When Designers Are Not in Control – Experiences from Using Action Research to Improve Researcher-Developer Collaboration in Design Science Research

Anders Hjalmarsson[1,2], Daniel Rudmark[3], and Mikael Lind[1,2]

[1] University of Borås, School of Business & Informatics, Sweden
[2] Viktoria institute, Sweden
[3] University of Borås, InnovationLab, Sweden
{Anders.Hjalmarsson,Daniel.Rudmark}@hb.se,
Mikael.Lind@viktoria.se

Abstract. Design science research (DSR) has received much attention in the past few years from the field of information systems. This paper argues that control in researcher-developer collaboration during artefact development has not yet received enough attention in design science research even though control is necessary for successful artefact instantiation. Experiences are presented from improving researcher-developer collaboration during DSR by using action research (AR) as means. These experiences are driven from the need to achieve meta-design control throughout the development of artefacts by non-researching system developers when DSR is performed in an authentic setting. The paper shows that the use of AR to both diagnose uncertainty and actively improve building activities may lead to sustainable improvements in researcher-developer collaboration, and hence achieve meta-design control in DSR performed in authentic development environments, as well as enhance progress in DSR methodology development.

Keywords: Design science research, action research, researcher-developer collaboration, meta-design control, authentic setting.

1 Introduction

In 50 years of research within the field of information systems (IS) a large amount of work has been dedicated to actual system construction. Recently, more and more attention has been paid to a theory of the artefact as such [1] and it has been speculated that researchers should actively engage in the actual design of the systems they study through Design Science Research (DSR) [2]. Within IS DSR a number of argumentations have been posed on the potential relationship between action research (AR) and DSR and whether these two methods of inquiry may benefit from joint uses [3, 4, 5]. This debate has so far concerned the environment in which the artefact is operating within, and to what extent the artefact should intervene (through evaluation) and be informed by (through design propositions) a particular environment and what potential role this plays for the final design theory. However, engaging in such research regarding artefacts, be it by focusing on technical principles [6] or on the construction of more socio-technical systems [7], requires complex construction activities.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 1–15, 2010.

Researchers pursuing DSR are constructing artefacts in the domain of their study. This does not necessarily mean that the researchers are the sole responsible for the implementation of the artefact: in fact, they may cooperate with other specialists in order to construct a design. Hence, particular actions taken as a result of the interplay between researchers and developers could have an impact on the final design theory [7]. Still, DSR methodology literature has yet to articulate how to make the best of such collaborative construction settings. Hence, this paper aims at contributing to the research activity build (construction, artefact development) in DSR and provide experiences how to control the relationship between researcher-developer when artefacts are constructed. More specifically, we do this by exploring Action Research (AR) as a means to develop meta-design control during artefact development in an authentic DSR setting. AR is here used as a structure in a DSR project concerned with platforms for architectural innovation using collaborative artefact construction. Experiences and the constituents of this use of AR are framed in accordance with canonical action research (CAR) [8]. This research is motivated, as discussed above, by the lack of knowledge concerning the interplay between researchers and developers engaged in construction in DSR projects; as such, the research question in this paper is *how to obtain meta-design control in researcher-developer collaboration during artefact construction?*

The paper is organized as follows: first, we present a review of DSR literature focusing on artefact construction and the need of meta design-control in design research. This is followed by a short overview of AR and some of its currently established relationships with DSR. Next, we detail our experiences from collaborative construction and how AR may be used to obtain control in researcher-developer collaborations. We conclude with a discussion of the experiences we made and how this use of AR relates to previous uses of 1) AR in DSR, 2) meta-design control and 3) AR in one's own organization.

2 Artefact Development, Authentic Settings and Meta-design Control in DSR

In DSR, knowledge about and understanding of a problem domain and its artefact-based solutions are achieved through a scientifically grounded implementation and evaluation of artefacts [9]. At the core of DSR thus lies the creation of artefacts solving problems that so far have not been solved. In DSR, the formulation of a problem typically proceeds the construction of artefacts [10], but it has been argued that the opposite also holds true [11]. Hence, it might be argued that the research problem and the artefact typically co-evolve over time [12] requiring sufficient method support to effectively deal with these changes.

A closer investigation of widely used frameworks in this line of research reveals similar although not identical structures. The most cited framework by Hevner et al. [9] argues that IS researchers in DSR are concerned with two major activities: develop / build (theories and artefacts) and evaluate (through a variety of methods ranging from simulations to case studies to field studies). These activities are influenced both by the environment (expressed as business needs) and by the existing body of knowledge (theories). Nunamaker et al. presented a framework [11] which was later extended by Venable [13]. This latter extended version places solution technology invention as the central activity in a DSR project, surrounded by theory building and evaluation (either naturalistic evaluation or artificial evaluation).

Considering the importance of the artefact construction process to DSR argued for in the frameworks we reviewed, the literature has so far paid insufficient attention both to DSR in an authentic setting and to the challenges that could occur during researcher-developer interplay in a development environment such as that. On a more general, problem-solving level, guidance to this process has been offered [10] and recently extended [14], but when it comes to the day-to-day handling of artefact construction an insufficient volume of investigations has been performed. In DSR these activities may not be considered to be merely instrumental on behalf of the researcher. Rather, as shown for example in Markus et al. [7], reflections based on the interplay between researchers and non-researching developers had a direct influence on principle no. 6 in DSR [7 p. 202].

Engaging in exploratory research ("*knowing through building*" [15]) requires a different set of skills and practices than other types of research approaches where the researcher is potentially more in control of the inquiring instruments (i.e. experiments, interviews, surveys). Since the actual construction activities for a certain artefact may lie outside the scope of the core competency of the researcher(s), these skills must be obtained elsewhere, for example by engaging professional developers and end users in an authentic development setting. Since the research is then concerned with exploring different options to achieve a desired future state, uncertainty and unanticipated changes are given and must be dealt with on a regular basis.

This emphasizes a need for alternative forms of control during DSR. Meta-design [16] is an emerging framework aimed at defining and creating social and technical infrastructures in which new forms of collaborative design can take place. It expands beyond the traditional notion of system development by allowing users as well as developers to become co-designers [17]. Meta-design is grounded in the idea that future uses and problems cannot be completely anticipated at the time of the design, when an artefact is developed: Similarly, we argue that problems during DSR cannot be completely anticipated in advance. In order to obtain control in the DSR process challenges must be identified and managed by the researchers together with the developers by using a jointly established meta-design framework. The aim of this paper is to explore AR constituting such a meta-design framework when DSR is performed.

3 AR as a Means to Improve Practice

AR is a methodology which intends to contribute both to practical concerns in problematic situations and to the scientific body of knowledge. Its origins can be traced back to Lewin [18] and one of his descendants, Blum, states that "*AR includes diagnosing social problems with a view of helping to improve the situation*" [19]. This is also evident in the work carried out by Susman and Evered, CAR, [8] which prescribes more structure to AR than Lewin and Blum's original work did, and, by doing so, adds more scientific rigour to the research process [20, 21]. In CAR, five canonical phases are described in a cyclic process: 1) diagnosing, 2) action planning, 3) action taking, 4) evaluating, and 5) specifying learning. Before iterating a research environment labelled, the client-system infrastructure has to be established. Susman has later described the phases in greater detail, based on a pragmatic perspective (anchored in John Dewey's conception of inquiry), where the working hypothesis is introduced as an important element of CAR [22].

Furthermore, Davison et al. introduce five principles for CAR which enable the researcher to evaluate the AR project in order to assess research quality [23]. These are: 1) the Principle of the Researcher–Client Agreement, 2) the Principle of the Cyclical Process Model, 3) the Principle of Theory, 4) the Principle of Change through Action, and 5) the Principle of Learning through Reflection. By applying these elements to the AR process, rigour in AR is strengthened, as is suitability as a framework in and for research situations in authentic settings; e.g. when problems occur during collaborative construction of artefacts in DSR conducted in an authentic development environment.

Being a methodology for improving practice, AR could be used in situations in which the researcher intervenes in an external organisation. However, as pointed out by Coghlan and Brannick, AR could also be performed from the inside of an organisation, and then aim at achieving both personal goals from the project as well as contributing to the organisation the researcher is a part of [24]. Amongst the challenges that occur are 1) the development of ways of knowing what is familiar for the organisation, and 2) the development of practical knowledge that enables the researcher to understand what is taking place, and how to appropriately intervene in her own organisation. Being up to the challenge of holding both an existing organisational role and an adopted AR role with the purpose to improve the ongoing practice is particularly important for project success.

4 Established Uses of AR in DSR

In recent years a number of researches have discussed the relationship between AR and DSR. In this paper we do not seek to confirm the validity of these argumentations, but rather to demonstrate how AR can be used as a framework to *obtain meta-design control in researcher-developer collaboration* during artefact construction when DSR is carried out in authentic settings.

Cole et al. suggest that AR could be used in DSR in two ways: 1) by adding an AR cycle in the last part of the evaluation to increase reflection and learning; 2) "*a DR project may be framed as an AR project if an organizational problem needs to be solved, and the action involves building a system" [4].* Iivari and Venable [5] find similar joint usage. Either AR can be used as a naturalistic evaluation method or when developing artefacts for and in socio-technical settings [13].

Lee finds striking similarities between AR and DSR and demonstrates how the research activities/research output framework proposed by March and Smith [26] may be framed using AR. Lee argues that a two-cycle AR version of March and Smith' framework could then serve for the development of the client's business practice and as a means for creating theories for the scientific community [25]. Interestingly, the build phase of an instantiation is however omitted in this framework [25].

In this paper, AR is used as means to improve DSR during the build process, i.e. construction/development activities. Our suggestion is that AR is a suitable framework for solving problems in relation to control that arises during collaborative artefact development. We hence add knowledge to the existing couplings between these two research approaches by recognizing that artefact development in an authentic design research setting may be of collaborative nature in which researchers, through the actions of others (i.e. developers), modify underlying technology. In such settings problems may arise between researchers and developers which need intervention in order for the DSR project to succeed (see figure 1), while simultaneously a way to increase the body of knowledge about DSR is achieved.

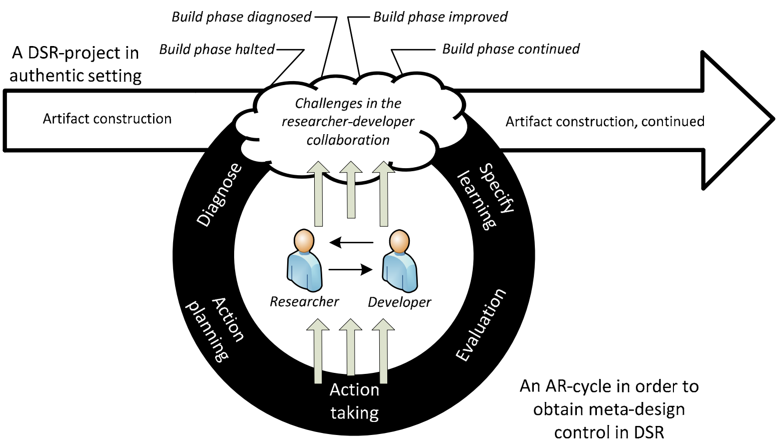

Fig. 1. AR as framework to obtain meta-design control in researcher-developer collaboration artefact construction is performed throughout DSR

We consequently propose another use of AR in DSR, this in addition to Cole et al. [4] suggestion of adding an AR cycle as a last part in the evaluation or framing of the DSR project as an AR project. AR is instead performed within a DSR-project as a means to obtain meta-design control when DSR is performed. The following case study demonstrate the challenges that may arise during artefact development, and how meta design control jointly could be obtained by using AR as a framework to improve the researcher-developer interplay.

5 Case Study: Improving Distributed Artefact Construction in DSR with AR

5.1 A Need of Control in DSR: The Halted Build Phase

The DSR in the subsequent case study was aimed to develop an architecture for a digital assistant through which users could induce their own custom-programmed services. It initially started in 2006 when an electronic assistant for students was elaborated in numerous workshops, verified through surveys and built as a prototype [27]. The evaluation of the implemented artefact indicated that it was considered useful and filled an important space not currently occupied. A first version of the assistant was built and evaluated in 2006 and 2007 [28].

In November 2008, a DSR project was organized for building an improved architecture with added functionality. It was also decided that the development methodology to be used for this DSR project was to adhere to agile principles – and the Scrum methodology was chosen [29]. This was due to the fact that the architecture should be explored together with prospective users which supposedly made a more plan-oriented method a bad fit and that the development organization was familiar with this method due to experiences from using this methodology in other projects. The project was staffed with two Scrum masters, both acting as DS researchers sharing far-reaching experiences in systems development, collaborative modelling, facilitation, and DSR. In addition, the core development team consisted of two senior developers/architects and one member responsible for quality assurance (QA). Later in the project one additional part-time developer joined the development team.

Sprint 1 unfolded roughly as planned. During this phase focus remained on testing the individual proposed technical frameworks that the architecture was to be built upon; e.g. the Google API and various open source software (such as MySql, JBoss and Spring).

In Sprint 2, the first working functionality was to be delivered. In the spirit of Scrum, it was important to deliver working features for evaluation early in the DSR project. The researchers found it especially important to test the implemented design with prospective users for naturalistic evaluation. At the start of Sprint 2, the activities were collectively defined and time-estimated during a four-hour session (as advised by the Scrum methodology [29]). Since this project was in the area of architectural innovation, the necessary time-estimated activities concerning system architecture identified at the time grew relatively large, ranging from approx. 20 to 160 hours per activity. These numbers were a result of the great uncertainty invoked by the process of building the innovative architecture. Although the task was considered complex, there was consensus in the team that the expected delivery was feasible.

Nonetheless, just one week before the testing was to begin the development team realized and communicated on the Scrum meeting that deliverables would not be completed in time. The construction of a component that translated the service need for information exchange had turned out to be more complex than initially estimated. The DSR researchers had stressed the importance of a truly generic framework, making it not only possible, but also inspiring for diverse stakeholders, to construct their own innovative services [30].

Although this vision served as an inspiration for the development team during Sprint 2, the consequences of realizing the ideal had become difficult to foresee during the development. For example, implementing a generic workflow engine revealed much more complexity than first anticipated; how to combine simplicity, flexibility, and third-party developer inspiration into an API required a lot of designing and re-designing. Nevertheless, in order to meet the delivery date, a first version of the artefact was implemented using less generic technology than first planned for. As a result, delivery happened on time, but the technology used in this implementation was reduced to an absolute minimum.

During the planning of Sprint 3 the objective was set to implement the ability to transfer information from the university scheduling system to an external calendar such as Google Calendar. More doubts were now raised during the planning as to whether the goal was actually achievable, but after discussing the matter and conducting collective time estimation the goal was maintained. About two weeks into Sprint 3 the development team announced that the deliverables would not be ready at the end of the Sprint. Each day that passed revealed that more things were necessary in order to succeed but were missing in the core architecture. This in turn made it difficult for developers to keep up with delivering the planned functionalities. At this point it was decided to black-box the scheduling system and instead build a mock-up service, using a correct programming interface but delivering nothing but predefined data (no real class schedules). This way a functional system could be delivered in order to QA the usability of the system, although the actual integration with the scheduling system was omitted. About a week later, this new goal turned out to be unreachable as well since the development work with the core architecture remained unfinished. The Scrum masters and developers tried to define yet another reduced set of features that could be delivered at the end of Sprint 3, but the ones that were considered achievable added too little value for the potential users compared to the QA resources needed in order to finalize them. Hence, all deliverables were postponed to Sprint 4.

5.2 Initiating an AR Cycle to Improve the Build Phase

At this stage it became apparent that the method-in-use needed to be reconfigured. In order to give the project new momentum and re-establish control, the DS researchers, acting as Scrum masters, and together with the rest of developers, agreed that the DS researchers should initiate an inquiry of the causes behind the problems. The results of this investigation would have provided the basis for a joint intervention to resolve them.

Hence, a client-system infrastructure was established between the developers and the researchers, allowing the former to switch roles from being DS researchers to being AR researchers. This effectively established an AR setting in which the AR researchers, with permission from the developers, implemented a CAR action cycle in the DSR project. The jointly-stated question was: *how should collaborative construction of architecture-intensive artefacts be pursued in order to continuously and incrementally deliver working functionality on time and with quality?* For the AR researchers the goal was both to contribute to the organization, by creating a sustainable improvement of development environment, and more generally, in line with Coghlan and Brannick [24], to improve the control and enhance the progress in the DSR project and develop sustainable knowledge for DSR as methodology.

5.3 Diagnosing the Halted Build Phase

Since time estimating was both central in this development environment and a familiar process to the developers, the AR researchers suggested that the estimated vs. actual time spent in the project predictions should be jointly analysed with the developers. The analysis clearly showed a large discrepancy between the two. This was thought to be caused by the estimation process used [29], collectively carried out during a four-hour meeting (for all activities). However, a lot of the difficulties in achieving the objectives emerged during implementation in the sprints. The absence of a thorough technical analysis was also thought to be one important cause to the halted build by all parties.

This was linked to a third cause, the fact that activities in the sprints were too large and not sufficiently defined. As a result, these became hard to follow-up in the daily scrums and potential warning signs were hence not identified. By dividing the activities into smaller entities (e.g. not exceeding 10 hours) it was believed it could be possible to both achieve better control on actual progress and potential obstacles, as well as increase the developers' work satisfaction through a sense of completion of smaller tasks carried out on a regular basis. This shift also implied that an on-going definition of tasks was encouraged, not only during the sprint planning, but also during the sprints themselves in order to stimulate technical analysis.

These changes guided the AR researchers to formulate a working hypothesis to guide the planning of the AR intervention: Continuous technical analysis, additional detailed activity scoping and more thorough follow-up's are achieved during collaborative construction of architecture-intensive artefacts via jointly reconfigured and appropriate ways of working in the project. This working hypothesis, grounded in the causes, of how to resolve the situation was presented to the developers and an agreement was reached to act upon it. At this stage the developers also underlined the importance that a reconfigured method should stimulate even more researcher-developer collaboration during the planning and the follow-ups. According to them, this was necessary in order to stimulate participation, shared responsibility and amplified technical analysis during the sprints. The intervention should therefore be done in a jointly fashion and result in a reconfigured method that in itself facilitated enhanced collaboration during the continuation of the build phase, redistributing control from the researchers to the researchers and the developers. Based on this comment, a revision was made in regard to the working hypothesis: Continuous technical analysis, additional detailed activity scoping and more thorough follow-up's are achieved during collaborative construction of architecture-intensive artefacts via jointly reconfigured and appropriate ways of working, which should facilitate the interplay between researchers and developers during build.

5.4 Action Planning: The Build Phase Improved

Equipped with an agreed-upon working hypothesis, the AR researchers were charged with the responsibility to plan the intervention. The researchers' first step was to apply activity theory [31] in order to grasp the notion of work from an existing conceptual framework. Activity theory states that subjects perform activities through tools in order to achieve objectives [32]. The causes elicited implied a need to reconfigure the method and this was perceived by the researchers as equivalent to tools in activity theory. Engeström states that in addition to tools, rules must be set up and used, as well as a division of labour which organises the collective accomplishment of the activity [33]. Therefore, the researchers planned the intervention so that it contained the development of a tool, rules and a revised division of labour (a reconfiguration of Scrum) to facilitate technical analysis, detailed activity scoping and more thorough follow-up's.

The tool should assist the splitting up of defined activities into smaller time-estimated actions. Consequently, a tool providing this functionality was set up and named the *sprintbook*. It consisted of one sheet in MS Excel for each Sprint. The specific sheet was divided into four sections with specific areas of concern for each section. By using the sprintbook, the researchers aimed to 1) separate and document larger activities into smaller ones, 2) keep track of the different prioritizations, and 3) distribute different activities to the participants in the DSR project.

In order to facilitate the interplay throughout the planning and follow-up meetings, additional rules and a revised division of labour was designed. The researchers turned to Conklin [34] for the theoretical principles in designing the rules. Conklin argues that collaboration in a social setting can be improved through collaborative displays constituted by a *notation*, an infrastructure and a facilitator.

The aim with this part of the intervention was to plan a more collaborative way for the whole development team to jointly transfer the agreements recorded in the sprint-book to individuals in the team during the Scrum meetings. The notation developed for this task included a set of simple rules, depicted in figure 2, and a collaborative display for documentation, in this case an ordinary whiteboard.

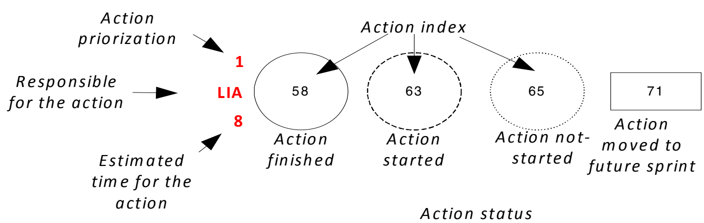

Fig. 2. Developed notation-in-action

The notation-in-action was consequently intended as a view over the same content that was available in the sprintbook. It was created to 1) make the agreements in the Sprint comprehensible for all participants at the same time, 2) to provide means to facilitate the joint tracking of achievement during the collaborative construction, and 3) to facilitate dynamically and cooperatively adjustments in the plan, based on any change of circumstances during the Sprint.

The third part in the reconfiguration of the method was the adjustment of the *division of labour* during the Scrum meetings. This part of the intervention was aimed at reconfiguring how meetings were conducted when the sprintbook was used together with the collaborative display. These aimed to assist the team to manage the meetings with a focus on potentially splitting up activities, estimate time, assign actions to individual developers, visualize the road to the sprint deadline and, also, to track progress in the project. Consequently, the combinations of these actions were designed to overcome the causes which had inflicted the project problems.

The new way to co-facilitate the meetings stated that one of the Scrum masters should facilitate the dialogue via the sprintbook. That amounted to asking the first Scrum-prescribed question to the developer in focus (What did you do yesterday?), based on the information in the sprintbook. As a response to the developer's answer, the second Scrum master had to relate the reply to the collaborative display on which the path to the sprint deadline was laid out. If changes in relation to the plan had occurred, this resulted in a proposed modification of the visualization of the Sprint. If the change was considered relatively trivial, an estimate was given upfront during this discussion. However, if the nature of the task was too complex to properly estimate during the meeting, a new task covering the needed technical analysis had to be brought up as an explicit task and its consequences on the sprint planning became immediately evident to the whole team. In this way, new unanticipated modules that needed to be constructed became explicit artefacts rather than just undefined subparts of a larger task (and for the designated developer to have in mind) as had been the case in previous sprints. When the group reached an agreement about the modification, then the first Scrum master reorganized the sprintbook and asked the next question (What will you do today?). This interaction circle was then repeated.

The revised division of labour consisted then in both new individual tasks for the two Scrum masters as well as a new structured interplay between the DS researchers and the developers, redistributing the control amongst them throughout the artefact construction.

5.5 Action Taking: The Build Phase Continued

The AR researchers reconfigured the method as described in section 5.4. They then notified the developers that a preliminary solution was ready to be used in the fourth sprint. The developers agreed that the reconfigured way of working was in line with their conception of the problems as well as the formulated suggestion (working hypotheses) and that it should be used during the sprint. The agreement to continue the build phase stated that the experiences from using the solution should be accumulated by the participants and discussed every week during the Scrum meetings. Changes in the method-in-use were consequently jointly identified, agreed-upon and implemented in the process, and put to effect during the following week, thus securing momentum to the DSR project. At the end of the sprint a major Scrum retrospective [29] was done to assess the progress in relation to the overall plan, as well as to evaluate the overall effects of the reconfigured method.

5.6 Evaluation

The retrospective clearly demonstrated that the intervention had effectively addressed the problems that halted the DSR efforts. Technical analysis was conducted with higher frequency by the developers in order to construct the functionality on a conceptual level before implementing it in the architecture. The development had now also included a more thorough activity scoping and although time estimations sometimes differed from actual time spent, warning signs were now early identified and each sprint delivered functionality for testing as planned.

Since the evaluation of the action taking was made a part of the system development work, it also became a natural part of the ongoing project. This unremitting evaluation played an important role in the intervention to 1) transfer the reconfigured method into daily work, 2) to evaluate its effects, and 3) to identify requirements for refinements. By executing these activities action taking was performed to secure the CAR principle of learning through reflection. In all, this implies that the work done in the action taking was guided by the CAR principle of change through action and has accomplished sustainable change.

The principle of theory in CAR influenced the intervention in two ways. After the joint diagnose of the causes which resulted in the breakdown, the AR researchers used Scrum methodology together with activity theory as a conceptual framework to grasp the constituents of an action. By using this theory generic categories became the foundation for planning the intervention as a whole. The reconfiguration of the method, as stated in the working hypothesis, was consequently directed with tools as a phenomena (the sprintbook and the collaborative display), rules as a phenomena (the logic in the sprintbook and notation used on the collaborative display) and division of labour during the meetings (the team model for co-facilitation). The logic in the sprintbook was based on experiences from the project and on the diagnosis of the problems at hand and it was then used as a theory together with collaborative theory to design the collaborative display [34]. The division of labour was at the same time based on classification of forms for co-facilitation [35]. Experiences from diagnosing the problem situation was induced and transformed into parts and logic for the reconfigured method, especially during the design of the sprintbook.

The principle of the researcher-client agreement came into effect when the initial client-system infrastructure was established. The agreement was however revised after this phase. The new agreement stated that the AR researchers themselves should plan the intervention and transform the working hypothesis into a reconfigured method which should resolve the temporary breakdown and stimulate improved collaboration. The developers did not want to participate in this preparation. One explanation is that the DS researchers, now acting as AR researchers, also acted as Scrum masters. Their perceived responsibility, from the developers point of view, was to provide methods for managing the project when they acted as Scrum masters. As systems developers they wanted suggestions from the researchers on concrete solutions, in line with the working hypothesis, to the problems that halted the build process. The AR framework gave the DS researchers a means to meet that expectation.

One important validity claim in AR is that the effects of the actions taken in the AR project should be sustainable or even irreversible [3]. Auditing the performance in the project provides convincing indications that the changes made via the AR intervention indeed became sustainable and that the improved results produced in sprint 4 and sprint 5 were carried on throughout the following sprints. Used in this way the AR framework worked as a structure to create sustainable changes in the development team.

5.7 Specify Learning

The learning outcomes from the AR cycle in this DSR project where fed into the revision of the reconfigured method, and became learning outcomes for this local practice. Significant learning outcomes were: 1) the joint planning of the Sprint was improved by the use of the collaborative display, the chosen co-facilitation form, and the designed workbook; 2) after the intervention, the technical analysis carried out through the reconfiguration of the method, was done more thoroughly as these analysis activities were now illustrated on the collaborative display and in cooperation assigned to individual developers who now to a higher degree a) felt responsibility to perform them, as well as after the intervention; b) could better understand the value the analysis brought in relation to later building activities in the sprint; 3) through the reconfigured method, the DS researchers could follow the progress in the DSR project much better, and as a result could identify signs of uncertainty and challenges much earlier on than before.

The contribution this project brings to the scientific body of knowledge is primarily that artefact development in DSR can be improved significantly by using AR as a means to enhance the researcher-developer interplay during build. In this authentic setting, CAR has proven to be a suitable framework to use in obtaining meta-design control during the build phase. An established framework, CAR, was used to diagnose the causes and take collaborative actions in order to improve researcher-developer collaboration in a authentic development environment. We argue that it is a valid structure to use for the establishment of a focused, but important, impact in an DSR project and at the same time results in findings which contribute to DSR as methodology.

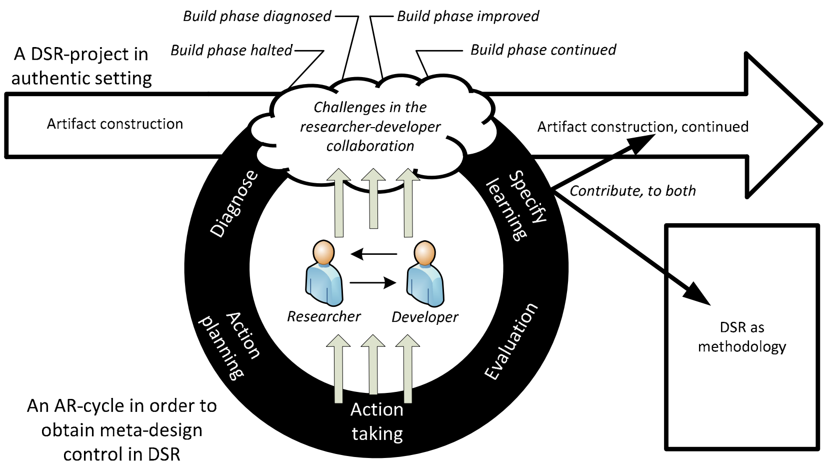

Fig. 3. AR as framework to obtain meta-design control in artefact construction and to contribute to DSR as methodology

6 Discussion: Using AR to Improve DSR When Designers Are Not in Control

This paper set out to explore and present experiences from managing challenges during collaborative construction of artefacts in DSR. AR has been used to both manage these challenges and develop the knowledge. As noted in the literature review, the discussion about AR/DSR has previously concerned intervention in a context similar to or actually where the artefact is to be used [4, 5, 9, 11, 13, 25] However, as identified in Markus et al. [7], researchers in DSR projects may not be developing and implementing technical artefacts by themselves, but rather participate in a researcher-developer collaboration to construct these artefacts. Moreover, the DSR community has not yet developed method components to support such DSR collaborations. As demonstrated by the empirical material, insufficient collaborative work procedures between researchers and developers indeed led to a halt in the progress, threatening the outcome of an entire DSR project. The data also shows that using AR to diagnose and actively change build activities led to sustainable improvements in a local DSR project.

As DSR is gaining increasingly wider acceptance and interest in the IS-field, both the background and research interests of the researchers in DSR become more and more heterogeneous. Although DSR has been identified to spring out of the engineering disciplines [36] there seems to be one expansion (among others) in recent years in IS DSR towards inquiring more socio-technical systems. This means that the researchers may have technology as one of many objects under study, compared to more traditional DSR projects having technical principles as their major focal point. In such multi-faceted research it seems likely that researchers in DSR projects, to different degrees, collaborate with specialists to modify the underlying technology. Since the researcher-developer interplay is mostly about communicating and transferring the researcher's design propositions to digital form, these two activities should be the focus of the rules governing this collaboration. We argue that the experiences presented in this paper add evidence to help understand this important interplay.

The research question in this paper was *how to obtain meta-design control in researcher-developer collaboration during artefact construction?*

Control was successfully obtained by redistributing it from the researchers to researchers and developers by means of an improved researcher-developer interplay facilitated by the results produced by the use of CAR. By using AR as framework, a resilient participatory design effort was thereby established to manage the problems that had occurred in DSR. In line with Fischer [17], our experiences is that meta-design control can be achieved when intended users (the developers and the researchers) of the results are brought into the process early and not misused during the design phase. Developers were a part of the CAR cycle, but were not forced to participate in all activities; i.e. in the action planning. The effort was characterised by a high degree of collaboration between developers and researchers: this was achieved through motivation, mutual support and sharing, and was also characterized by familiarity [24]. This also means that the development method was not completely changed or replaced. Instead the AR cycle resulted in *revised* method components based on the methodology already in place, which was well-known by the development team, adding a high degree of familiarity which allowed for the successful redistribution of control.

Hence, we conclude that AR, in addition to previously suggested uses [4, 5], may serve as a valuable and focused measure to be used throughout artefact development in authentic settings in order to develop wider knowledge about DSR as well as obtain control and enhance progress during DSR ... when designers are not in control.

References

1. Orlikowski, W.J., Iacono, C.S.: Research commentary: desperately seeking the "IT" in IT research: A call to theorizing the IT artifact. Information Systems Research 12, 121–134 (2001)
2. Au, Y.A.: Design science I: The role of design science in electronic commerce research. Communications of AIS 7 (2001)

3. Järvinen, P.: Action research is similar to design science. Quality & Quantity 41, 37–54 (2007)
4. Cole, R., Purao, S., Rossi, M., Sein, M.: Being proactive: Where action research meets design research. In: Proceedings for ICIS 2005, pp. 325–336 (2005)
5. Iivari, J., Venable, J.: Action research and design science research: seemingly similar but decisively dissimilar. In: Proceedings for ECIS 2009, Verona, Italy, June 8-10 (2009)
6. Gavish, B., Gerdes, J.: Anonymous mechanisms in group decision support systems communication. Decision Support System 23(4), 297–328 (1998)
7. Markus, M.L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quarterly 26(3), 179–212 (2002)
8. Susman, G., Evered, R.: An assessment of the scientific merits of action research. Administrative Science Quarterly 23, 582–603 (1978)
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MISQ 28, 75–106 (2004)
10. Vaishnavi, V.K., Kuechler Jr., W.: Design science research methods and patterns: innovating information and communication technology. Auerbach Pub. (2007)
11. Nunamaker, J., Chen, M., Purdin, T.D.M.: Systems development in information systems research. J. of Management Information Systems 7(3), 89–106 (1991)
12. Purao, S.: Design research in the technology of information systems: truth or dare. Pennsylvania State University (2002)
13. Venable, J.R.: A framework for design science research activities. In: Proceedings of the Information Resource Management Association Conference, Washington, DC, USA, May 21-24 (2006)
14. Gericke, A.: Problem solving patterns in design science research: learning from engineering. In: Proceedings for ECIS 2009, Verona, Italy, June 8-10 (2009)
15. Purao, S., Baldwin, C.Y., Hevner, A., Storey, V.C., Pries-Heje, J., Smith, B., Zhu, Y.: The sciences of design: observations on an emerging field. Harvard Business School Finance Working Paper No. 09-056 (2008)
16. Fischer, G., Giaccardi, E.: Meta-Design: A framework for the future of end user development. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End user development: empowering people to flexibly employ advanced information and communication technology, pp. 427–457 (2006)
17. Fischer, G.: Meta-design: expanding boundaries and redistributing control in design. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007, Part 1. LNCS, vol. 4662, pp. 193–206. Springer, Heidelberg (2007)
18. Lewin, K.: Frontiers in Group Dynamics. Human Relations 1(1), 5–41 (1947)
19. Blum, F.: Action research: a scientific approach? Philosophy of Science 22(1), 1–7 (1955)
20. Baskerville, R.L., Wood-Harper, A.T.: Critical perspective on action research as a method for information systems research. J. of Information Technology 11, 235–246 (1996)
21. Baskerville, R.L., Wood-Harper, A.T.: Diversity in information systems research methods. European J. of Information Systems 7(2), 90–107 (1998)
22. Susman, G.: Action research: a sociotechnical perspective. In: Morgan, G. (ed.) Beyond method: strategies for social research, pp. 95–113. Sage Publications, Thousand Oaks (1983)
23. Davison, R.M., Martinsons, M.G., Kock, N.: Principles of canonical action research. Information Systems Journal 14, 65–86 (2004)
24. Coghlan, D., Brannick, T.: Doing action research in your own organization, 3rd edn. Sage Publications, Thousand Oaks (2010)

25. Lee, A.: Action is an artifact: what action research and design science offer to each other. In: Kock, N. (ed.) Information systems action research: an applied view of emerging concepts and methods, pp. 43–60 (2007)
26. March, S.T., Smith, G.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
27. Albinsson, L., Lind, M., Forsgren, O.: Co-design: an approach to border crossing, Network Innovation. In: Cunningham, P., Cunningham, M. (eds.) Expanding the knowledge economy: issues, applications, case Studies, pp. 977–983. IOS Press, Amsterdam (2007)
28. Lind, M., Albinsson, L., Forsgren, O., Hedman, J.: Integrated development, use and learning in a co-design setting: experiences from the incremental deployment of e-Me. In: Cunningham, P., Cunningham, M. (eds.) Expanding the knowledge economy: issues, applications, case Studies, pp. 773–780. IOS Press, Amsterdam (2007)
29. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice Hall PTR, Upper Saddle River (2001)
30. von Hippel, E., Katz, R.: Shifting Innovation to users via Toolkits. Management Science 48(7), 821–833 (2002)
31. Kaptelinin, V., Nardi, B.A.: Acting with technology: activity theory and interaction design. The MIT Press, Cambridge (2006)
32. Kuutti, K.: Activity theory and its applications to information systems research and development. In: Nissen, H.-E., Klein, H., Hirschheim, R. (eds.) Information Systems Research: Contemporary Approaches and Emergent Traditions, North Holland, Amsterdam, pp. 529–549 (1991)
33. Engeström, Y.: Activity theory and individual and social transformation. In: Opening address at the 2nd International Congress for Research on Activity Theory, Lahti Finland (1990)
34. Conklin, J.: Dialouge mapping: building shared understanding of wicked problems. Wiley, Chichester (2006)
35. Hogan, C.: Understanding facilitation: theory and principles, p. 36. Kogan Page (2002)
36. Kuechler, W., Vaishnavi, V.: The emergence of design science research in information systems in North America. J. of Design Research 7(1), 1–16 (2008)