POP: An Instrument to Decide on the Adoption of Participatory Design

Helder Cognaco de Oliveira, Marcelo da Silva Hounsell(✉),
and Isabela Gasparini

Graduate Program in Applied Computing, Department of Computer Science,
Santa Catarina State University, Joinville, SC, Brazil
heldercdo@gmail.com, {marcelo.hounsell,
isabela.gasparini}@udesc.br

Abstract. Participatory Design (PD) is an approach that promotes the involvement of end-users in interactive software design. PD can be beneficial to software quality but can also raise concerns on pragmatical levels. There is no technique to help designers decide on adopting PD besides their experience on the matter. This paper proposes an objective questionnaire that gives clear indications for this decision, with confidence grading and coherence analysis. The instrument, called POP, can be used by non experts on PD, designers, developers and software analysts. The instrument has been validated by PD experts and by interactive software developers. POP was conceived for the development of educational video games but can be applied to a wider variety of systems. Results show that POP is able to clearly indicate when a project can benefit from PD, and to also give clues on the difficulties it would be facing. POP allows PD to be considered into projects that otherwise, would never evaluate how beneficial the PD approach could be to system development and to end-users.

Keywords: Participatory design · Decision-making · Questionnaire

1 Introduction

In Human-Computer Interaction (HCI), interactive software development approaches that engage end-users, directly or indirectly (i.e. in which the needs, wants, and limitations of end-users of a product, service or process are given attention at each stage of the design process), are classified as User-Centered Design (UCD). One of such approaches is Participatory Design (PD), in which end-users are involved directly, becoming active members of the development team [1]. According to Abras et al. [2], UCD can help designers by providing an understanding of factors (psychological, organizational and others) that affect the usage of a software. This can help the software development with high efficiency and efficacy, while managing end-users expectations about the final product. However, PD also has its downsides, such as the need for more resources (like time and money) during development, and the possibility that the resulting product will be too specific, or hard to adapt to a different group of users.

M. Kurosu (Ed.): HCI 2016, Part I, LNCS 9731, pp. 141–152, 2016.
DOI: 10.1007/978-3-319-39510-4_14

During the development of interactive software, the development team might have to make a decision on whether to adopt PD practices or not. This paper proposes an instrument to help developers, software analysts and designers on making such decisions, and is structured as follows. The second section presents the fundamentals of PD, while the third section focuses on related works. The content of the instrument is presented in the fourth section, with the instrument validation being described in the fifth section. The discussion regarding obtained results is shown and the conclusions are presented, respectively, in the sixth and seventh sections.

2 Participatory Design

Participatory Design (PD) is an approach that transforms end-users in active members of the development team, helping the construction of different aspects of a software. PD started in Scandinavia (Denmark, Sweden and Norway) in late 60 s, when workers started to pressure their unions for more democracy on deciding which information systems would be used in their work environments [1]. What they were effectively asking for was to choose the systems that they judged more adequate, whether because of its usability or its features.

PD became popular around the globe. In America, it started being used as a mean to develop software that effectively identified and satisfied end-users requirements [3]. In practical terms, PD consists on integrating an end-user in the development team, participating in different steps of a software life cycle, such as requirement analysis, construction/prototyping and evaluation/testing [4].

For the application of PD, there are different techniques that can be used, such as: BrainDraw [4], a round-robin graphical brainstorming technique; CARD [1], which describes processes and activity flows through the usage of cards and; PICTIVE [1], which creates low-tech prototypes of user interfaces.

3 Related Works

There is no instrument that helps designers to decide on choosing to use PD or not. However, following related works defined certain aspects of the PD approach that can be evaluated to a certain degree with the instrument proposed in this paper.

Bratteteig and Wagner [5] proposed a classification of the types of decision made during PD, as a way to analyze how the division of power occurs between developers and users. The decisions are summarized as follows:

- Values and concepts: how the participation occurs and how much will it affect the final design;
- Vision implementation: how the design will be applied; what are the identified solutions to the problems the software is trying to solve;
- Negotiations with the outside world: decisions not directly related to the software like, for instance, the choosing of the users who will participate in the design;

- Non-decisions: decisions made without deliberation or communication; inexplicit decisions. The authors describe the action of "accepting a 'gift'" as a good example of such decision.

These decisions are made considering certain variables, like participants' profiles and applicable techniques. It is necessary to choose a technique that is adequate to the user [6].

Inspired by the work of Greenbaum and Halskoy [7], a classification was proposed by Bergvall-Kåreborn e Ståhlbrost [8], regarding the motivations of software development with PD. According to this classification, there are three distinct perspectives:

- Ethical (democracy): human beings have the right to influence their destinies. Therefore, they also have the right to influence technological decisions that affect their professional and private lives, like choosing an adequate software;
- Curiosity (theoretical): projects motivated by the curiosity regarding the nature of participation, best practices in participation, good and bad outcomes (for the participants and for the software);
- Economic (pragmatic): projects with this motivation focus on achieving best results as possible, regarding software quality and acceptance.

Despite the rationalization made by these works on the motives and values of shared decisions, they do not present an objective way to evaluate whether PD should be opted, or not. The proposal of an objective instrument that is easy to use, can help the analysis of the outcomes that PD can bring to a software development project.

The instrument, named POP, presented in the next section, was based on empirical knowledge from researchers, and do not conflict in anyway with presented classifications.

4 Objective Participatory Questions (POP)

The Objective Participatory Questions (in Portuguese namely *Perguntas Objetivas Participativas*, hereby called POP), is a questionnaire, created to acknowledge and evaluate specific aspects of the participatory process, prior to opting for this kind of approach.

The creation of POP was based on empirical knowledge, obtained by a team that developed persuasive educational video games about drug addiction. Despite being created in a very specific context, POP content is generic enough so that any interactive software development team can benefit from it. The questions in POP were originated from the advantages and disadvantages observed before, during and after the participatory development of educational video games.

The questionnaire is composed of 11 questions, available in Appendix A. Each question deals with a specific topic in the context of PD, as experienced by the researchers. These topics are:

- Technical Benefits: some benefits and harms can easily be identified by designers and domain experts. Therefore, it is important to take their opinion on the necessity of using PD, or not;

- Personal Benefits: a participation that brings personal benefits can be of great value to those involved;
- Logistics: a participation with complex logistics like, for instance, a group of users with limited time to participate, should not be advised;
- Profile: to depend on a very specific kind of user can make it difficult for a participatory process to happen. Generic users are easier to be obtained, so it is more likely that enough participants will be found to help the design process;
- Volatility: a group of volatile users can harm long-term participatory processes, with little homogeneity on the participation and decision making;
- Group Size: large groups of end-users can be hard to coordinate and organize for participatory sessions, which could delay development. Besides, finding a participatory technique that handles a large group of participants can prove difficult. According to Muller [4], PD techniques support up to 14 participants in average, to a maximum of 40 participants;
- Empathy: empathy among developers, domain experts and end-users can facilitate complex multidisciplinary projects, for it helps different teams to work together appropriately. Antipathy, on the other hand, can make software development difficult in important steps of its life cycle, such as requirements analysis;
- Conceptual Contribution: the participation of end-users during requirements analysis is a first step to develop a software with PD that is adequate to end-users expectations;
- Technical Contribution: participation in technical steps of development can help to assure the correct implementation of previously established requirements, with end-users helping to translate them into data, models, code, etc.;
- Conceptual Framework: as conceptual steps in the software life cycle result in the creation of requirements, which are the fundamental to the software development, it is important to use techniques for participation that help to obtain end-users' requirements with clarity, in order to avoid as much rework as possible;
- Technical Framework: inappropriate participatory techniques in technical steps of development may result in a low quality user interface, which is hard to comprehend and to work with.

For each question, there are four possible answers: positive, neutral, negative, and abstentious. The positivity/negativity characterization of each answer is represented by points that are assigned to them. Moreover, each question has a description text to help designers understand its context.

During the development of educational video games using PD, the authors noticed that, regarding decision-making, certain aspects of participation were more important than others. To reflect that, some questions that have bigger influence in the recommendation of PD, have answers with absolute values that are higher than the answers in less influential questions. The definition of which questions have higher valued answers was based on researchers' empirical experiences.

4.1 Decision-Making

The results from the questionnaire can be analyzed through three indicators, which are based on the following variables:

- s: is the sum of points assigned to all the questions answered by the designer;
- r: is the amount of questions that were effectively answered, meaning questions which answers were different than "d" (abstentious);
- m: sum of the absolute value of all questions answered by the designer. It represents the best possible score the designer could have obtained with the set of questions he answered.

With all these three variables, it is possible to calculate the indicators:

- Final Indication (*fi*): it is obtained by analyzing the value of s. If the value of s is positive, then end-user participation is recommended. If it is negative, than participation is not recommended. If it is zero, then it is not possible to conclude on whether participation is recommended or not.
- Confidence (*cf*): represents how confident the *fi* indicator is, based on the amount of questions effectively answered. The higher the confidence, the more grounded the recommendation is. This indicator is calculated by Eq. 1.

$$cf = (r/11) * 100 \tag{1}$$

- Coherence (*cr*): represents how coherent the *fi* indicator is, based on the trend observed on answered questions. In other words, it shows how many questions were answered positively, if participation is recommended, or answered negatively, if participation is not recommended. This indicator can only be calculated (Eq. 2) if s is different than zero.

$$cr = [(|s| + m)/2m] * 100 \tag{2}$$

Analyzing all these three indicators, designers can obtain a recommendation on whether using the participation of end-users or not, how confident that recommendation is, and how coherent among each other the answers were.

For instance, the application of this instrument can result in a final indication that participation is recommended (because s is positive), with a low 13 % of confidence (because many questions were left unanswered), but with a high 84 % of coherence (because most answered were positive).

5 Validation

This instrument was validated in two different ways:

- Experts' Validations, to validate if POP is relevant as a decision making tool for PD supported development, experts were consulted, with unstructured interviews;

- Practitioners' Validation, to validate if the usage of POP reflects decisions taken by designers in real development scenarios, designers used the instrument in both current and past projects (retroactively).

5.1 Experts' Validation

During this initial validation, the questionnaire was evaluated by three PhD professors, with experience in PD whether teaching about it in class or applying it in research projects and software development. To help with this validation, they were provided with a list of justifications for each question, explaining the reasons each of them were created, how relevant they were to PD, etc. These justifications are not present in Appendix A.

After they analyzed the instrument, each one participated individually in an unstructured interview. Based on feedback obtained through these interviews, some adjustments were made to the instrument: removal of some questions; changes in the points assigned to the answers; improvements in the description of certain questions.

Questions removed were focused on applying POP into a specific development methodology. With the removal of these questions, POP was made generic enough so that it can be applied into many other development methodologies.

The point system was changed from a scale of −10 to 10 points, to a scale of −3 to 3 points. This way, it is easier to calculate the questionnaire results, for the variables will have smaller values. Moreover, it was decided that certain questions would have higher valued answers than others, in order to emphasize the importance of certain aspects to the practice of PD. Questions that are more relevant have answers ranging from −3 to 3, while less relevant questions have answers ranging from −1 to 1.

Finally, the description of some questions were improved, in order to clear them up and make them easier to understand. Some professors judged that certain descriptions were not really helping the understanding of the question's context, and some contained words not commonly used by practitioners. Where possible, descriptions were changed to include advantages and disadvantages in each context, and more examples of application.

5.2 Practitioners' Validation

During the second validation, designers, HCI specialists and developers answered the questionnaire in order to see if its results were similar to the decisions they had made. Some designers answered the questionnaire retroactively, that is, as if they were at the start of a project that is already finished. This way, it could be observed if the results were consistent with the choice of the designers.

Professors who participated in the first validation, helped the researchers to select designers for the second validation. In total, eight cases were analyzed (results are shown at Table 1) with designers enrolled to each project using POP, whether it was retroactively, or not. Selected cases were:

Table 1. Results obtained during the second validation of POP

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
PD?	Yes	Yes	Yes	Yes	Yes	No	No	No
fi	4	18	14	16	12	−5	−10	−4
cf	75 %	100 %	66,7 %	83,3 %	100 %	75 %	100 %	83,3 %
cr	62,5 %	90,9 %	100 %	94,4 %	83,3 %	66,7 %	66,7 %	66,7 %
Retroactive?	Yes	Yes	Yes	Yes	No	No	Yes	No

- Case 1: Development of three persuasive educational video games, about the "12 Steps" program to fight drug addiction. Participation helped the creation of end-users' requirements, and graphical elements of the games;
- Case 2: Development of a software for healthcare industry, which managed multidisciplinary medical treatments in a health institute for the elderly. Professionals participated to identify multidisciplinary tasks and to define software interfaces;
- Case 3: Development of a retail software, with end-users participating during requirements analysis and prototyping;
- Case 4: Development of an electronic document management software for law practice. Participants helped on requirements analysis and made suggestions during ergonomic evaluation;
- Case 5: Development of an educational video game to help drug addicts during treatment, to ease their way back into society. Participants helped to create end-users' requirements and the storyline of the game;
- Case 6: Development of an educational video game to help patients who suffered stroke, to reacquire standing balance during rehabilitation;
- Case 7: Development of an educational video game to alphabetize children with Down's syndrome;
- Case 8: Development of an education video game to improve math literacy in children with Down's syndrome.

Each column of Table 1 presents data related to a certain case. The "PD?" row shows whether each case decided to use PD or not. The following three rows shows the results of each indicator generated by POP, respectively, Final Indication, Confidence and Coherence. The last row shows whether POP was applied retroactively or not.

All recommended suggestions (*fi*) obtained by using POP were consistent with the decisions taken by the designers. Low coherence values (below 70 %) were obtained in cases 1, 6, 7 and 8. In particular, case 1 had the lowest value, for the aspects of logistics and volatility had negative answers. The results indicate that, although recommended, the participation could prove to be difficult to execute, since at least some questions were answered negatively. It is up for the person making the decision to analyze whether it is worthy or not to use PD considering the logistical problems he or she is going to face, and if the volatility of end-users participating is harmful or not to the software development.

6 Discussion

POP does not rely on a specific development methodology and therefore, designers can use it to identify if a participatory approach is recommended, regardless of the development methodology.

Experts' validation resulted in changes that made the instrument more adequate to professionals of the field, by using the right vocabulary. Moreover, the structure of questions, explanations and answers were also improved, making the instrument easier to be used.

Practitioners' validation indicates the instrument is consistent with decisions made by designers in real scenarios, meaning POP is a valid instrument for decision making, with relevant contexts being reflected throughout its questions, and its different valued answers.

While answering the questionnaire, the designer responsible for case 2 criticized a question regarding how the resulting software would be used. If the software was to be used spontaneously, then participation would be recommended to motivate users. If the usage was obligatory, like in a class activity or corporate training, then participation would not be recommended. After deliberation, this question was removed from the instrument, because it was judged that both scenarios could benefit from end-user participation. Even if a software is going to be used regardless of users' motivation, it can benefit from better analysis and be more adequate to users.

During both validations, no participant questioned or indicated the existence of a similar instrument for decision making in PD. This suggests that POP is a novel instrument.

Because of the way POP is composed, there is no need for a deep knowledge in UCD, or even PD. Therefore, a designer can analyze if PD is recommended or not and, if necessary, can focus on studying participatory practices.

It seems unusual for designers to consider PD in their projects, unless they have a certain affinity with this approach. With POP, more designers can consider PD, for there is no need for previous knowledge to use it.

POP was created to be a tool in a video game development methodology, but it is structured in a generic way, so that it can be applied to any interactive software development methodology.

The instrument can help designers whose motivation are classified as pragmatic [8], because it helps them evaluate certain viability issues (logistics, volatility, empathy, benefits and contributions) that are important to projects within this classification.

This viability information can also benefit decision making in projects with curiosity (theoretical) motivation [8], because if POP results do not recommend participation (with high confidence and coherence values), then the end-user participation will be difficult to execute, with the risk of incorrect data being generated. These projects can also benefit from POP when the question regarding personal benefits is answered positively. This was observed in cases 1 and 5, during the second validation. In both cases, the direct participation of drug addicts to develop educational video games resulted in the transformation of the participatory process into a therapeutic process, with the appropriate help of field professionals.

However, POP does not help projects with exclusively ethical motivations, because arguments that are fundamental to these projects (such as democracy at workplace) addresses psychological and social concepts that are beyond the scope of this research.

Questions related to participation in conceptual and technical steps of development can help in decisions regarding division of power [5]. These decisions are influenced by the way participation is executed. Besides, by classifying the level of contribution in participation, it is possible to analyze the amount of freedom that will be given to participants.

The final form of POP, presented in Appendix A, is slightly different than the one used during the validation processes. Some questions were removed, in order to address only relevant contexts of PD, and the overall writing was improved, to make POP easier to understand and to use.

Although in certain cases POP may not recommend end-user participation, there are still other UCD approaches that could be considered by designer, developers and software analysts, such as Contextual Design and Collaborative Design.

7 Conclusions

Although there are studies that classified different kinds of participatory practices, motivations, division of power, and other contexts related to Participatory Design (PD), no objective decision-making tools for designers were found.

The instrument proposed by this paper, POP (*Perguntas Objetivas Participativas*, or Objective Participatory Questions in English), is a questionnaire that evaluates the context of a software development project, and generates quantifiable information, in order to help a designer on making a decision about using PD during development. POP can be used by designers, regardless of their previous knowledge on UCD, or even PD itself.

The instrument has been validated in two different ways, by experts and by practitioners, and none of them questioned its novelty. Validation fine-tuned the instrument that is presented in its final form in Appendix A.

Results obtained by the use of POP, indicate whether PD is recommended or not, but the final decision is taken by the designers, which can be opposite to POP's recommendation. In such cases, the results of POP indicate if the project is losing a good opportunity to involve end-users or, oppositely, how difficult and disadvantageous it would be to execute their participation.

It is believed that this is a novel tool for decision-making in PD, and that it can benefit developers and designer, by providing them with information to support their final decision. This could help spread the practice of PD by showing to designers who previously would not consider this approach, how it can be beneficial to their projects.

A. Appendix

Before answering the questions below, reflect about: (a) What is the target audience (end-users and their characteristics)? (b) Where can individuals from the target audience be found, to eventually participate?

<table>
<tr><td colspan="4">POP 1: (Technical Benefits) What is the technical impact of end-users participating in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Participation can bring technical benefits to development such as requirements that are more adequate to end-users, but can also bring harm such as delays and difficulties in conciliating end-users' desires with domain experts' desires.</td></tr>
<tr><td>(a) Beneficial (+2)</td><td>(b) None (0)</td><td>(c) Harmful (−2)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 2: (Personal Benefits) What is the personal impact for end-users to participate in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Participation can bring inexplicit personal benefits to end-users, such as improvements in interpersonal relationships, occupational therapy, acquisition of technical knowledge, and others, but can also make users uncomfortable when discussing certain aspects of the software.</td></tr>
<tr><td>(a) Beneficial (+2)</td><td>(b) None (0)</td><td>(c) Harmful (−2)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 3: (Logistics) How difficult are the logistics of allowing end-users to participate in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Participatory tasks may require end-users transportation, resource allocation, scheduling and etc.</td></tr>
<tr><td>(a) Easy (+3)</td><td>(b) Neutral (0)</td><td>(c) Hard (−3)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 4: (Profile) How difficult is it to find and involve end-users that fit into a desirable profile for participation in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Generic profiles can be filled by most end-users, while specific profiles can only be filled by a small group of end-users.</td></tr>
<tr><td>(a) Easy (+2)</td><td>(b) Neutral (0)</td><td>(c) Hard (−2)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 5: (Volatility) Considering the duration of the development project, how volatile is the group of end-users that would participate in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Some groups of end-users are volatile, for they constantly change in formation or organization, while non-volatile groups can remain unchanged during the whole software development.</td></tr>
<tr><td>(a) Not volatile (+3)</td><td>(b) Variable (0)</td><td>(c) Volatile (−3)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 6: (Group Size) What is the size of the group of end-users that would participate in the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Participatory Design techniques can, in average, support groups of 2 to 14 participants. At most, these techniques can support up to 40 participants.</td></tr>
<tr><td>(a) Up to 14 people (+2)</td><td>(b) Between 15 and 40 people (0)</td><td>(c) More than 40 people (−2)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 7: (Empathy) What is the level of empathy between technical staff and end-users?</td></tr>
<tr><td colspan="4">Explanation/Example: Technical staff can be composed of programmers, designers, analysts, etc. Empathy must be seen as personal and professional proximity between both groups. With a high level of empathy, both groups share a vision about definitions and objectives of the software, know the specific vocabulary of each domain, have similar previous experience working together, etc. With a low level of empathy, the groups don’t know each other from previous works, have difficulties communicating adequately, etc.</td></tr>
<tr><td>(a) High (+2)</td><td>(b) Neutral (0)</td><td>(c) Low (−2)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 8: (Conceptual Contribution) What is the potential for contribution from end-users while participating in conceptual steps of the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Conceptual steps, like requirement elicitation and analysis, are prior to codification and construction of graphical elements of the software. End-users can contribute during analysis by presenting their perspectives about the software. However, in some cases the requirements cannot be interfered by end-users.</td></tr>
<tr><td>(a) High (+1)</td><td>(b) Neutral (0)</td><td>(c) Low (−1)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 9: (Technical Contribution) What is the potential for contribution from end-users while participating in technical steps of the software development?</td></tr>
<tr><td colspan="4">Explanation/Examples: Technical steps involve the creation of interfaces, definition of software architecture, codification, evaluation, assessment, etc. Participation in such steps may allow end-users to contribute with graphical elements to the interface, codification with high-level programming tools, evaluations through interviews, interaction tests, focus groups and controlled experiments.</td></tr>
<tr><td>(a) High (+1)</td><td>(b) Neutral (0)</td><td>(c) Low (−1)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 10: (Conceptual Framework) What is the level of knowledge (tools, processes, methods, techniques) to involve end-users in conceptual steps of the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Requirements’ analysis and elicitation can be executed with participation through brainstorming techniques and interviews.</td></tr>
<tr><td>(a) High (+1)</td><td>(b) None (0)</td><td>(c) Bad (−1)</td><td>(d) Unknown (0)</td></tr>
</table>

<table>
<tr><td colspan="4">POP 11: (Technical Framework) What is the level of knowledge (tools, processes, methods, techniques) to involve end-users in technical steps of the software development?</td></tr>
<tr><td colspan="4">Explanation/Example: Multimedia elements of the software can be created with participation of end-users thought audio recordings, photographs, drawings, etc. Authoring tools can enable participation in codification. Interaction tests and questionnaires can be used during tests and evaluations.</td></tr>
<tr><td>(a) High (+1)</td><td>(b) None (0)</td><td>(c) Low (−1)</td><td>(d) Unknown (0)</td></tr>
</table>

References

1. Preece, J., Rogers, Y., Sharp, H.: Interaction Design: Beyond Human-Computer Interaction. Wiley, New York (2011)
2. Abras, C., Maloney-Krichmar, D., Preece, J.: User-centered design. In: Bainbridge, W.S. (ed.) Berkshire Encyclopedia of Human-Computer Interaction. Berkshire, Great Barrington (2004)
3. Chin, G.: A case study in the participatory design of a collaborative science-based learning environment. Virginia Polytechnic Institute and State University, Blacksburg (2004)
4. Muller, M.J.: Participatory practices in the software lifecycle. In: Helander, M.G., Landauer, T.K., Prabhu, P.V. (eds.) Handbook of Human-Computer Interaction, 2nd edn. Elsevier, Amsterdam (1997)
5. Bratteteig, T., Wagner, I.: Disentangling power and decision-making in participatory design. In: Proceedings of the 12th Participatory Design Conference: Research Papers, vol. 1, pp. 41–50 (2012)
6. Slegers, K., Duysburgh, P., Hendriks, N.: Participatory design with people living with cognitive or sensory impairments. In: CHI 2014 Extended Abstracts on Human Factors in Computing Systems, pp. 49–52 (2014)
7. Greenbaum, J., Halskov, K.: PD a personal statement. Commun. ACM **36**(6), 47 (1993)
8. Bergvall-Kåreborn, B., Ståhlbrost, A.: Participatory design: one step back or two steps forward?. In: Proceedings of the Tenth Anniversary Conference on Participatory Design, pp. 102–111 (2008)