Demonstrating the elaborated action design research (eADR) model to address communication challenges during software development

Anthea Venter[1] and Marne de Vries[2*]

[1] Department of Industrial and Systems Engineering; University of Pretoria; South Africa; venter.anthea@gmail.com; +27 72 560 3344

[2*] Department of Industrial and Systems Engineering; University of Pretoria; South Africa; Room 3-20; marne.devries@up.ac.za; +27 82 822 1384

Abstract Communication is a powerful, universal, everyday activity that impacts many, if not all, aspects of our lives. The *importance* and *complexities* of communication are however still often overlooked. Inadequate communication during software development projects at a medium-sized enterprise, Company-GIS (CGIS), caused delays in project tasks, negatively impacting quality, on-time delivery, and delivery within budget. In this study, we argue that a collaborative methodology was required to ensure adequate diagnosis of the problem, creating multiple artefacts to address the communication problems. We argue that a recent research methodology called action design research (ADR) is best suited for a practice-inspired problem. Acknowledging the latest developments within the ADR discipline, we believe that the *elaborated action design research* (eADR) process model addressed the need to explicitly show our knowledge contributions in the form of two artefacts: (1) An *Agile requirement engineering solution* (ARES) to address the communication problems, the primary artefact; and (2) A *taxonomy* of perceived communication problems, a secondary artefact. In this experience report, we indicate that the eADR process model was instrumental in facilitating diagnosis, followed by design, implementation, and evolution of the primary artefact, the ARES. The practical demonstration of ARES also highlighted the need for further evolution of the eADR as a method artefact. Based on our reflections on applying the eADR, we suggest changes to the eADR including an additional design cycle phase, *suggestion*, between *problem formulation* and *artefact creation*, as well as merging *reflection* and *learning* into a single phase.

Keywords Action design research, agile, communication, project management, requirements engineering, software development

Introduction

At the start of a research project, the researcher has to select the most appropriate research methodology to answer a primary research question. Once the researcher has decided on using a design-based approach where a solution needs to be *designed* to address an existing problem, three main options are available to the researcher: (1) Action Research (AR), (2) Design Science Research (DSR), and (3) Action Design Research (ADR), where ADR emerged as a new research methodology to combine the strengths of AR and DSR (Sein et al. 2011). The action research paradigm and the design-based paradigm differ primarily based on their *emphasis* on the kind-of-knowledge that is created (Collatto et al. 2018; Sein et al. 2011).

The *kind-of-knowledge* for pure AR relates to the two main objectives of AR, namely intervention/change in practice and production of knowledge (Collatto et al. 2018). For AR, the intervention need not involve the rigorous design of a solution artefact, but the focus is rather on *change* or *improvement* within an enterprise context that allows for the *generation of learning* (Coughlan and Coghlan 2002).

The design-paradigm, evident in DSR and ADR, seeks to generate knowledge on *designing one or more artefacts* or *prescribing a solution* (Dresch et al. 2015). March and Smith (1995) define four types of prescriptive knowledge, namely constructs, models, methods, and instantiations. *Constructs* provide the vocabulary and symbols used to define and understand problems and solutions. *Models* are designed representations of the problem and possible solutions, e.g., mathematical and diagrammatical models, whereas *methods* are algorithms, practices, and recipes for performing tasks. *Instantiations* are the physical realisations that act on the natural world, such as an information system that stores and retrieves data. A *design theory*, which is an abstract, coherent body of prescriptive knowledge describes the principles of form and function, methods, and justificatory theory that are used to develop an artefact, and can include constructs, models, methods, and instantiations. Wieringa (2014) provides a practical definition of an artefact, also clarifying what an artefact is *not*. He indicates that a wide variety of useful things that can be *designed* may be classified as artefacts. Elements, such as people, values, desires, fears, goals, and norms are *not* artefacts, since they cannot be designed by the researcher, but rather become part of the problem *context*. Wieringa (2014) emphasises the complex interplay between an artefact and its context, and the researcher has to study this interaction, rather than the artefact alone.

In ADR, the development of the artefact has to achieve the objectives that were originally proposed by the *researcher* (Dresch et al. 2015), focusing on generalisation of solutions to address a certain class of problems (Sein et al. 2011). The *evaluation* of the artefact is therefore essential to analyse the artefact's ability to meet the intended objectives (Dresch et al. 2015) and its utility when applied in the external environment for which it was designed (Hevner et al. 2004). Within the design-paradigm, DSR does *not require collaboration* of people that are interested in the research when the designed artefact is tested, which ensures that the researcher performs technology decisions in a rigorous way (Iivari and Venable 2009). Sein et al. (2011) highlighted the need for *joint collaboration* of researchers with participants in the environment in which the research is conducted when an ensemble of artefacts are designed to address an existing class of problems. A new research approach emerged, called action design research (ADR) to combine the strengths of two existing approaches. ADR combines the collaborative approach of AR to ensure *organisation relevance* and the systematic design approach of DSR to *ensure technological* rigour in designing and evaluating novel artefacts (Sein et al. 2011).

Collato et al. (2018) conducted a systematic literature review to determine whether a new, combined approach, i.e. ADR, is indeed necessary. Their research indicated that existing researchers differ on how they use AR and DSR in combination. Their research highlighted three possibilities of combining AR and DSR: (1) Using AR under the DSR paradigm; (2) Using AR and DSR as mixed methods; and (3) Using ADR as a new method, i.e., a standalone research method in its own right. Searching for publications with keywords "action research" associated with "design science" OR "design science research" OR "prescriptive research" and the specific words "action research design" and "action design research", they extracted 43 sampled articles from the databases EBSCOHost, ScienceDirect and Google Scholar from 2007 to 2017 (Collatto et al. 2018). From the sampled articles, 13% applied ADR as a new method. Even though ADR has been proposed for particular disciplines, such as the emerging enterprise engineering discipline (De Vries and Berger 2017), Collato et al. (2018) suggested that future studies should propose a more detailed method to operationalise ADR. Mullarkey & Hevner (2019) suggested that the existing ADR methodology of Sein et al. (2011) could be further extended to emphasise that multiple intervention cycles may exist for an ADR study. Their main objective was to make more explicit the knowledge generation in ADR studies, claiming that the elaborated ADR or eADR process model could better support users to structure key decisions and activities necessary to rigorously apply ADR (Mullarkey and Hevner 2019).

In this experience report, we indicate that the eADR model by Mullarkey and Hevner (2019) was instrumental in facilitating diagnosis, followed by design, implementation, and evolution of the primary artefact of our study. In addition to this, we provide suggestions on how the model can be modified based on our reflections on applying the eADR in our study.

Next, we present the structure for the remaining sections of this article. The section called *Theoretical Background on eADR* introduces the eADR process model of Mullarkey & Hevner (2019) which includes four design stages, namely Diagnosis, Design, Implementation, and Evolution, as well as five design cycle phases that are based on Sein et al.'s (2011) work. Examples of studies that have since applied the eADR abstraction by Mullarkey and Hevner (2019) are included in the *Applications of eADR* section. The *Objectives of the Experience Report* section includes the purpose and approach of our report. Applying the eADR process model to our study, the subsequent sections indicate that a single cycle of the *Diagnosis* stage was needed for our study, followed by a single cycle for the *Design* stage and a single cycle for the *Implementation* stage. The section named *Reflections* provides a synthesis of our collaborative learning in applying the eADR process model. In the *Conclusion* section, we suggest that the eADR process model is also applied to our existing research project during the evolution stage to further evolve the newly developed artefacts.

Theoretical Background on eADR

ADR combines the activities of AR and DSR, constructing an artefact to address the problem and addressing a problem in a specific organisational setting (Sein et al. 2011). ADR therefor addresses both technological rigour and organisational relevance. The original ADR process model suggested by Sein et al. (2011) identifies four phases, namely (1) Problem formulation, (2) Building, intervention, and evaluation (BIE), (3) Reflection and learning, and (4) Formulation of learning. In ADR, researchers in collaboration with participants *identify a problem* and suggest a theory-ingrained *artefact or course of action*, *design/demonstrate/intervene/evaluate* the artefact or course of action, *reflect*, and *communicate the formalised results*. Since the artefact or course of action evolves within a real-world context Sein et al. (2011) also emphasize that the first three phases are highly interactive and iterative in nature.

Building on these original ADR concepts, the eADR process model, included in Fig.1, has been abstracted from a real-world study performed by Mullarkey and Hevner (2015). The nature of their main artefact is an information system. Mullarkey and Hevner (2019) re-defined the ADR phases presented by Sein et al. (2011), defending four intervention stages: (1) Diagnoses, (2) Design, (3) Implementation, and (4) Evolution. The goal of the Diagnosis stage is to analyse the importance of the problem domain, and identify the relevant kernel design theories, any existing socio-technical artefacts, and the goals of the ADR project. The second key stage, Design, focuses on the identification and conceptualisation of the proposed artefact's design, which should address the problem class identified. The third stage, Implementation, supports the instantiation of artefact(s) through an implementation and is focused on building and evaluating the artefact as a real-life intervention. Lastly, the artefact should evolve as the problem environment changes. Sein and Rossi (2019) agree that the cyclical representation of ADR is a worthwhile elaboration, and state that the explicit Diagnosis stage is a valuable extension, since problem diagnosis is the first step to understanding what needs to be solved.

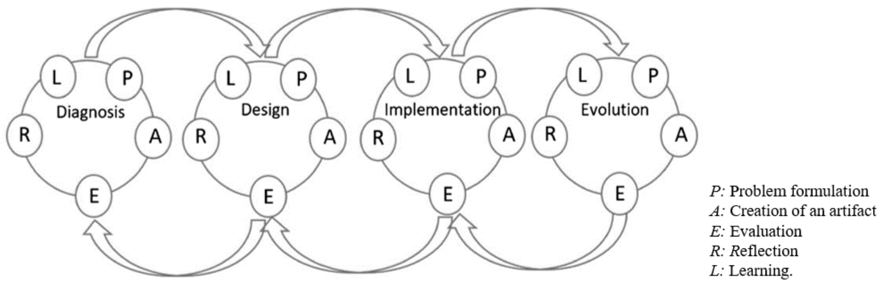

Fig.1 The eADR process model of Mullarkey and Hevner with PAERL phases (2019)

While Sein et al. (2011) clearly delineate the different phases of ADR, details on the phases are not provided and the BIE engagement is left open to interpretation by the researcher (Mullarkey and Hevner 2019). Sein and Rossi (2019) state that this high level of abstraction of the method was deliberate, allowing for the actual application of the method to emerge in use. Mullarkey and Hevner (2019) show that each stage consists of a cycle of (1) Problem-formulation, (2) Artefact development, (3) Evaluation, (4) Reflection, and (5) Learning. Therefore, Mullarkey and Hevner (2019) propose that an artefact should be built and evaluated in every ADR stage, and that reflection and learning take place during each cycle of an ADR stage. The artefact in each cycle is built and evaluated to address the problem formulated in that ADR cycle and each iteration learns from prior cycle(s) and modifies the problem formulation. Sein and Rossi (2019) believe that Mullarkey and Hevner's (2019) paper offers excellent suggestions to make all the stages of ADR transparent by including specific activities for performing research tasks, making the ADR method more accessible to researchers.

In addition to the redefined ADR stages and phases, Mullarkey and Hevner (2019) found that the presence of multiple entry points with clear definitions allows the team to investigate their project goals and identify the entry point that best aligns with the project's purpose. They argue that the abstraction of artefacts at all ADR stages informs research and practice and that the nature of innovative artefacts is often different depending upon the ADR stage involved. Sein and Rossi (2019) however disagree with the proposed enhancement of multiple entry points to ADR, stating that ADR's essential spirit is that there is one entry point to the cycle, namely, problem formulation. They are of the opinion that Mullarkey and Hevner have not made a convincing case for combining Peffers et al.'s (2007) framework with ADR and state that by combining two approaches that are epistemologically incommensurate has implications for the legitimation of knowledge claims for researchers following either process. The Diagnosis stage, as depicted in Fig.2, was the initial point of entry in our study. According to Mullarkey and Hevner (2019) this entry point is problem centred.

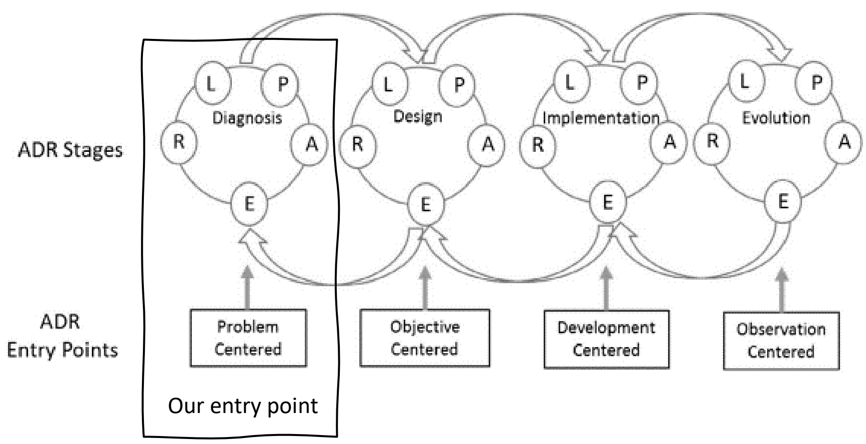

Fig.2 The eADR process model of Mullarkey and Hevner (2015), indicating this study's entry point

In the following section we include examples in which the eADR abstraction by Mullarkey and Hevner (2019) has since been applied to other studies.

Applications of eADR

Studies that applied the eADR and had a *problem-centred* entry point include Bergan et al. (2020) that identified a need, and conceptualised and implemented *organisational structures and governance processes* to facilitate the digital transformation of a city, whereas Gimpel et al. (2020) conducted one diagnosis and three design cycles to create *a set of work system transformation principles* to provide guidelines for practitioners aiding in setting up and conducting digital transformation initiatives. Daniel et al. (2018) diagnosed, designed, and implemented *an artefact to analyse social video network content* and found that the iterative approach of the eADR assisted in developing multiple innovative artefacts towards an implementable social video content analysis solution.

Mullarkey et al. (2019) used eADR to diagnose, design, and implement *automated systems for the repeatable synthesis of data into insights* that aided daily, weekly, and monthly decision-making by creating artefacts such as *data sets, guidelines, decision trees, and dashboards*, and found that the eADR provided a significant benefit in its focus on the abstraction and evaluation of artefacts during fast, flexible, iterative investigation efforts. Venkatraman and Sundarraj (2022) also had a *problem-centred* entry point, eventually creating three artefacts to *assess the healthcare analytics readiness of hospitals.* Due to limitations after the design of their pilot artefact, they required a deeper understanding of the problem space with a broader set of cases and more hospitals. Once the problem was better understood, a new artefact was designed, implemented, and evaluated. Kutz et al. (2022) used eADR to design and implement a *virtual team classroom environment* artefact, promoting effective student teamwork by considering and actively engaging the student. During Evaluation of the implemented artefact, they identified dissatisfaction with the variety of tools required in the solution, which warranted an additional cycle of eADR. According to Kutz et al. (2022), the eADR process enabled them to yield *non-obvious diagnoses and actionable steps for continually incorporating the ever-changing social aspects unique to students*.

Van Wyk et al. (2020) and Dixon and Sundaraj (2020) applied the eADR and used an *objective-centred* entry point. Van Wyk et al. (2020) designed, developed, and evaluated a *prototype of an incident management system* to support sensemaking of emergency incidents and their immediate environment with the use of digital technology, and Dixon and Sundaraj (2020) designed a *game* as an artefact to detect and perform containment actions on a cybersecurity threat.

Tarpey and Mullarkey (2019) re-designed a *Labour Planning and Allocation Cycle (LPAC) system* i.e., a work and organisational structures/systems artefact, and started with *problem diagnosis*. The eADR method applied to the LPAC model development, however, differed from the model provided by Mullarkey and Hevner (2019), as presented in Fig.1. For the LPAC, the stages consisted of Problem Diagnosing, Concept Design, Build, and Implement, and the phases consisted of *evaluate*, *reflect*, and *intervene* as depicted in Fig.3.

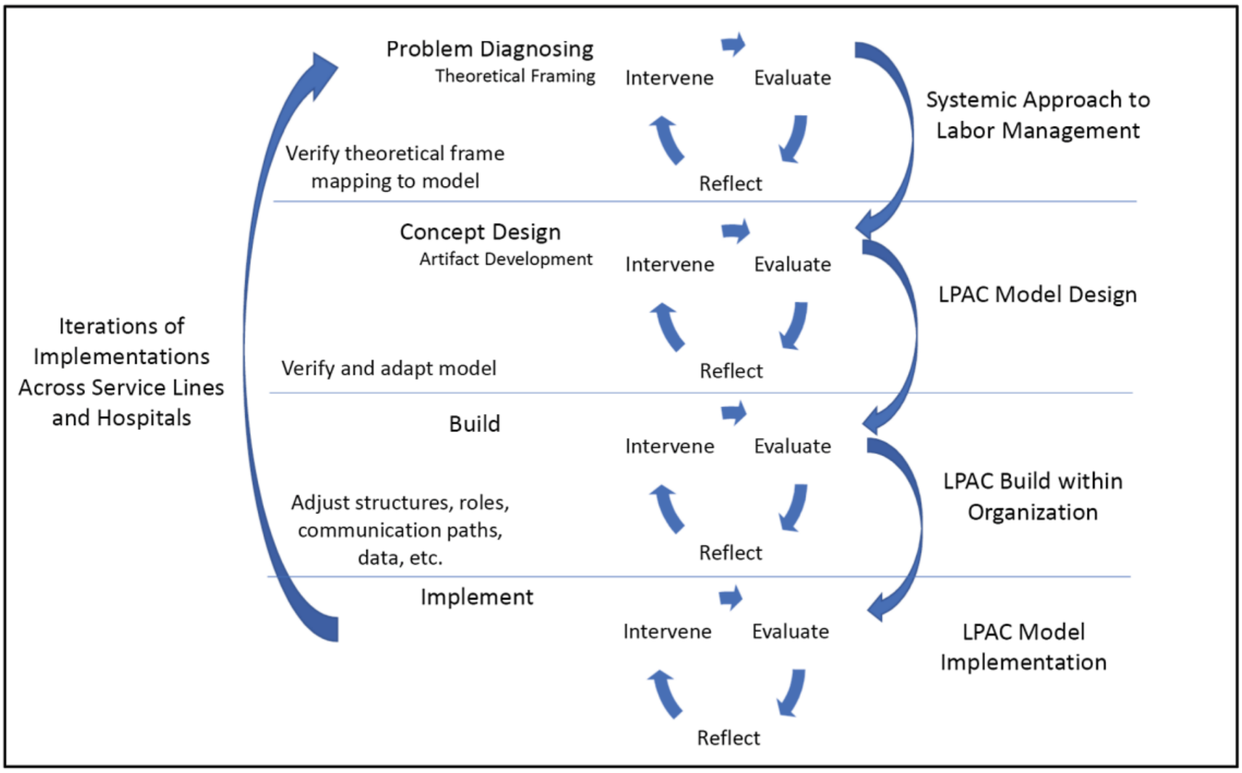

Fig.3 The eADR method applied to LPAC model development, from Tarpey and Mullarkey (2019)

In the following section we discuss the objective of our research as well as the approach followed to report on our experience using the eADR model to create a method-artefact as the main artefact.

Objectives of the Experience Report

Our study identified a real-world problem at Company-GIS (CGIS), a medium-sized enterprise that experienced communication challenges with their software development projects. Since a collaborative approach was needed in co-designing a solution that only addresses the problems at CGIS, but also contributing to the existing knowledge base, addressing a general class-of-problems, ADR was selected as an appropriate research methodology. Using the initial version of ADR proposed by Sein et al. (2011) the contributions of the study were under-represented due to the simplified four-phase structure of the ADR method, which consists of (1) Problem formulation; (2) Building, intervention and evaluation; (3), Reflection and learning; and (4) Formalisation of learning. Most of our research efforts were expended during the problem formulation phase, but the contributions added during this phase are not considered to be major contributions within the four-phased model. The elaborated action design research (eADR) process model of Mullarkey & Hevner (2019) was therefore useful in highlighting our contributions during the problem formulation stage.

For our study, the main artefact referred to as the *Agile requirement engineering solution* (ARES), is a set of practices combined within an existing Agile software development framework called Scrum (Scrum.org n.d.-a; Scrum.org n.d.-b), to address communication problems at CGIS.

Study objective: *Since the eADR process model is relatively new, with limited demonstrations, additional empirical research is needed to demonstrate its usefulness.*

Our study, presented as an *experience report*, provides additional evidence regarding the usefulness of the eADR process model by *practically* demonstrating how we used the eADR model to develop a method-artefact as the main artefact, also within a different context and where a different kind of artefact is developed.

In the following sections we elaborate on how three of the four different main eADR stages, namely Diagnosis, Design, and Implementation, were applied in the study conducted at CGIS to gradually develop multiple artefacts or artefact versions through these three stages instead of combining activities into one single BIE stage as suggested by Sein et al. (2011). We indicate the four main stages by using capitalisation, e.g., Diagnosis, Design, Implementation, Evolution. When we reference the five phases that repeat per stage, we use italics, e.g. *problem formulation*, *artefact creation*, *evaluation*, *reflection* and *learning*. The Evolution stage was excluded from the study, since it forms part of future work, as discussed in the *Conclusions and Future Research* section. The demonstration content was extracted from the original study, packaged by Venter (2022).

Diagnosis Stage

During the Diagnosis stage, the main researcher, also a participant-observer at CGIS, identified a real-world problem at CGIS. To understand the identified problem, the researcher conducted individual in-depth interviews with seventeen practitioners at CGIS. Focus was placed on *project management* concerns due to unacceptable project delays and the interviews conducted therefore included questions related to the process of executing software development projects at CGIS.

The main researcher obtained synthesized root cause analysis results by creating a current reality tree (CRT) based on the individual interviews. The diagnosis, which was evaluated with CGIS practitioners in focus group sessions, revealed that project tasks take longer than expected, negatively impacting on-time delivery, quality of delivery and delivery within budget. *One* of the seven root causes identified was *inadequate communication*. In an enterprise consisting of ±90 employees, with all team members predominantly based at the same office space in a South African city, it was a concern that communication was neglected, and misalignment frequently occurred. Inadequate communication was a prominent theme. The main researcher also observed similar communication challenges on a software development project while working at a different, larger enterprise within a different industry, indicating that a larger class-of-problems might exist. In search for guidance on an appropriate artefact to adopt or adapt as a solution, the main researcher conducted a systematic literature review (SLR).

An SLR was conducted to identify relevant peer-reviewed articles, following the guidelines from Okoli (2015). The review took place between 1 September 2020 and 31 October 2020 and a list of relevant studies was identified. The researcher thematically analysed the results and determined that *inadequate communication* is a problem that exists within the broader software development sector. The SLR indicated that general terms were used for *inadequate communication* in the literature. A similar finding was identified during the in-depth interviews that were conducted during the initial problem diagnosis at CGIS, which hampered a comprehensive and unambiguous understanding of the *inadequate communication* problem. Examples of terms that were used interchangeably, included *communication problems* and *communication gaps.*

The findings during further diagnosis did not assist in extracting an artefact as a solution from the existing body of knowledge, that could be adopted or adapted at CGIS in addressing the *inadequate communication* problem. Of the SLR studies that referred to inadequate communication during a specific phase, 93% identified the requirement elicitation/analyses phase as problematic. Although the requirements elicitation phase was highlighted as a prominent theme in literature, also in terms of communication, we had to ensure that we also identified other possible contributors towards inadequate communication.

After three iterations of the *problem formulation* phase in the Diagnosis stage, as depicted in Fig.4 , it was determined that an artefact was needed to improve the rigour of diagnosing the problems at CGIS instead of prematurely suggesting a solution artefact to CGIS. An *artefact*, i.e., a taxonomy of perceived communication problems, would be useful to further diagnose the problems at CGIS. Yet, this artefact had to be designed, since it did not exist in the current body of knowledge.

For our CGIS-study, we had to perform multiple iterations of problem investigation and problem re-formulation. For every iteration, a new insight about the problem leads to further problem investigation during the next iteration. Fig.4 indicates that multiple iterations of problem investigation generated *problem re-formulation* for each iteration. Currently the eADR does not emphasise the iterative nature of problem investigation.

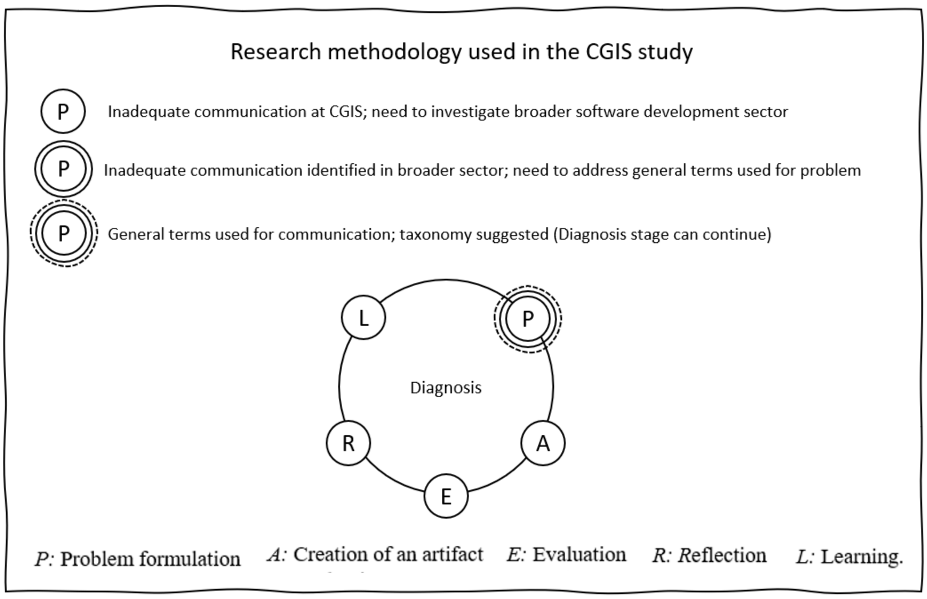

Fig.4 Problem formulation phase iteration, within the Diagnosis stage

We followed both a deductive and inductive approach in developing a taxonomy as an *artefact*. Using the four types of prescriptive knowledge, identified by March and Smith (1995), the *taxonomy* artefact can be classified as a construct, since it provides a vocabulary used to define and understand problems. Littlejohn and Foss' book (2005) on the Theories of Human Communication was used as an authoritative source on communication. We used the main categories of their book as the main categories for an initial theory-ingrained taxonomy to identify different types of *inadequate communication*.

To identify and understand the *types* of communication problems at CGIS, an additional round of interviews was conducted with fifteen CGIS employees. Using thematic analysis of the interviews, the researcher extended the initial taxonomy into a rudimentary taxonomy of the *perceived* communication problems at CGIS, as included in Fig.5. Note that the codes in the taxonomy presented in this report are slightly different to the original diagram. This was solely done to improve readability in this article.

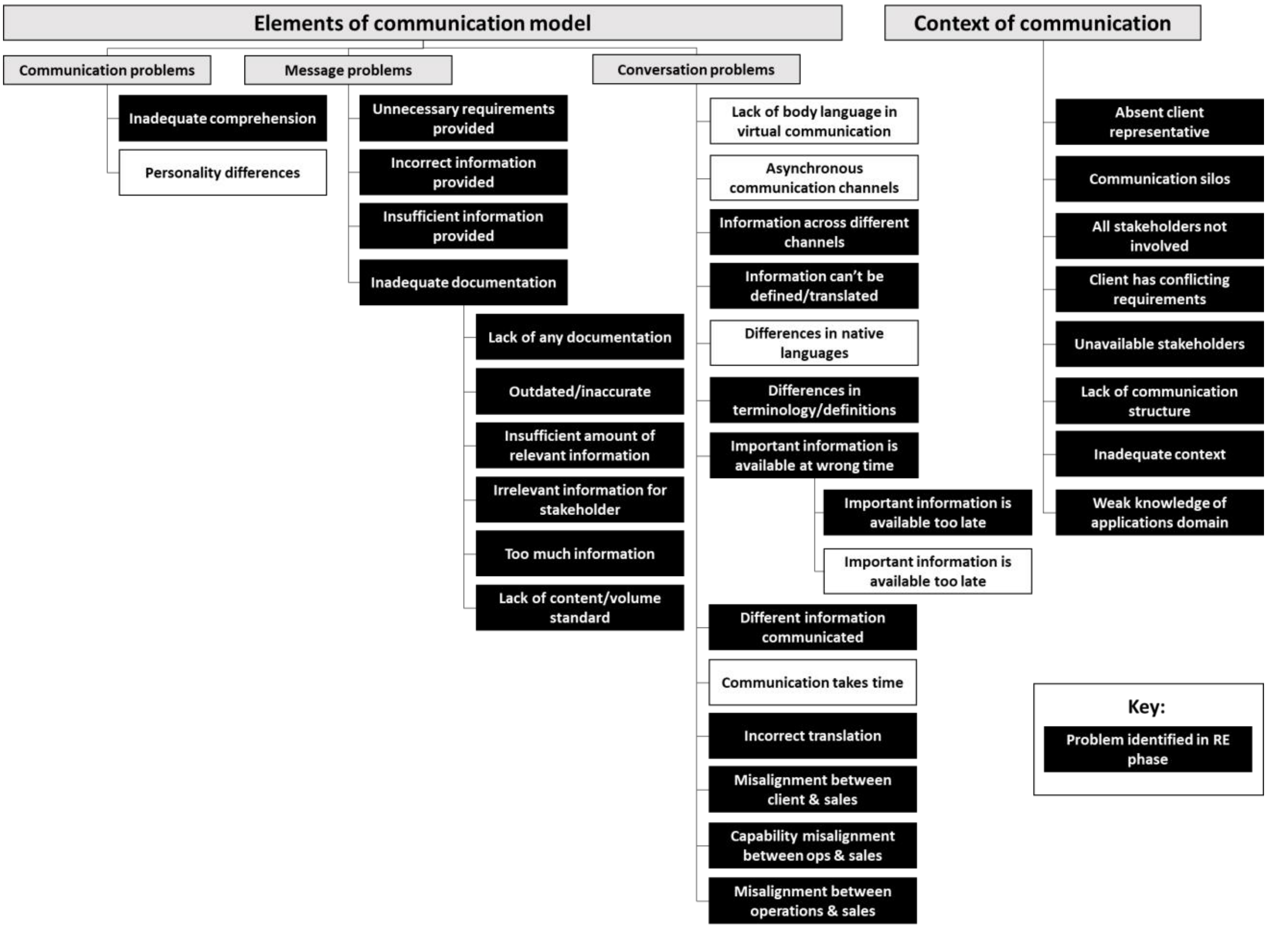

Fig.5 The rudimentary taxonomy of perceived communication problems at CGIS

The rudimentary taxonomy, created as an artefact, would enable the researcher to construct a solution artefact during the Design stage. The theory-ingrained *taxonomy* was created based on the collaboration between the researcher and practitioners at a real-world enterprise. With reference to the remaining phases conducted as part of the Diagnosis stage, namely *evaluation*, *reflection* and *learning*, Fig.6 provides a summarised interpretation of the remaining phases. The taxonomy was not *evaluated* by CGIS practitioners or other practitioners/enterprises since the taxonomy was not created as the primary artefact of our study. For future work, we recommend that the taxonomy should be validated at other enterprises that also experience communication problems during software development projects. The taxonomy assisted during the *reflection* phase, regardless of the evaluation phase being omitted, and indicated that requirements elicitation was indeed a concern regarding communication at CGIS. This *learning* provided the necessary focus when searching for a solution artefact within the requirements elicitation knowledge base of software development during the Design stage.

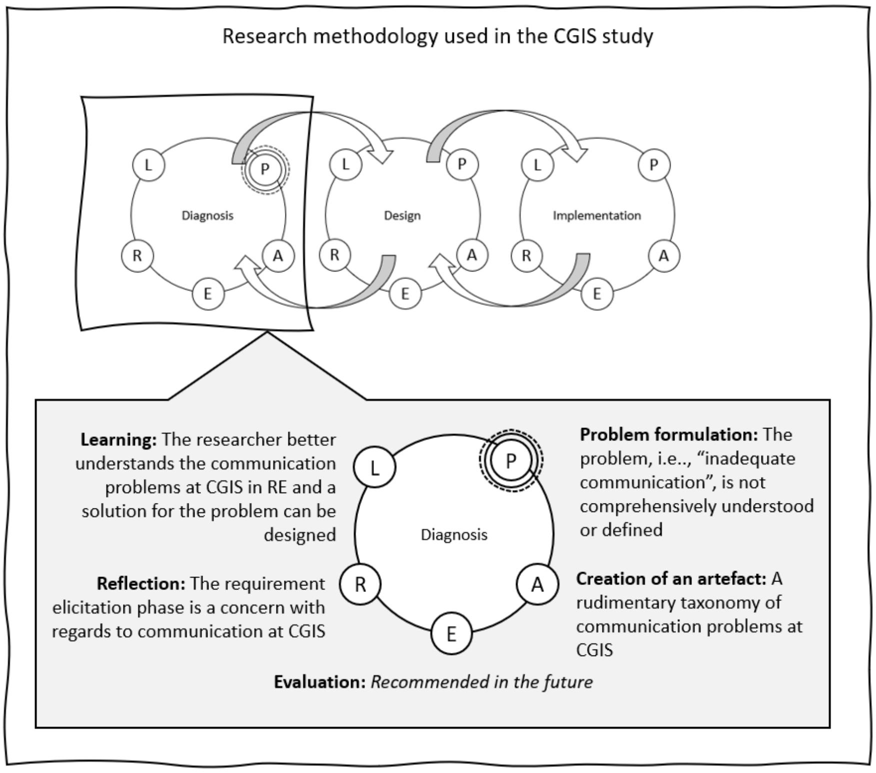

Fig.6 Summary of Diagnosis stage

Design Stage

Since the researcher was able to better understand the communication problems after this cycle of Diagnosis, the Design stage could be initiated, re-starting a cycle of *problem formulation, creation of an artefact, evaluation, reflection,* and *learning*. Based on findings from the Diagnosis stage, a new problem could be formulated, i.e., that the requirement elicitation (RE) phase was a concern regarding communication at CGIS. A more focused literature search indicated that a number of practices have already been created to facilitate communication during the RE phase of software development projects. In the 15th State of Agile Report (Digital.ai 2021), it was reported that the implementation of Agile has positively impacted visibility, business/IT alignment, delivery speed/time to market, and managing distributed teams. The Agile framework could positively impact projects and stakeholder communication at CGIS, and our aim was therefore to investigate an Agile framework and/or Agile RE practices as a suggested ensemble of solution *artefacts* to address the specific communication challenges identified at CGIS.

Scrum (Scrum.org n.d.-b) was selected as the Agile framework, to structure the practices that formed part of ARES, as it is the most used Agile framework according to the 15th state of Agile report (Digital.ai 2021), and according to Measey (2015) it is simple to understand. Scrum was therefore ideal for CGIS, not using any Agile framework before. The Requirements Specification for Developer (RSD) approach replaced the default Agile practice of user stories, as research conducted by Medeiros et al. (2020) indicated that the RSD approach integrates functional and technical requirements in a single view, therefore providing sufficient information required for coding, addressing the main deficient of user stories. Practices that were implemented in addition to the Scrum framework and the RSD approach included product planning, a product roadmap, and face-to-face virtual meetings in which attendees had their cameras switched on and could see their team members' faces.

It was important for the researcher to include the practitioners in the design of the solution, as the project team had more experience on software development projects and the researcher expected the process of implementation to be more efficient if the practitioners felt that they had contributed to the solution. The initial Agile RE solution, called ARES, was proposed, and discussed (i.e., evaluated) with CGIS practitioners by conducting in-depth individual interviews with the project team that would be participating in the implementation of the solution. Four CGIS practitioners were interviewed and were able to make suggestions and propose changes to the ARES presented by the researcher.

According to the four types of prescriptive knowledge identified by March and Smith (1995), the ARES is a *method* as it consists of an adapted Agile framework, i.e., Scrum, and RE methods that provide instructions and guidelines on how to perform communication tasks during software development projects. The ARES created in the Design stage and initially proposed to the CGIS practitioners is excluded from this article, as it was presented as a Microsoft PowerPoint presentation. The final ARES which incorporated the feedback obtained during the Design stage, and was implemented at CGIS, is included in the following section in Fig.8. The *problem formulation, creation of an artefact, evaluation, reflection,* and *learning* for the Design stage is summarised in Fig.7

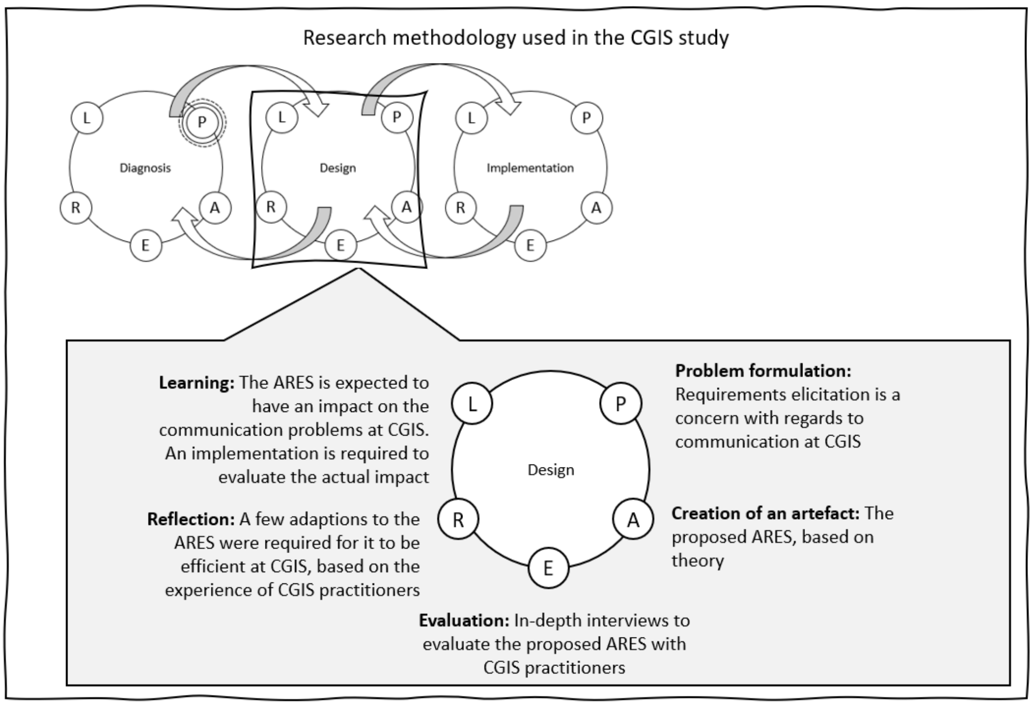

Fig.7 Summary of Design stage

The Implementation stage is included in the following section.

Implementation Stage

The Implementation stage started with *problem formulation*, aligned with the learning that was formulated as part of the Design stage, namely that a real-world implementation of the ARES was required to determine if the ARES addressed the communication problems at CGIS. Initial workshops with participants were used to further refine the suggested and theoretical version of ARES during the *artefact creation* phase. According to the four types of prescriptive knowledge identified by March and Smith (1995), the new version of ARES could still be classified as a *method.* The suggested changes to the initial version of ARES were included, and the ARES that evolved during the Implementation stage provided instructions and guidelines on how to perform communication tasks during software development projects, based on the *collaborative* design between the main researcher and the practitioners.

The graphical representation of the ARES, including the process and Agile practices that were implemented at CGIS after refinements were made, is included in Fig.8. The shading in Fig.8 aims to distinguish the Scrum components from the other Agile practices incorporated into the ARES.

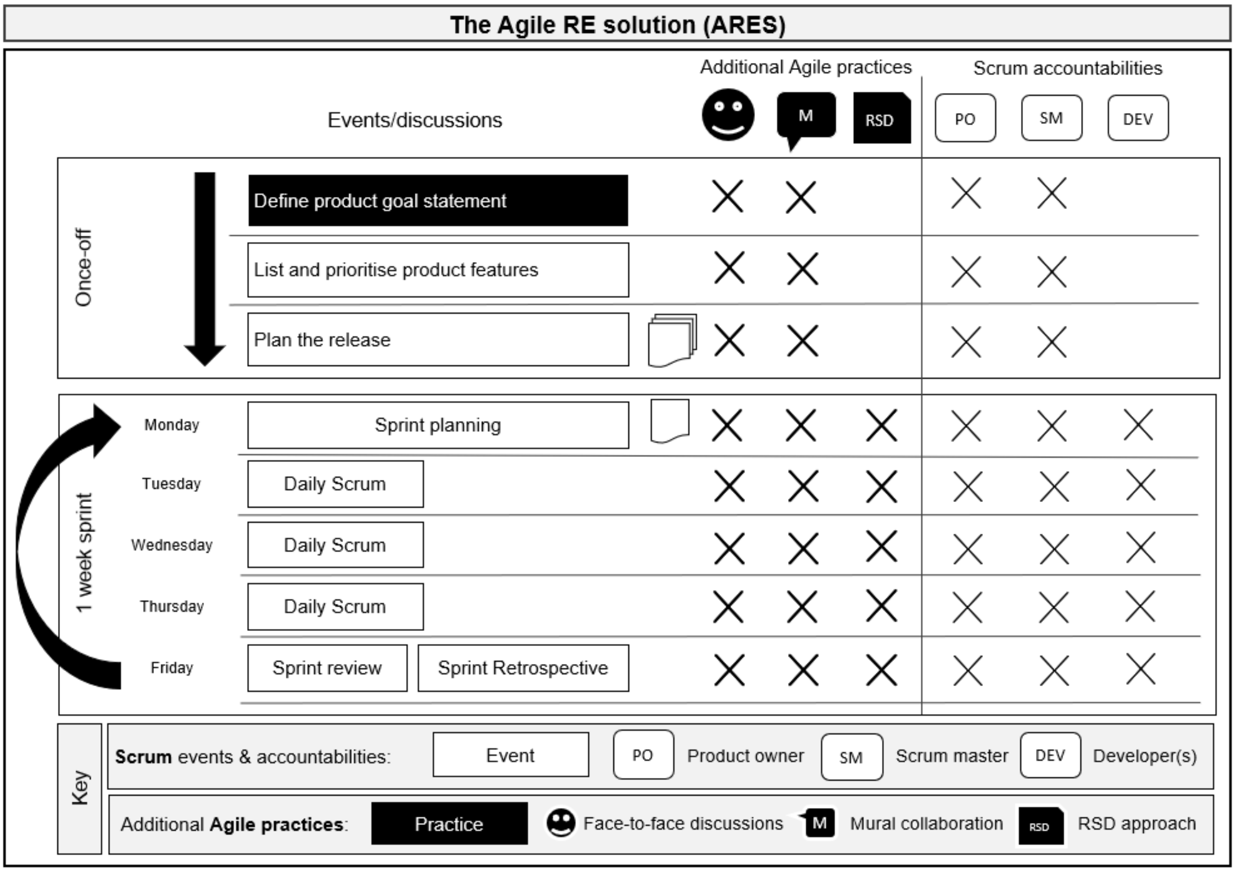

Fig.8 The Agile RE solution (ARES)

After the ARES had been implemented for two one-week sprints, i.e., a total of two weeks of applying ARES, interviews were conducted with four CGIS practitioners to determine whether the artefact in the Implementation stage had addressed communication problems as expected, and to identify improvement opportunities. The artefact was therefore *evaluated*, and the main researcher determined that the ARES addressed certain communication challenges at CGIS but that the problems that occurred *most frequently* were not necessarily addressed. In addition to this, the researcher *reflected* on the effectiveness of the ARES and determined that the ARES increased effective communication between stakeholders during RE at CGIS, but it was difficult to determine the impact on project delivery. During the *evaluation* phase of the ARES, the team also identified deficiencies of the implemented solution which included concerns regarding the documentation, testing, and scalability of the ARES as a solution. From the identified deficiencies a new *learning* emerged, namely that additional evolution of the ARES artefact is needed, as well as additional implementations of the artefact to further validate the findings.

The *problem formulation, creation of an artefact, evaluation, reflection,* and *learning* for the Implementation stage is summarised in Fig.9 .

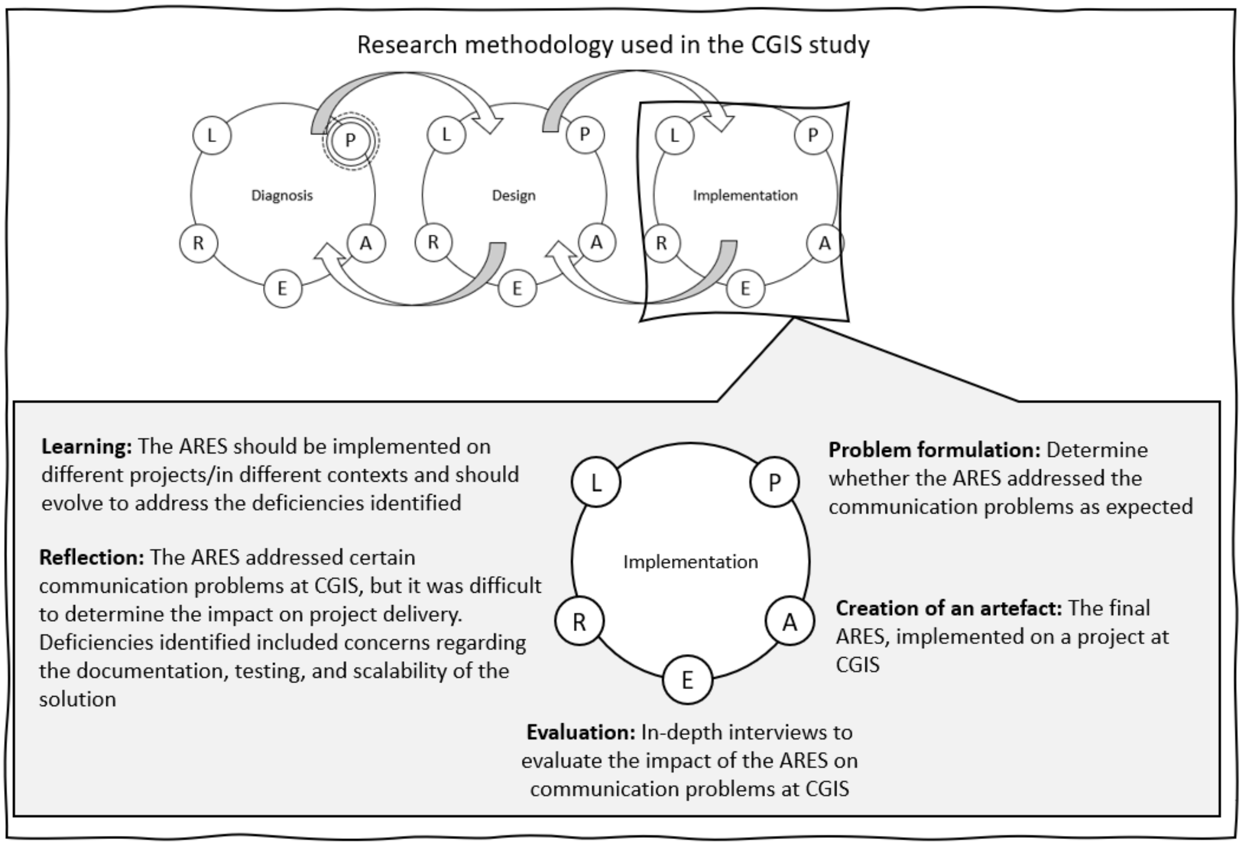

Fig.9 Summary of Implementation stage

In this study, we gradually assembled multiple artefacts by using three of the four stages of the eADR process on a real-world project at CGIS. In the following section, insights are provided regarding potential elaborations and adaptions of the eADR process, based on our experiences.

Reflections

This study covered three of the four main eADR stages to develop and implement a main artefact, i.e., an Agile RE solution (ARES). The Diagnosis stage was the initial point of entry and the researchers, in collaboration with CGIS participants, identified a problem with how software development projects were executed at CGIS. Considering that our study used the Diagnosis stage as an entry point, we cannot provide additional insights regarding the value of other entry points in the eADR.

During the Diagnosis stage, an initial *problem formulation* required multiple iterations of problem analysis. We identified a different kind of problem in one of these iterations, namely that a taxonomy was needed to structure possible causes for inadequate communication. We found that the *possibility* of conducting multiple problem analysis iterations is currently under-represented in the eADR.

The eADR was useful to highlight that multiple artefacts may be created and evaluated during different stages of the eADR. During the Diagnosis stage, a rudimentary taxonomy was created as an artefact to focus our literature search for an appropriate solution artefact, which indicated that an Agile-adapted solution could possibly address the specific communication problems at CGIS. Once the problem at CGIS was comprehensively diagnosed, in collaboration with participants, the Design stage was initiated to create the solution artefact, the ARES, that was expected to address the specific communication problems experienced at CGIS.

Both the Diagnosis stage and the Design stage included five phases, namely *problem-identification, artefact-creation, evaluation, reflection,* and *learning*. Mullarkey and Hevner (2019) refer to the first phase as the "Problem Formulation and Planning activity". Even though *problem formulation* includes *planning*, our experience indicated the need for an additional phase that reflects decision-making, rather than planning. The DSR phases presented by Vaishnavi & Kuechler (2004/5), included artefact *suggestion* as an explicit phase, which we believe is currently under-represented in the eADR. For our study, both the Diagnosis stage and the Design stage, required additional decision-making to close the gap between the *problem formulation* and *artefact creation* phases. During the artefact *suggestion* phase, the main researcher, in collaboration with the participants, need to draw on their knowledge and experience about the real-world problem context to consider multiple possible solution artefacts prior to suggesting an artefact that would best address the formulated problem. In the Diagnosis stage, for example, the vague definition for inadequate communication was *formulated as a problem*, and an artefact *suggestion* was made by the main researcher, i.e., to create a taxonomy that would structure the different causes of inadequate communication. During the Design stage, the taxonomy artefact was useful to provide a more focused *problem formulation* regarding communication problems during the RE phase of software development. Again, artefact *suggestion* was performed by the main researcher in collaboration with research participants, when an artefact was *suggested*, i.e., an Agile framework to address the communication problems associated with the RE phase of software development. A more detailed artefact *suggestion*, i.e. ARES, included an Agile framework and Agile practices, based on evidence presented in the 15^{th} State of Agile Report (Digital.ai 2021), indicating that the implementation of Agile has positively impacted team alignment. An additional motivation for selecting an Agile framework emerged from the research participants, i.e., that CGIS had already started to incorporate Agile practices at the enterprise.

Another key finding during the study was that the main researcher had difficulty in differentiating between the *reflection* and *learning* phases that are embedded in each stage of the eADR process. In the process, proposed by Sein et al (2011), *reflection and learning* are actions that are grouped together in a single phase. The separate phases in the eADR introduce unnecessary complexity. We suggest that the eADR is adapted by combining the two separate phases (*reflection* and *learning*) into a single phase, named *reflection and learning*. We also found that the *problem formulation* for each stage was defined by rewording the previous stage's *reflection* and *learning* and that these phases were closely related in our study.

Based on the reflections discussed in this section, an adapted eADR process is proposed in Fig.10. Regardless of the entry point, the *problem formulation* design cycle phase is a requirement per stage and may require multiple iterations. Arrows have been added around this phase to visually depict those multiple iterations. For our CGIS-study we required three iterations of problem investigation during the Diagnosis stage. Thus, we cannot claim that multiple iterations would also apply to the other stages.

With regards to the *design cycle phases*, it is recommended that an additional design cycle phase, *suggestion*, should be added between *problem formulation* and *artefact creation*. The *reflection* and *learning* should also be grouped into one design cycle phase called *reflection and learning*, since the distinction between the two separate phases is vague.

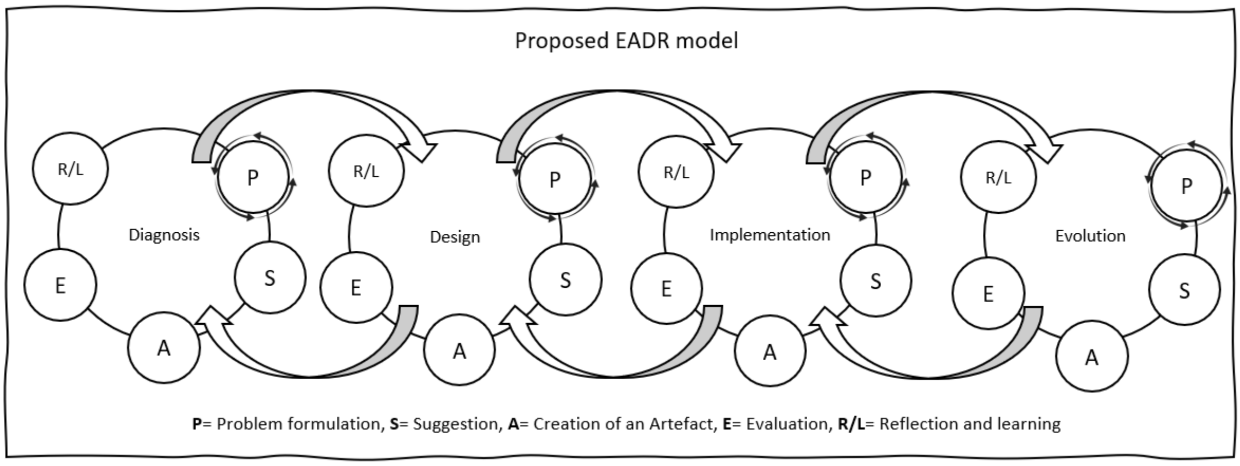

Fig.10 Proposed changes to the eADR model

Comparing the eADR that was suggested by Mullarkey and Hevner (2019) with the implementation of eADR when Tarpey and Mullarkey (2019) re-designed a Labour Planning and Allocation Cycle (LPAC) system (refer to Fig.3), the implemented version of the eADR deviated substantially from the suggested eADR process, both in terms of the stages as well as the phases, as indicated in Figure 1 of Tarpey and Mullarkey (2019). For the LPAC study, Tarpey and Mullarkey (2019) used four stages, namely Problem Diagnosing, Conceptual Design, Build, and Implementation. Within each of the stages, they incorporated only three phases in their eADR method application, namely *intervene*, *evaluate* and *reflect*. Hence, we believe that the initial eADR process that was suggested by Mullarkey and Hevner (2019) highlight the fact that multiple stages may exist for an ADR study, where each stage includes multiple phases. Our experience confirmed the need to adapt the eADR within the context of a particular study, especially in terms of phases. In addition, our study provided a real-world demonstration of the eADR process, extracting valuable insights to further enhance the eADR model. Since the eADR process is also an artefact that needs to evolve in future, the next section provides recommendations for future work and further evolution of the eADR process.

Conclusions and Future Research

ADR is a relatively new research methodology that developed from two existing research paradigms, namely AR and DSR. Although Collatto et al.'s (2018) study indicated that only 13% of sampled studies applied ADR as a new method, we agree with Mullarkey and Hevner (2019) that the initial work of Sein et al. (2011) conceal the multiple contributions that emerge from an ADR-based study.. Using an alternative version of ADR, called the eADR process, suggested by Mullarkey and Hevner (2019), this study provided a demonstration of the eADR to further validate its usefulness. In this section we conclude on the *eADR's usefulness*, indicating that the eADR elucidated the multiple artefact-contributions that emerged from our study during different stages of our research. We provided a *practical demonstration of eADR* to develop a *method-artefact* as the main artefact, which differs from Mullarkey and Hevner's (2015) demonstration context where they developed an information system as an artefact, whereas Tarpey and Mullarkey (2019) developed a new system as artefact, i.e. "new work and organisational structures/systems". In this experience report, our eADR's demonstration further evaluated the eADR as an artefact and whether its existing structure of stages and phases reflected our real-world participative interactions during our research. Based on our reflections, we suggested changes to the eADR process, indicating the need to further *evolve the eADR*. In the subsequent paragraphs we highlight the main reflections from our study.

eADR usefulness. The eADR was useful in elucidating multiple artefact-contributions during the course of our study. The rigour of evaluation of the two artefacts that were created during our study, differed. The *taxonomy*, developed during the Diagnosis stage, still requires evaluation in practice, whereas the *ARES* was designed and implemented within a real-world context. For future work, the *taxonomy* should be validated at other enterprises that also experience communication problems during software development projects. One of the main categories of the *taxonomy*, the context of communication, should be further developed, based on other prominent clusters of communication-related problems. As an example, our research at CGIS indicated that communication problems are more prominent within the requirements elicitation phase of software development projects. The main artefact, *ARES*, was designed and implemented at CGIS and is still in use after implementation. It is recommended that the Scrum team and relevant stakeholders should be interviewed again, since the evaluation results obtained after prolonged use of the *ARES*, could be different than the results originally obtained. In addition to this, it is recommended that the *ARES* should be implemented on other project(s) at CGIS. Therefore, additional evaluation iterations are suggested in the Implementation stage, as well as additional cycles of the Implementation stage.

Practical demonstration of eADR to evaluate the eADR. Our study validated three of the four stages of eADR, namely Diagnosis, Design, and Implementation. According to Mullarkey and Hevner (2019), an artefact-based methodology should also facilitate future development of the newly created artefact during additional Evolution stages, which was excluded for the purpose of this article. For future research, it is recommended that the Evolution phase is used to address deficiencies identified in the Implementation phase, e.g., adding additional Agile practices to the ARES in creating a modified ARES as an artefact, or scaling the solution by creating a scaled ARES as an artefact. During the practical demonstration, we also validated the need for multiple phases, e.g., *problem formulation*, *artefact creation*, *evaluation*, *reflection* and *learning*, within the four main stages. Reflecting on the usefulness of the five phases, we highlighted the need for an additional phase, called artefact *suggestion* and a consolidation of two phases (*reflection* and *learning*) into a single phase, namely *reflection and learning*. Our experience also indicates that multiple iterations of the *problem formulation* phase may be required, which in our case also prolonged the duration and progress of our study.

Evolving the eADR. Our study indicated that the eADR was useful to guide and structure our study, elucidating different artefacts as contributions during different stages of the eADR process. Yet, the practical demonstration also highlighted the need for further evolution of the eADR. We agree with Sein and Rossi (2019) that the eADR process highlighted the importance of a separate Diagnosis stage, but that future research and demonstrations are necessary to further validate the suggested changes that we communicated in this article.

Compliance with Ethical Standards

The CGIS practitioners (i.e., participants) were informed about the study, their right to decide whether to participate or not, i.e., that participation is voluntary, and their right to withdraw from the study at any time to ensure *autonomy*. Written consent was obtained from participants before every interaction, i.e., before every round of interviews and the focus group session. Participants in this study provided consent that opinions, feedback, and any other data gathered could be included and directly referred to in the study and research. Consent was also specifically obtained per participant for recording the interview(s)/focus group session and including the answers/responses in this study. The University of Pretoria's Faculty of Engineering, Built Environment and IT's ethical committee approved the study.

For *beneficence*, confidentiality was crucial in the study. Although data, e.g., participant details, opinions, and feedback, obtained in the study was kept for traceability, the data was anonymised in the study so that it could not be linked back to a specific person. It could however be linked back to a specific role. Anonymity of the employees participating in the research was also ensured by not referencing the enterprise directly in the study/research, i.e., CGIS was used as a pseudonym.

Statements and Declarations

Funding and Conflict of Interests

This study was partially funded by the University of Pretoria, providing some funding to the primary researcher, Anthea Venter. The secondary researcher is currently employed by the University of Pretoria, a research institution that expects researchers to perform research as part of their duties. No additional funding was received to assist with the preparation of this manuscript. The main researcher was employed at CGIS which could create some bias during data-gathering. However, the content included in this article focuses on the use of ADR as a research methodology and *not* on the data gathered for the ADR study. The authors therefor have no competing interests to declare that are relevant to the content of this article.

Data Availability

The primary researcher obtained ethical clearance from the Ethical Clearance Committee (University of Pretoria) prior to commencing with data-gathering. The datasets generated during and/or analysed during the eADR-study are available from the corresponding author on reasonable request. However, the content included in this article focuses on the use of ADR as a research methodology and *not* on the data gathered for the eADR study.

References

Bergan P, Mölders A, Rehring K Towards Designing Effective Governance Regimes for Smart City Initiatives: The Case of the City of Duisburg. In: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020. pp 2323-2332

Collatto DC, Dresch A, Lacerda DP, Bentz IG (2018) Is Action Design Research Indeed Necessary? Analysis and Synergies Between Action Research and Design Science Research. Systemic Practice and Action Research 31:239-267 doi:10.1007/s11213-017-9424-9

Coughlan P, Coghlan D (2002) Action research for operations management. Int J Oper Prod Man 22:220-240 doi:doi:10.1108/01443570210417515

Daniel C, Mullarkey M, Hevner AR (2018) Capturing User Generated Video Content in Online Social Networks. In: Designing for a Digital and Globalized World. Springer International Publishing : Cham, pp 333-347. doi:10.1007/978-3-319-91800-6_22

De Vries M, Berger S (2017) An action design research approach within enterprise engineering. Systematic Practice and Action Research 30:187-207 doi:10.1007/s11213-016-9390-7

Digital.ai (2021) 15th State of Agile Report https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494.

Dixon PDR, Sundarraj RP An e-ADR (elaborated Action Design Research) Approach Towards Game-based Learning in Cybersecurity Incident Detection and Handling. In: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

Dresch A, Lacerda DP, Attunes JAV (2015) Design science research: a method for science and technology advancement. Springer, New York

Gimpel H, Kerpedzhiev G, König F, Meindl O Teaching an old work system new tricks: Towards an integrated method for work system transformation in times of digitalization. In: Twenty-Eighth European Conference on Information Systems (ECIS2020), 2020.

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Quarterly 28:75-105

Iivari J, Venable J (2009) Action research and design science research - seemingly similar but decisively dissimilar. In: 17th European Conference on Information Systems. Verona, pp 1-13

Kutz D, Cumbie B, Mullarkey M (2022) Incorporating the student perspective in designing a virtual team classroom environment: an elaborated action design science research approach. Journal of Research in Innovative Teaching & Learning ahead-of-print doi:10.1108/JRIT-02-2022-0007

Littlejohn SW, Foss KA (2005) Theories of Human Communication. Thomas/Wadsworth, Belmot

Measey P (2015) Agile foundation: Principles, practices and frameworks. BCS, London

Medeiros J, Vasconcelos A, Silva C, Goulão M (2020) Requirements specification for developers in agile projects: Evaluation by two industrial case studies. Information and Software Technology 117 doi:10.1016/j.infsof.2019.106194

Mullarkey MT, Hevner AR (2015) Entering Action Design Research. In: DESRIST 2015, LNCS, vol 9073. Springer, Switzerland, pp 121-134. doi:DOI: 10.1007/978-3-319-18714-3_8

Mullarkey MT, Hevner AR (2019) An elaborated action design research process model. European Journal of Information Systems 28:6-20 doi:10.1080/0960085X.2018.1451811

Mullarkey MT, Hevner AR, Grandon Gill T, Dutta K, th International Conference on Design Science Research in Information S, Technology Dt (2019) Citizen Data Scientist: A Design Science Research Method for the Conduct of Data Science Projects. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11491 LNCS:191-205 doi:10.1007/978-3-030-19504-5_13

Okoli C (2015) A guide to conducting a standalone systematic literature review. Communications of the Association for Information Systems 37 doi:https://doi.org/10.17705/1CAIS.03743

Scrum.org (n.d.-a) The Scrum Guide.

Scrum.org (n.d.-b) What is SCRUM? https://www.scrum.org/resources/what-is-scrum.

Sein M, Henfredsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. MIS Q 35:37-56 doi:doi: 10.2307/23043488

Sein MK, Rossi M (2019) Elaborating ADR while drifting away from its essence: A commentary on Mullarkey and Hevner. European Journal of Information Systems 28:21-25 doi:https://doi.org/10.1080/0960085X.2018.1527189

Tarpey RJ, Mullarkey MT (2019) Extending Design Science Research through Systems Theory: A Hospital System of Systems. In: DESRIST 2019, vol LNCS 11491. pp 108-122

Vaishnavi V, Kuechler W (2004/5) Design Research in Information Systems. http://desrist.org/design-research-in-information-systems. Accessed 11 December 2009

van Wyk Q, van Biljon J, Schoeman M, th International Conference on Design Science Research in Information S, Technology Dt (2020) Knowledge Visualization for Sensemaking: Applying an Elaborated Action Design Research Process in Incident Management Systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12388 LNCS:142-153 doi:10.1007/978-3-030-64823-7_14

Venkatraman S, Sundarraj R (2022) Assessing organizational health-analytics readiness: artifacts based on elaborated action design method. Journal of Enterprise Information Management ahead-of-print doi:10.1108/JEIM-10-2020-0422

Venter A (2022) An Agile solution to address communication challenges during software development requirements engineering (Dissertation). University of Pretoria, Pretoria

Wieringa RJ (2014) Design science methodology for information systems and software engineering. Springer-Verlag, Berlin