency can be identified in different ways. The group can judge each contribution on certain criteria, the facilitator can monitor input based on one or more criteria, or the contributions can be organized or compared to check completeness or consistency. The adjustment interventions for quality exist in a detection method; a way to find quality deficiencies and a way to improve the quality. In many cases focus-

ing attention on the quality deficiency will be sufficient; a group member will improve the quality of the contribution. For instance, a group member can edit his contribution to improve its quality or can re-classify a contribution when inconsistently organized. In some cases, the specific quality criteria for contributions or modifications need to be further discussed and defined. In other cases, consensus should be created about the definition of quality criteria.

Besides making adjustment interventions when quality deficiencies are detected, there are some ways to assure quality directly. These methods involve the direct emphasis or enforcement of the constraints for contribution. For instance, when a group is brainstorming for solutions for a particular problem, it helps to explicitly define the problem and to keep the problem statement visible for reference. When people define risks in terms of cause and effect, a template to formulate risks accordingly might be useful. Also, in some cases it might be useful to train group members with an exercise to create contributions that meet specific quality requirements.

5 Quality Assurance Framework

We can now define a framework based on the quality dimension identified and the methods for making adjustment interventions and emphasizing contribution constraints. This framework is intended to give an overview of quality assurance modifiers and distinguishes methods for prevention of quality deficiencies, for discovery of such deficiencies and for fixing them. In the category of deficiency prevention we find quality assurance modifiers that are mainly instruction interventions. These techniques are used to directly guard and guide the quality of contributions or modifications to the set of contributions shared by the group. In the category of deficiency discovery we find mainly adjustment interventions for detecting some quality deficiency and an action to mark or fix these deficiencies. Finally, the interventions for fixing deficiencies are again instruction interventions that prescribe an action on all identified deficiencies. It is assumed that a deficiency detection modifier is used earlier to identify deficiencies.

The framework in Table 2 offers an overview of modifiers based on the ThinkLet book (Briggs and Vreede 2001). The ThinkLet book offers an overview of best practices of GSS use, captured as design patterns (Vreede et al. 2006). The ThinkLet library has been validated in various ways; the ThinkLets have been documented best practices, and were reviewed by professional facilitators (Briggs et al. 2001). They have been recognized in GSS session transcripts (Kolfshoten et al. 2004), and they have been used in various case studies (Boehm et al. 2001; Bragge et al. 2005; Grünbacher et al. 2004b; Vreede et al. 2006). Based on the ThinkLet library, we distilled modifiers that can be used for quality assurance. Some of these modifiers were described independently; others are distilled parts of other ThinkLets. It is a first step to identify quality modifiers, and can be extended with more techniques for quality assurance. While the ThinkLet library has evolved over the years, there is no indication that this set of techniques is a complete set of facilitation techniques. Therefore, we can not assert the completeness of the set of modifiers. This set provides a starting point, and will need to evolve as a “sub pattern language” instrumental to the ThinkLets pattern language. The modifiers in Table 2 are classified as modifiers for prevention, discovery

and fixing. Modifiers for prevention and fixing are usually instruction interventions, while modifiers for discovery are usually adjustment interventions, as they require detection of a deficiency.

To give an overview, we present the rule definition of the modifiers in Table 3. In this table we indicate whether modifiers are instruction interventions or adjustment interventions. Adjustment interventions require deficiency detection, and action based on this deficiency detection. Instruction interventions are direct measures to improve the quality of the concepts and ideas shared by the group. In the next section we describe each of the modifiers and give an example of how they can be implemented.

6 Quality Assurance Modifiers

We identified modifiers for deficiency prevention, deficiency discovery and deficiency fixing as listed in Table 2. The left column shows the quality modifiers. In the table an 'x' marks the suitability of a quality assurance technique for evaluating a particular quality dimension. Each modifier rule is listed in Table 3 and is briefly described in Sect. 6.1. The rule contains various <parameters> indicated between '<>'. These need to be instantiated with actual criteria, roles or other parameters to use the modifier in a specific context. The different modifiers can often be applied to ThinkLets for different patterns of collaboration. For instance an Input template can be used to ensure the quality of a generate ThinkLet, e.g. FreeBrainstorm, but it can also be used when clarifying concepts, for instance in combination with Fast Focus. By nature, Adjustment interventions require some evaluation. However, generally we can state that these quality modifiers are used for four of the six patterns of collaboration: Generate, Reduce, Clarify and Organize (and not for Evaluation and Consensus Building ThinkLets). This is because quality assurance makes most sense while the contributions of group members are still being shaped and developed, not when they are ready for evaluation and decision making.

6.1 Deficiency Prevention

6.1.1 Input Template

To ensure the quality of a contribution, we can create an input template. This template is used to support participants in framing their contributions in a more complete, precise, consistent, parsimonious or relevant way. While this rule can be used as a guideline or instruction presented by the facilitator, it can also be enforced through technology restriction, e.g. multiple fields need to be completed in order to add a contribution.

<role> make their contributions using the <input template>, incomplete templates are not added to the list.

Instantiation example: To add a risk, please specify the risk in terms of cause and effect. You can only submit your risk when it is expressed in terms of cause and

Table 2 Framework for quality assurance in collaborative work practices

	Dimensions of quality					Relevance
	Correctness	Consistency	Completeness contribution	Completeness combined contributions	Parsimoniousness	
Deficiency prevention						
Input template	X	x	x		x	x
Constraint emphasis	X	x	x		x	x
Constraint re-emphasis	X	x	x		x	x
Contribution training	X	x	x		x	x
Definition of quality criteria	X	x	x		x	x
One up	X					x
Deficiency discovery						
Input monitoring	X	x	x	x		x
Quality evaluation	X		x		x	x
Comparison		x		x		x
Step by step checking/review	X	x	x		x	x
Cluster check		x		x		
Deficiency fixing						
Expert fixing	X	x	x	x	x	x
Chaufeur fixing	X	x	x	x	x	x
Parallel fixing	X	x	x	x	x	x

Table 3 Modifier rules

Modifier	Rules
<i>Deficiency prevention</i>	
Input template (instruction intervention)	<role> Make their contributions using the <input template>, incomplete templates are not added to the list
Constraint emphasis (instruction intervention)	Clearly state the <constraints> to the activity, and keep them visible during the activity
Constraint re-emphasis (instruction intervention)	<role> Explains the <quality criteria> to the group giving examples based on <quality deficiencies> in the set of contributions from the group
Contribution training (instruction intervention)	<role> Make their contributions using the <input template>, incomplete templates are not added to the list <role> Consider each <contribution> and judge it based on the <quality criterion/criteria>. If the <contribution> does not meet the <criterion>, ask group members to highlight or improve it <role> Look at the highlighted <quality deficiencies>, can we improve or remove them from our list? <role> improves the contribution or <role> removes the contribution
Defining quality criteria (instruction intervention)	<role> Proposes a definition for the <quality constraint> for contributions, ask the group members if they understand and agree with the definition
One up (instruction intervention)	<role> Add a <contribution> that is better than the <set under consideration> on <criterion>
<i>Deficiency discovery</i>	
Input monitoring (adjustment intervention)	<role> Monitor input of the group members, if the <contribution> does not meet <quality criterion> ask <role> to edit the contribution to improve its quality
Quality evaluation (adjustment intervention)	<role> Consider each <contribution> and judge it based on the <quality criterion>. If the <contribution> does not meet the <criterion>, ask group members to highlight or improve it
Comparison (adjustment intervention)	<role> Consider the set of <contributions> to find <quality deficiency> and highlight it
Step by step checking (Adjustment Intervention)	<role> Look at the highlighted <quality deficiencies>, can we improve or remove them from our list? <role> improves the contribution or <role> removes the contribution
Cluster check (instruction intervention)	<role> Move <contributions> to the <cluster> indicating part of the <scope> it belongs to <role> Verify if each <cluster> in the <scope> is sufficiently covered
<i>Deficiency fixing</i>	
Expert fixing (instruction intervention)	<expert> Look at the highlighted <quality deficiencies> and revise the <contribution> to resolve the quality deficiency
Chauffeur fixing (instruction intervention)	<chauffeur> Record the revision of the <contribution> to resolve the <quality deficiencies> as suggested by the group
Parallel fixing (instruction intervention)	<Participant> Look at the highlighted <quality deficiencies> and revise the <contribution> to resolve the quality deficiency

effect, for example, “Risk: The stock market crashes, reducing the value of our stock, increasing the cost of raising new capital, thereby blocking our ability to fund the project”.

6.1.2 Constraint Emphasis

All actions in a ThinkLet are taken under constraints. A constraint is a restriction to the activity, for instance a brainstorming topic can be seen as a restriction to the scope of the brainstorming activity. Emphasizing the constraints of activities will sharpen the input of participants. This is a very straightforward technique, but in practice it might be overlooked. In any collaborative activity it is useful to have a clear description or definition of an activity's constraints (brainstorm question, evaluation assignment, etc.) and to keep them visible throughout the activity.

Clearly state the <constraints> to the activity, and keep them visible during the activity.

Implementation example: The scope of the brainstorm is explained and the brainstorming question is projected on the wall during the brainstorm.

6.1.3 Constraint Re-Emphasis

This technique has been known as 'one minute madness' (Briggs and Vreede 2001). It is used to verify if participants understood the quality criterion used. After a first round of contributions, modifications, etc., the facilitator can stop the group and go through the list, verifying if the contributions have the required quality. If not, the facilitator can point out quality deficiencies, and explain again, the quality criteria.

<role> explains the <quality criteria> to the group giving examples based on <quality deficiencies> in the set of contributions from the group.

Instantiation example: We are looking for problems, please refrain from indicating solutions. For instance here it says, we do not have a monthly meeting, but this indicates a desired solution, not the problem that calls for this solution.

6.1.4 Contribution Training

This technique works the same as the contribution template, but first we let participants fill out the template for an example case, then we perform a "clarification of criteria" to bring contributions of insufficient quality to their attention and we train participants to create contributions that meet the quality criteria. Then we start working on the real topic. This technique is a combination of modifiers.

<role> make their contributions using the <input template>, incomplete templates are not added to the list.

<role> consider each <contribution> and judge it based on the <quality criterion/criteria>. If the <contribution> does not meet the <criterion>, ask group members to highlight or improve it.

<role> look at the highlighted <quality deficiencies>, can we improve or remove them from our list? <role> improves the contribution or <role> removes the contribution.

Instantiation example: Please brainstorm risks with respect to events that happened when you went to work this morning using the input template. We will go through your contributions and refine them to ensure that you formulate precise and consistent risks, than we will start working on the risks involved in this project.

6.1.5 Defining Quality Criteria

To ensure that participants understand and accept the criteria used for contributions it is useful to define them in discussion with the group. Creating definitions is simplest based on a proposal for the definition, for instance from a dictionary or other independent source. Some quality criteria are relevant for the entire process; others are only relevant for a specific step in the process.

<role> proposes a definition for the <quality constraint> for contributions, ask the group members if they understand and agree with the definition.

Instantiation example: Let's define a task as an activity with a specific objective. For instance, "organize a workshop to create the basis for our strategy report". Does everyone agree with this definition and understand how to precisely contribute tasks?

6.1.6 One Up

This technique is used to get contributions that excel on a specific quality criterion. It is used to encourage participants to come up with contributions that excel on this criterion. To increase the effect, participants can be asked to also present the argument that describes why the new contribution is better than the current set on this specific criterion.

<role> add a <contribution> that is better than the <set under consideration> on <criterion>.

Instantiation example: Please choose solutions that are cheaper than the ones we already selected for further consideration.

6.2 Deficiency Discovery

6.2.1 Input Monitoring

Input monitoring can be done by the facilitator, a chauffeur or an expert. In a brainstorming activity, unclear, imprecise, inconsistent, incomplete, or irrelevant contributions can be highlighted for improvement during convergence. Items that are added or selected to the smaller set can be monitored and refined during organizing; items that

are related can be monitored. During evaluation and consensus building monitoring is possible, but it is preferable to use high quality input for these activities.

<role> monitor input of the group members, if the <contribution> does not meet <quality criterion> ask <role> to edit the contribution to improve its quality.

Instantiation example: The facilitator monitors the input of the group members. If the solutions are not relevant to the problem under discussion, ask the contributor to edit the contribution to improve its relevance to solve the problem.

6.2.2 *Quality Evaluation*

When monitoring is difficult, for instance in large groups, or when the facilitator is busy with other interventions, a quick quality evaluation can be made after the input is generated or collected. In this case participants or people in a specific role are asked to highlight contributions that are not meeting one or more of the quality criteria. For this modifier a domain specific checklist could be used.

<role> consider each <contribution> and judge it based on the <quality criterion>. If the <contribution> does not meet the <criterion>, ask group members to highlight or improve it.

Instantiation example: Please read through the list of problems and mark each problem that is unclear to you. Please clarify contributions that you made and that are highlighted.

6.2.3 *Comparison*

Some quality criteria require comparison of contributions, for instance to judge if contributions are consistent, or to see if the entire set of contribution is complete. For this purpose we need to judge the set of contributions rather than the individual contributions.

<role> consider the set of <contributions> to find <quality deficiency> and highlight it.

Instantiation example: Please go through the list of tasks and find and highlight tasks that can be decomposed in sub-tasks to meet a similar abstraction level as the rest of the list.

6.2.4 *Step by Step Checking*

Once a list of contributions is created, it can still contain quality deficiencies. These deficiencies can be highlighted using comparison or quality evaluation and need to be resolved. There are two options to resolve quality deficiencies. When the quality deficiency is caused by a lack of relevance, the contribution can be removed. When the quality deficiency has to do with completeness, consistency, precision or parsimoniousness, the deficiency can be resolved by modifying the contribution.

<role> look at the highlighted <quality deficiencies>, can we improve or remove them from our list? <role> improves the contribution or <role> removes the contribution.

Instantiation example: Please look at the highlighted problems, these are unclear. Please either clarify your contribution or mark it with a cross. These will be removed by the facilitator.

6.2.5 Cluster Check

With a cluster check the completeness of the set of contributions is evaluated, along with the relevance and consistency of the set of contributions. The method is to create labels for clusters, and to move the contributions into the best fitting cluster. Based on this the group gets an overview of clusters for which only a few contributions are generated. When the clusters cover the scope of the problem, the group can see if the contributions are relevant to one or more aspects of the scope and the consistency of the contributions within one cluster can be verified.

<role> move <contributions> to the <cluster> indicating part of the <scope> it belongs to.

<role> verify if each <cluster> in the <scope> is sufficiently covered.

Instantiation example: Cluster the ideas to the four key focus points; let's see if we covered these focus points sufficiently.

6.3 Deficiency Fixing

6.3.1 Expert Fixing

In this modifier an expert reviews the deficiencies and fixes them based on his/her expertise. In some cases it is important to show the participants the revisions made and to verify acceptance of those revisions.

<expert> look at the highlighted <quality deficiencies> and revise the <contribution> to resolve the quality deficiency.

Instantiation example: The group has highlighted criteria that have a deviating abstraction level in comparison to the overall list. The expert is asked to resolve these issues by either decomposing or generalizing the contributions.

6.3.2 Chauffeur Fixing

In this modifier, a chauffeur fixes deficiencies in discussion with the group. Each deficiency is discussed and the revision is recorded by the chauffeur.

<chauffeur> record the revision of the <contribution> to resolve the <quality deficiencies> as suggested by the group.

Instantiation example: The group has highlighted risks that do not meet the contribution template. The group discusses each risk and sharpens it's formulation to fit the template.

6.3.3 *Parallel Fixing*

In this modifier, participants get the right to edit contributions that contain a quality deficiency. It is possible to allow participants to only edit their own contributions with a deficiency.

<Participant> look at the highlighted <quality deficiencies> and revise the <contribution> to resolve the quality deficiency.

Instantiation example: The group has highlighted ideas that are outside the scope of the solution space. The participants get a chance to rephrase their idea to better fit the scope. They can edit their ideas to create a better fit, in parallel.

7 Implementation and Implications

We have been experimenting with modifiers for ensuring quality in different group processes and contexts. Here we discuss the example of the collaborative EasyWinWin method to requirements elicitation and negotiation. It is based on Boehm's Theory-W to software requirements definition (Boehm and Ross 1989). EasyWinWin adopts group facilitation techniques helping stakeholders to diverge and converge, to evaluate their choices, and to attain consensus (Boehm et al. 2001). The approach has been successfully applied in many industrial settings and is also part of software engineering education in many universities (Grünbacher et al. 2007).

Table 4 shows the use of modifiers in selected negotiation steps together with the quality dimensions addressed by the intervention. It is important that these interventions are optional and it depends on the group's experience with that method. For instance, when working with new groups it is advisable to use interventions for deficiency prevention and on-the-fly deficiency discovery via input monitoring. With more experienced groups the focus shifts to deficiency discovery to ensure consistency and completeness of negotiation results.

We have been using the techniques described above in different combinations with a variety of ThinkLets. Each of these techniques can be implemented though facilitation. However, facilitation might not be easy, especially when large amounts of contributions are processed or created during the activity. Therefore, support tools should be developed which enable participants when highlighting quality deficiencies and when restricting contributions or modifications using a specific template.

Especially if quality means clarity, precision and thus some level of shared understanding among the group members detection of discrepancy will require time and considerable cognitive effort. The more constraints to a contribution, the more cognitive effort it requires to create or modify the contribution. Therefore, introducing quality criteria needs to be considered carefully. For instance in a creative process, too many contribution constraints can harm creativity; too much cognitive effort is

Table 4 Use of modifiers in the EasyWinWin group process

Step	What	Quality dimension(s)	Modifier
Review and expand negotiation topics	Stakeholders are asked to check the suitability of the proposed list of negotiation topics with respect to the statement of purpose	Completeness consistency relevance	Quality evaluation (deficiency discovery)
Brainstorm stakeholder interests	Stakeholders enter a few initial examples of Win conditions. The facilitator clarifies things needing attention when submitting brainstorming ideas	Relevance parsimoniousness	Contribution training (deficiency prevention)
	The facilitator shows negotiation topics while stakeholders are brainstorming	Completeness relevance	Constraint emphasis (deficiency prevention)
Converge on Win conditions	An expert checks whether the list of Win conditions covers all brainstorming ideas	Completeness	Expert fixing (deficiency fixing)
Identify issues and options	An expert checks whether the issues submitted really describe constraints, risks, or uncertainties with a Win condition	Correctness	Input monitoring (deficiency discovery)
	An expert checks whether the options proposed really describe a solution or way forward with an issue	Correctness	Step-by-step checking (deficiency discovery)
Negotiate agreements	An expert checks whether the agreements achieved by the group are consistent	Consistency	Comparison (deficiency discovery)

spent on the careful construction of the contribution, which leaves too little cognitive capacity for the processing of stimuli that trigger creativity (Santanen et al. 2004).

The same problem can occur for the facilitator. When too many quality constraints need to be verified it might be better to split the verification process into several steps to work on various quality constraints. For instance, a group can first rephrase contributions to improve precision, and then check for consistency or relevance. While it would be theoretically possible to check all three quality criteria at once, it might be challenging, and splitting the task might even be more efficient, and result in a higher quality result.

It is also very important to consider the need for certain qualities, and more important, the need to establish this quality in parallel with the entire group. For instance, it might not be necessary to create precision or consistency with the group. In such a case one group member could modify the contributions to improve their quality and send the result for verification and approval to the other group members. However, when creating a code of ethics for instance, precision completeness might be important to establish as a group as this will also increase support for the code and consistency in its execution. After all, if a code of ethics is interpreted differently, it is of little use in guiding behavior.

8 Conclusions and Future Work

This paper presents a framework for quality assurance in collaborative activities. Based on the framework we offered an overview of techniques that can be used as a variation on ThinkLets to ensure quality during different collaborative activities that create different patterns of collaboration. Furthermore, we presented implications for the instantiation and implementation of these quality assurance techniques.

We predict that all quality interventions will increase cognitive effort of the task, but when combined well, and taken into account the additive effect of the cognitive load imposed by each quality constraint, techniques can be designed that make optimal use of the cognitive capacity of the group, creating a high quality outcome given the time and effort availed for the task.

Based on this quality framework we can examine the effect of these techniques on the quality of collaborative results. The framework offers both the interventions to improve quality, and the intended effect on quality (increased precision, consistency, etc.). Also, it would be useful to compare these effects with the usefulness of the results and satisfaction with the results. It would also be interesting to measure the effect of these quality interventions on cognitive effort of the task, and in comparable conditions of group and task on the efficiency of the task. Finally, further research could indicate a similar approach is feasible to guard quality of the group interactions and relations, e.g. to detect dissatisfaction, conflict, or obstruction of the process, and to guard behavioral rules.

The techniques presented here can be used by facilitators to design and modify their interventions in groups to ensure specific qualities of the collaborative result. They can also be used by designers of group support systems to create tools that support the group and the facilitator in quality assurance. Building quality assurance into the GSS tools will not only help facilitators, it might even enable groups to effectively use such tools without support of professionals. Last the framework can be used as a research model to study the effect of these quality interventions.

Further, it would be interesting to develop intelligent agents that automatically detect specific qualities of a contribution, or quality deficiencies. In domain specific settings, self-learning agents can be built to more accurately detect quality deficiencies. The use of such agents could initially be created based on rules of thumb. For instance, when contributions need to be specific and very short contributions are made, these are unlikely to meet the quality requirements. Based on analysis of the contributions

that were altered or improved, we might be able to detect more precise indicators of the quality deficiency, thus creating a self-learning system.

References

- Badler N, Bindiganavale R, Bourne J, Palmer M, Shi J, Schuler W (1998) A parameterized action representation for virtual human agents. Paper presented at the Workshop on embodied conversational characters, Lake Tahoe, California
- Badler NI, Palmer MS, Bindiganavale R (1999) Animation control for real-time virtual humans. *Commun ACM* 42(8):65–73
- Basili V, Caldiera G, Rombach HD (1994) The goal question metric approach encyclopedia of software engineering. Wiley, New York
- Boehm B, Grünbacher P, Briggs RO (2001) Developing groupware for requirements negotiation: lessons learned. *IEEE Softw* 18(3):46–55
- Boehm BW, Ross R (1989) Theory-W software project management: principles and examples. *IEEE Trans Softw Eng* 15(7):902–916
- Bragge J, Merisalo-Rantanen H, Hallikainen P (2005) Gathering innovative end-user feedback for continuous development of information systems: a repeatable and transferable E-collaboration process. *IEEE Trans Prof Commun* 48(1):55–67
- Briggs RO (1994) The focus theory of team productivity and its application to development and testing of electronic group support systems. Tucson
- Briggs RO, Kolfschoten GL, de Vreede GJ (2005) Toward a theoretical model of consensus building. Paper presented at the Americas conference on information systems, Omaha
- Briggs RO, Qureshi S, Reinig B (2004) Satisfaction attainment theory as a model for value creation. Paper presented at the Hawaii international conference on system sciences, Los Alamitos
- Briggs RO, de Vreede GJ (2001) ThinkLets, building blocks for concerted collaboration. Delft University of Technology, Delft
- Briggs RO, de Vreede GJ, Nunamaker JF Jr. (2003) Collaboration engineering with ThinkLets to pursue sustained success with group support systems. *J Manage Inf Syst* 19(4):31–63
- Briggs RO, de Vreede GJ, Nunamaker JF Jr., David TH (2001) ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems. Paper presented at the Hawaii international conference on system sciences, Waikoloa, HI
- de Vreede GJ, Briggs RO (2005) Collaboration engineering: designing repeatable processes for high-value collaborative tasks. Paper presented at the Hawaii international conference on system science, Waikoloa, HI
- de Vreede GJ, Briggs RO, Kolfschoten GL (2006) ThinkLets: a pattern language for facilitated and practitioner-guided collaboration processes. *Int J Comput Appl Technol* 25(2/3):140–154
- de Vreede GJ, Briggs RO, Massey AP (2009) Collaboration engineering: foundations and opportunities. *J Assoc Inf Syst* 10(3):121–137
- de Vreede GJ, Koneri PG, Dean DL, Fruhling AL, Wolcott P (2006) Collaborative software code inspection: the design and evaluation of a repeatable collaborative process in the field. *Int J Coop Inf Syst* 15(2):205–228
- Fjermestad J, Hiltz SR (2001) A descriptive evaluation of group support systems case and field studies. *J Manag Inf Syst* 17(3):115–159
- Grünbacher P, Halling SB, Kitapci H, Boehm BW (2004a) Integrating collaborative processes and quality assurance techniques: experiences from requirements negotiation. *J Manage Inf Syst* 20(4):9–29
- Grünbacher P, Halling M, Biffi S, Kitapchi H, Boehm BW (2004b) Integrating collaborative processes and quality assurance techniques: experiences from requirements negotiation. *J Manage Inf Sys* 20(4):9–29
- Grünbacher P, Seyff N, Briggs RO, In HP, Kitapci H, Port D (2007) Making every student a winner: the WinWin approach in software engineering education. *J Syst Soft* 80(8):1191–1200
- Higgs M, Plewnia U, Ploch J (2005) Influence of team composition and task complexity on team performance. *Team Perform Manag* 11(7/8):227–250

- Kolfschoten GL, Appelman JH, Briggs RO, de Vreede GJ (2004) Recurring patterns of facilitation interventions in GSS sessions. Paper presented at the Hawaii international conference on system sciences, Los Alamitos
- Kolfschoten GL, Briggs RO, de Vreede GJ, Jacobs PHM, Appelman JH (2006) Conceptual foundation of the ThinkLet concept for collaboration engineering. *Int J Hum Comput Sci* 64(7):611–621
- Kolfschoten GL, van Houten SPA (2007) Predictable patterns in group settings through the use of rule based facilitation interventions. Paper presented at the group decision and negotiation conference, Mt Tremblant
- Kolfschoten GL, Rouwette E (2006) Choice criteria for facilitation techniques: a preliminary classification. Paper presented at the international conference on group decision and negotiation, Karlsruhe
- Kolfschoten GL, Santanen EL (2007) Reconceptualizing generate ThinkLets: the role of the modifier. Paper presented at the Hawaii international conference on system science, Waikoloa, HI
- Krone KJ, Jablin FM, Putnam LL (1987) Communication theory and organizational communication: multiple perspectives. In: Jablin FM, Putnam LL, Roberts KH, Porter LW (eds) *Handbook of organizational communication: an interdisciplinary perspective*. Sage Publications, Newbury Park
- Leibold M, Probst GJB, Gibbert M (2005) *Strategic management in the knowledge economy: new approaches and business applications* (2nd edn). Wiley-VCH, Berlin
- Levi D (2007) *Group dynamics for teams* (2nd edn). Sage Publications, Inc, Thousand Oaks
- Newell A, Simon HA (1972) *Human problem solving*. Prentice-Hall, Englewood Cliffs
- Osborn AF (1953) *Applied imagination*. Scribner's, New York
- Reinig B (2003) Toward an understanding of satisfaction with the process and outcomes of teamwork. *J Manag Inf Syst* 19(4):65–83
- Reinig BA, Briggs RO, Nunamaker JF Jr. (2007) On the measurement of ideation quality. *J Manag Inf Syst* 23(4):143–161
- Rich E (1983) *Artificial intelligence*. McGraw-Hill, Singapore
- Santanen EL, de Vreede GJ (2004) Creative approaches to measuring creativity: comparing the effectiveness of four divergence ThinkLets. Paper presented at the Hawaiian international conference on system sciences, Waikoloa, HI
- Santanen EL, de Vreede GJ, Briggs RO (2004) Causal relationships in creative problem solving: comparing facilitation interventions for ideation. *J Manag Inf Syst* 20(4):167–197
- Shepherd MM, Briggs RO, Reinig BA, Yen J, Nunamaker JF Jr (1996) Social comparison to improve electronic brainstorming: beyond anonymity. *J Manag Inf Syst* 12(3):155–170
- Thomas DM, Bostrom RP, Gouge M (2007) Making knowledge work in virtual teams. *Commun ACM* 50(11):85–90
- Troy LC, Hirunyawipada T, Paswan AK (2008) Cross-functional integration and new product success: an empirical investigation of the findings. *J Mark* 72(6):132–146
- Valacich JS, Jessup LM, Dennis AR, Nunamaker JF Jr (1992) A conceptual framework of anonymity in group support systems. *Group Decis Negot* 1:219–241
- Wand Y, Wang RY (1996) Anchoring data quality dimensions in ontological foundations. *Commun ACM* 39(11):86–95
- Wang RY, Storey VC, Firth CP (1995) A framework for analysis of data quality research. *IEEE Trans Knowl Data Eng* 7(4):623–640
- Winston PH (1984) *Artificial intelligence*. Addison-Wesley, Reading
- Zhang J (1997) The nature of external representations in problem solving. *Cogn Sci* 21(2):179–217