

# Communications of the Association for Information Systems

---

Volume 40

Article 6

---

4-2017

## The Appreciative System, Learning, and Its Impact on Information Systems Design

Frank Stowell

*University of Portsmouth*, [frank.stowell@port.ac.uk](mailto:frank.stowell@port.ac.uk)

Shavindrie Cooray

*Curry College*

Follow this and additional works at: <http://aisel.aisnet.org/cais>

---

### Recommended Citation

Stowell, Frank and Cooray, Shavindrie (2017) "The Appreciative System, Learning, and Its Impact on Information Systems Design," *Communications of the Association for Information Systems*: Vol. 40 , Article 6.

DOI: 10.17705/1CAIS.04006

Available at: <http://aisel.aisnet.org/cais/vol40/iss1/6>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Communications of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).



## The Appreciative System, Learning, and Its Impact on Information Systems Design

**Frank Stowell**

Faculty of Technology  
University of Portsmouth, UK  
*frank.stowell@port.ac.uk*

**Shavindrie Cooray**

Department of Management  
Curry College, USA

---

### Abstract:

Many researchers and practitioners recognize that there is a link between the failures of information system (IS) provision and flawed Information Requirements Definition (IRD). Misinformation arises from defective communication between clients and analysts and lead to situations where the actual requirements of clients are not identified and properly represented in the subsequent information system. Recent research suggests that this could be the result of inquiry methods that do not subscribe to known learning theories and instead focus on contextual factors affecting client learning. In this paper, we explore the underpinning ideas of client-driven requirements definition and attempt to find a way of “navigating” the gap between what the client wants and what the technical expert can provide. The approach described stimulates client learning, which we suggest is a fundamental component of a successful outcome. We propose a method of requirements analysis that has shown its value in helping to overcome the communication gap between client and developer while creating a collaborative learning environment. The lessons learnt from this research may provide an interface for other technology driven development methods.

**Keywords:** Requirements Analysis, Information Systems, User-centered Development, Client-Led Design, Collaborative Learning, Appreciative System.

---

This manuscript underwent editorial review. It was received 05/17/2016 and was with the authors for 2 months for 2 revisions. Steven Alter served as Associate Editor.

## 1 Introduction

Few, if any, areas of human interest have not employed computer-processing power in some way to improve performance, but, as the employment of digital technology increases, so too has the number of reported “failures” (Mukherjee, 2008; Nelson, 2007). In this paper, we argue that failure relates to the way in which one defines an information system’s requirements. We believe that individuals place too much emphasis on eliciting data to satisfy a technological solution; whereas the focus should be on helping clients learn about the situation of interest before identifying requirements.

Researchers such as Wastell, (1999) and Kirsch and Beath, (1996) have documented the positive effects of client learning, but there still seems to be a greater concentration by scholars on the impediments to collaborative learning rather than to the process itself (Maichrzak, Beath, Lim, & Chin, 2005). Majchrzak et al. (2005, p. 654) add, “while there is little question that clients should learn, there is less clarity about how best to facilitate client learning during developer-client meetings”.

Although some scholars have espoused clients’ participating in requirements definition (Mumford & Henshall, 1979; Wood-Harper, Antill, & Avison, 1985; Stowell, 1985, 1991; Langefors, 1995), their ideas focus on getting clients to list their requirements or on the process of eliciting the clients’ requirements. We believe that, for a method to be truly client driven, it should fulfill three criteria: 1) the participants or clients should be able to learn about the wider “system” and not just their part in it, 2) they should be able to express their opinions without the inquiry method getting in the way (i.e., a method that does not have to satisfy the basic principles of its own design), and 3) the clients should be able to describe the requirements to the IT expert in a way relevant to them and to the expert.

As Majchrzak et al. (2005, p. 655) point out, “most of IS research is not grounded in extant theory and research about learning”. It seems self-evident that one cannot draw requirements from clients unless they recognize their perceptions, why they view the situation in that way, and learn to appreciate others’ views. In this respect, our approach to obtaining clients’ requirements differs from earlier efforts. We differentiate between client learning and knowledge elicitation. We focus on helping clients learn about the situation before moving on to eliciting their requirements. We have found that, as clients learn, given the right tools, they can define their needs in a way that makes sense to them and to the IT expert.

Our approach also draws on Champion, Stowell, and O’Callaghan’s (2005) argument that one should “navigate” the gap in understanding between clients and IT developers in a client-driven manner in order to action a requirements-definition process that they control. Champion et al.’s (2005) approach builds on client learning, which itself builds on a theory of learning that Vickers (1983a) suggests. Our approach is client driven and recoverable by both clients and third parties. Taken as a whole, the approach makes developing a genuine client-driven design method a practical possibility.

We are not alone in expressing such interest. For example, recent *MIS Quarterly* papers have called for IT developers to use client-driven methods for defining requirements to combat the effects of analyst-introduced misinformation (e.g., Appan & Browne, 2012). Specifically, they have called for “IS researchers to identify new approaches for stimulating client learning early in the IS design process” since “teams with more client learning achieved better IS design-phase outcomes” (Majchrzak et al., 2005). We also need research into methods for defining requirements in light of information systems’ apparent persistent level of failure (e.g. Baskerville, 2012; Sarkar & Valacich, 2010; Ashrafi & Ashrafi, 2008; Eberlein & Leite, 2007; Winter, Brown, & Checkland, 1995).

This paper proceeds as follows. In Section 2, we review the literature on failure. In Section 3, we discuss the theoretical foundations underlying our research and suggest an approach to define requirements that focuses on client learning as a first step before one elicits their requirements. In Section 4, we discuss the results of a field study that we undertook to explore the ideas. Finally, in Section 5, we discuss our findings and conclude the paper.

## 2 Background

For some years, researchers have suggested that a clear relationship between failure and inadequate requirements definition exists. Despite these suggestions, IS failures continue to rise. In this context, researchers note that failure depends on an IS’s inability to satisfy client requirements (Fowler, Horan, & Cope, 2007; Fortune & Peters, 2005; Doherty & King, 2005). DeMarco and Lister (1987) suggest failure is “not so much technological as sociological in nature”. Rosenkranz, Charaf, and Holten (2013) add that the

problem lies between clients' expressing their requirements and developers' translating them into a technical solution. Indeed, researchers have for some time recognized the disparity between user requirements and technical interpretation, which they refer to as the "horrible interface" (Morris & Travis, 2001) or "gap" (Peppard, 2001; Champion et al., 2005) between client and developer.

Recent studies have shown that current methods cannot produce an effective process for defining requirements because they do not address factors such as analyst-introduced misinformation and a lack of client learning (e.g. Hardy-Vallee, 2012; Appan & Browne, 2012; Majchrzak et al., 2005). Some evidence shows that technical experts' influence can overly influence the development process. For example, Appan and Browne (2012) demonstrate how misinformation fed by analysts can influence the facts that clients recall. Lin and Silva (2005) show how an IT project team can subtly manipulate the user group into submitting to their ideas, and Daivdson (2002) shows how the technical experts can guide "other ISD participants' interpretations".

Research into analyst-introduced misinformation calls for analysts to reduce their influence when at the requirements-development stage of information systems development (ISD). However, we then need to know how to enable clients to express their requirements and in a form that one can use as part of the IS's definition. In other words, we need to determine how to bridge that gap between the client and the technical expert. Champion et al. (2005) argue that the gap (the gap between ideas for purposeful action and a design for a serving information system) is the difference between the clients' understanding of requirements and the requirements that developers present. Such discrepancies occur since clients know their organizational needs and developers know technology, but neither is an expert in the other's domain. As such, developers usually take the lead in developing a specification based on how they interpret clients' needs, which does not always match the actual client requirements.

Champion et al. (2005) suggest that one should *navigate* the gap in communication rather than *bridge* it. The Oxford English Dictionary (Bridge, n.d.) defines bridging (verb) as "to build a bridge over" and bridge (noun) as "a structure spanning and providing passage between two points". It defines navigating (verb) as to "make or find one's way across: steer" to "plan or direct the route or course" (Navigate, n.d.). To get from one side to the other, one has to navigate across many intermediary points using multiple "navigational aids" that suit the journey. Champion and Stowell argue that one could navigate the path from client requirements in natural language to requirements in a technical format (e.g., IT specification) by providing clients with suitable tools to gradually learn their way to an initial technical specification. They attempted to navigate the gap via an approach known as client-led information systems creation (CLICS). Although clients initially drove the process, the developers were still in control and asked to provide classes for the IT specification to represent, which compromised the approach's client-driven nature.

Popular agile methods advocate eliciting few requirements and only minimal documentation. In such methods (such as XP), IT development teams deal directly with an onsite client representative who provides the requirements in several cycles. While little research has investigated the effects of reduced requirements elicitation in agile methods, some researchers claim that it leads to less trust between clients and developers, inadequately verified requirements, and a narrow view of requirements primarily influenced by the onsite client representative (Ramesh, Cao, & Baskerville, 2010)

In this paper, we draw on recent research (Appan & Browne, 2012; Majchrzak et al, 2005) and propose a way to develop requirements that adopts learning theory and focuses on the client. In our approach, client learning precedes requirements elicitation. We focus on ensuring that clients do not list their original perceptions of requirements but instead articulate actual requirements that they identify through a structured learning process. In doing so, believe that we also address the effects of analyst-introduced misinformation. By creating a process of learning, we enable clients to collectively put together their IS and produce an initial design specification. Doing so enables clients to navigate the gap between themselves and the IT developers. We stress that our approach contrasts with agile methods that require a developer to produce IT specifications and with other approaches to defining information requirements (e.g., Stowell & West, 1994; Bell & Wood-Harper, 1998, Avison, Wood-Harper, Vidgen, & Wood, 1998; Graham, 1994).

## 3 Theoretical Foundation

### 3.1 Towards a Systemic “Appreciation” of an Organization

Several theories inform our study, such as Husserl’s (2002) phenomenology, and Gadamer’s (1998) hermeneutics. In particular, Churchman’s (1971) work on inquiring systems, Checkland’s (1999) work on soft systems, and Vickers’ (1983) notion of the appreciative system also inform our study. These ideas provide the basis of interpretivist thinking about learning and how one might inquire into something. This framework of ideas includes the “system” notion, which is characterized by the notions emergence, hierarchy, communication, and control (Checkland, 1981, p. 75). We favor this definition because “it provides...a neutral vision of what a system can be taken to be and allows the observer to translate this into a description that makes sense to them” (Stowell & Welch, 2012, p. 84).

We posit that, to approach IS development from the clients’ perspective, the analyst should put to one side how the “system” presently operates and begin by defining “what” the system is. Only then can the analyst, with the clients, define what technology support the clients require. This perspective necessarily embraces both the human (social, psychological) and the technological (computing, digital communication systems), much as Borje Langefors envisaged

#### 3.1.1 Appreciation

Gaining an understanding of an organization relies on interpreting several factors including speech, the written word, and observation. Individuals interact with the world from a position of pre-understanding or prejudice, which includes assumptions implicit in the language used in a situation (Gadamer, 1998. Ludwig Wittgenstein directs one to think about language “in situ, embedded in the lives of those who speak it” (McGinn, 1997, p. 45). So, in requirements definition we, as analysts, should seek to understand the clients’ world “in situation” (verstehen) and in terms of their words and the way they view the world—free from our conceptual, political, and linguistic predisposition.

Thus, the analyst’s task is not just to find out what a client requires but to gain an “appreciation” (Vickers, 1983a) of the wider situation. The shared process of “finding out” and learning should help the clients’ to take ownership and control of the process rather than be passive accomplices, which means that the methods one uses should be transparent and comprehensible to them (some of whom might be technologically naïve) and provide an effective means of producing a design specification that the developers can use.

In this sense, Vickers’ (1983a) notion of the appreciative cycle of learning suggests a way of making sense of the world that might provide the basis of a method for defining an IS that is grounded in learning theory. Vickers suggests that the way in which we learn and make sense of the “world” differs since individuals’ appreciative settings (unique experiences, values, and biases) shape how they view a situation. The appreciative cycle begins with interests and concerns, which are subjective and influenced by individuals’ appreciative settings. When a situation of interest arises, people make what Vickers describes as “reality judgments”: that is, we select the facts that we perceive to be relevant to the current situation. Vickers describes this selection as appreciating “what is the case”.

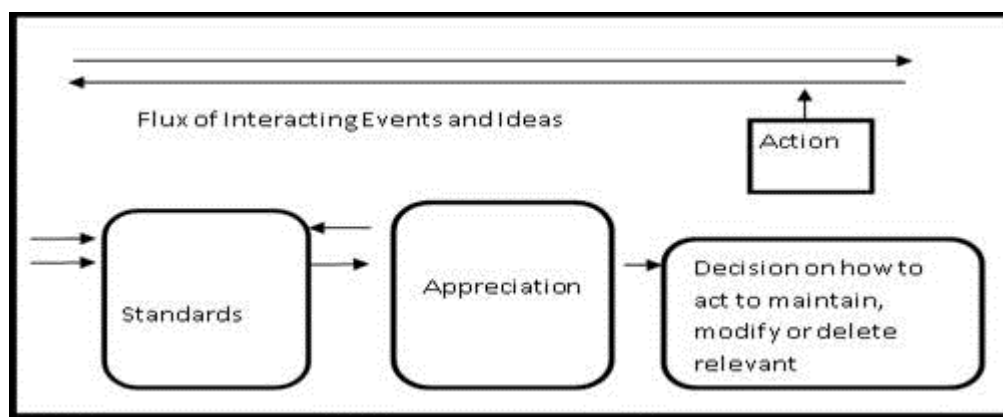


Figure 1. Vickers’ (1983) Appreciative Cycle

Next, we consider “what ought to be the case” in an ideal situation with unlimited resources and no constraints. This approach is consistent with what Vickers (1983b) calls a “value judgment” and, once again, our appreciative settings (which continuously change) influence it. From reality and value judgments come hypothetical relationships that one deems significant enough to add/maintain/modify or delete in the given situation. An “action judgment” follows in which one explores the feasibility of implementing the said relationships. The appreciative cycle’s main outcome is learning. As individuals make judgements and take action, they learn from those actions, which, in turn, cause their appreciative settings (standards, biases, values) to change. As appreciative settings of individuals change, so do their future judgements, which are based on their altered appreciate settings.

Although Vickers’ (1983a) ideas on learning and sensemaking spawned research (e.g., Armson, 2011; Ison 2005; Checkland, 1999), research has, to our knowledge, never explored the similarity between appreciation and navigation. Both notions appear to advocate a gradual, iterative and cyclic learning process based on interpretivism, although they differ at the level of consciousness of the user. For instance, appreciation is a tacit, unconscious learning process that humans go through when making judgements/decisions, while navigation is an explicit, conscious, and stepwise process. The similarity between navigation and appreciation led us to the question:

**RQ:** Can the explicit use of Vickers’ (1983a) cycle of learning provide the theoretical grounding to enable clients to learn or navigate the gap in communication and, thereby, enhance the requirements-definition process?

### 3.2 The Role of “Appreciation” in Navigating the Gap

Champion and Stowell’s (2005) argue that one should use simple systemic tools during navigation to ensure that all stakeholders participate in defining requirements. To this end, we selected rich pictures, activity models, PEARL, and relationship diagrams since they subscribe to the interpretivist paradigm on which we base our research and since they are tools that require minimal training to use. By combining these ideas, we developed an approach to navigating the gap and facilitating client learning by explicitly following Vickers’ (1983a) cycle of “appreciation”

First, we allow clients to learn about their own and others views of the situation under inquiry before they identify a problem or think about ways of addressing it. In this instance, we used a rich picture, part of soft systems methodology (SSM) (Checkland, 1999). We ask that the clients to collectively identify a “system of information” (Langefors, 1995, p. 55) that they believe will help improve their situation. The “system of information” is represented as activities in an activity model. The activity model that clients produce represents the activities or business processes they identify as those that can improve their situation (note that, typically, the activity model represents the client’s perception of the situation, while a design specification relates to the developers’ interpretation). We used this part of SSM first in order to produce activity models. Once the clients produced the activity models, we diverged from SSM because we focused on facilitating client learning by providing clients with simple systems tools for them to navigate from an activity model to a design specification by explicitly following a process of learning (i.e., Vickers’ (1983, p. 40) appreciation)).

Navigation from an activity model (AM) to a design specification requires analyzing the individual activities in the AM. To assist in this process, we chose the mnemonic PEARL (Champion & Stowell, 2001) as the tool to be used to make “reality” and “value” judgments (Vickers, 1983a, p. 40) about important activities in the activity model.

We adapted PEARL beyond that that Champion and Stowell (2001) envisage. Champion and Stowell developed PEARL as a tool to establish the authenticity of an inquiry and as a means of documenting the manner or atmosphere in which a social inquiry occurs. We sought to discover if one could use PEARL as a tool for knowledge elicitation and sensemaking as part of our approach to navigation. We chose it because it draws attention to a variety of issues, such as informal power (commodities), which other sensemaking tools ignore. Additionally, PEARL, shares the underlying notion on which this research builds; namely, interpretivism (Bullock & Trombley, 2000, p. 442; Jary & Jary, 1995, p. 336; Schutz, 2011; Checkland 1999, pp. 273-277).

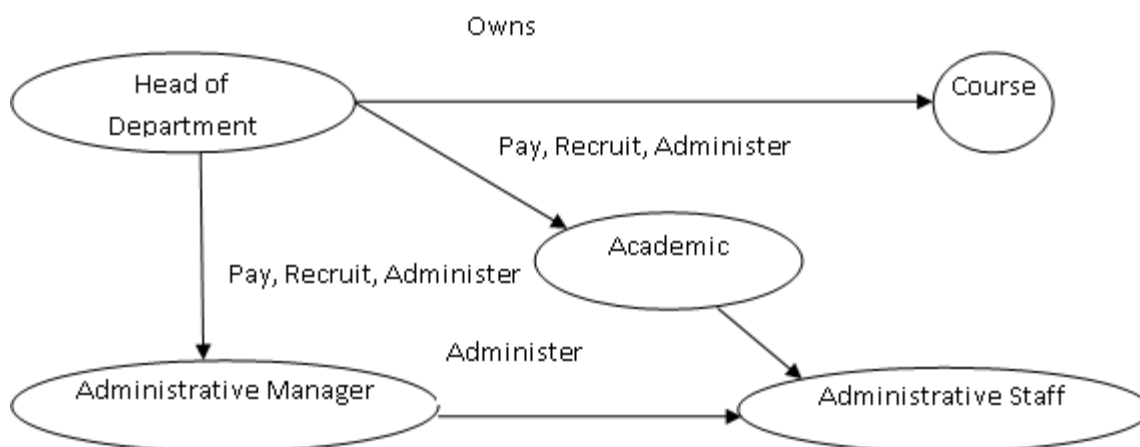
We used the questions corresponding to the five PEARL elements (See Table 1 below and Stowell & Welch, 2012) to encourage clients to reflect on “what is the case” (corresponding to Vickers’ (1983a, 1983b) reality judgments at the present time) for the activities they consider to be important in the activity model produced earlier. Thereafter, we used questions corresponding to the five elements in PEARL to

explicitly reflect on “what ought to be the case” (corresponding to Vickers value judgement in an ideal environment) for the activities the clients consider to be important in their activity model.

**Table 1. List of PEARL Questions**

Elements of PEARL	Issues to reflect on “What <i>is</i> the case” for each activity (reality judgments)	Issues to reflect on “what <i>ought</i> to be the case” for each activity (value judgments)
P: participants	Who is Involved in the activity, who is excluded, and why? Why are they involved? What is their role in the activity?	Who ought to be involved/excluded in the activity and why? What should be their role in the activity?
E: engagement	How are the participants involved? What methods are used to engage participants? What are the environmental influences in which an activity takes place?	How should participants involved? What methods should be used to engage participants?
A: authority	Formal authority associated with activity- what are the environmental influences? What embedded authority do the tools for engagement have? Why were they chosen and what influences the outcomes?	What should be the formal authority associated with activity?
r: relationships	What kind of informal power or commodities (Stowell, 2014, Stowell & Welch, 2012, pp. 116-118) do people use to influence others (examples include using gender, sociability, and verbal skills)?	What kind of informal power or commodities do people use to influence others?
L: learning	The theoretical and practical outcomes from the activity, judgements about how these were achieved, and assessment about the ownership of outcomes	What should be the theoretical and practical outcomes from the activity? Who should have the ownership of outcomes?

We use PEARL in this way to make clients’ reality and value judgments explicit, which will help them to gain a clearer understanding of what outcome they would like see occur (in an ideal situation). As clients discuss the PEARL elements, the analyst documents their verbal responses in a relationship diagram (See Figure 2). One should not confuse a relationship diagram with an entity relationship model. Rather, a relationship diagram is a simple diagram that one uses to “display connections between related components or concepts respectively and help in preliminary sorting out of those components or concepts within a particular situation” (Shipp, 2005).



**Figure 2. An Example of a Relationship Diagram**

One asks participants to reflect on the PEARL questions (see Table 1 above) first in relation to “what is the case” at present in their situation (reality judgements) and then with relation to “what ought to be the case” (value judgments) in an ideal setting. In this case, the answers that clients give to P and A (in PEARL) become “roles” in the relationship diagram and answers that clients give to E, r, and L became relationships in the relationship diagram. Figure 2 shows an example of how a relationship diagram documents the roles in a problem domain (managers, academics, administrative staff, head of departments) and the relationships between them.

The next stage is to determine the feasibility of implementing the ideas in the relationship diagram by considering the relationships, processes, and attitudes that might be influenced if the said relationships were implemented.

By carrying out an influence analysis, the clients can investigate the mismatches between “what is the case” and “what ought to be the case” that they identified earlier, which means that they can select a set of relationships they think can be practically implemented. By exploring the mismatch between “what is the case” and “what ought to be the case”, the clients can make what Vickers (1983a) refers to as an “action judgment”; that is, highlight the relationships that need to be maintained if the corresponding activity was implemented. One can then modify the relationship diagram to reflect only feasible relationships.

In the final step, one converts the feasible relationship diagram into a use-case diagram. A use-case diagram belongs to the UML standard and is a common design specification that many developers use. A use-case diagram represents the client’s view on the behavior that the software (IT) should exhibit. However, it is not overtly technical, and a layperson could produce a use-case diagram with the guidance of a more knowledgeable analyst, which is particularly true in our case since clients do not need to produce a use-case from scratch. Instead, one can use the feasible relationship diagram that clients produce as a framework to create the use-case diagram. Developers typically use use-case diagrams to produce class diagrams that are the basis for object-oriented software design (Ashrafi & Ashrafi, 2008), which leads to software code. Once produced, the use-case diagram becomes a contract between the developers and the clients in which the developer agrees to provide the means to enact the interactions (specified in the use-case diagram) between the clients and software. While the use-case diagram is the closest to a technical specification (of the models used), it is not a comprehensive software specification. The approach does not aim to provide such a detailed technical specification but to expose users’ views and document them in a way that users can understand and use as an agenda for discussions with the IT developers. Users can also use the portfolio of models as a basis for communication with developers and as a means to evaluate the technology that developers provide.

IT developers can also use the portfolio of models as a way of producing a detailed software specification. The clients and developers can use the use-case diagram to trace back the source and, hence, provide transparency and logical coherence to the process.

## 4 The Field Study

Prior to the study reported here, we tested our initial ideas on a learning framework for determining IS requirements in a field study at a university (see Cooray & Stowell, 2007). The lessons learned from that study resulted in significant changes to the approach, and we sought a setting outside of an academic environment to test the new ideas. It was at this juncture that we began talking to the chief librarian of the Central Library of the City of Portsmouth through an employee of the Portsmouth City Council who was aware of our research. The Portsmouth Central Library (which is the largest library among 13 that the Portsmouth City Council manages) became an ideal setting for this task. The library provides services such as lending CDs, DVDs, and more than 250,000 books in 30 different languages. It also provides drop-in sessions for job seekers, those researching family history, or visually impaired people wanting to use computers. The library also organizes special events for children, reading groups, and clubs. It provides some services online and has an active presence on social media.

At the time of the study, the library was going through a period of disruption and was seeing dwindling numbers of patrons possibly due to the rise of online book sellers, changing public perceptions of libraries, and café-style book shops—a phenomenon commonly seen by commercial enterprises. The control of the central library had changed from the Hampshire Council over to the Portsmouth City Council a decade prior to the study, and the technologies used had not been significantly updated since. The main technology used for the previous decade at the library was Galaxy, which is a commercial software package. Two staff members in a unit known as IT Services managed the technology at the library. While the library’s IT services could suggest technology changes, the final approval lay with the Portsmouth City Council. After a decade of minimal technology change, the city council had finally approved a request to phase out Galaxy and bring in a more current software package. Due to the cost constraints set by the city council, the library intended not to build custom-designed software from scratch but to buy a commercial software package from a vendor that IT services could customize. This build versus buy dilemma (e.g., Ledeen, n.d.; Cohn, 2014) is an age-old software question that most organizations face, and many like the



library decide on the buy option due to lower costs. While IT Services had narrowed down the vendor software options to a couple of choices, the managers also felt a sense of unease and tension in the library staff possibly due to the upcoming technology changes.

The managers and library staff felt that the software might not meet their needs and that their jobs could be at stake. The prevailing situation imposed a fairly high pressure situation in which the managers at the library had to decide on a software package that met the cost approved by the city council in the time frames it set. The selected software had to also satisfy staff's need in a manner that improved staff effectiveness and paved the way for a smooth change process. The managers generally expected to first understand the cause for tensions in the library and second to identify the information that staff required to do their tasks more effectively. They wanted to empower staff by making them feel that they had more control over the process of transition and give them a better sense of their information needs so that they could advocate for themselves better. Additionally, the identified information needs of staff would play a pivotal role in helping IT services to decide on which packaged software would best provide the required information. The managers felt that understanding staff members' information needs was more crucial in this situation since, when building software, software developers would conduct a more formal analysis and design phase. In accordance with the decision to buy vendor software, we emphasized that the study would provide a portfolio of models (including rich pictures, activity models, relationship diagrams, and use-case diagrams) that IT services could use to ascertain which candidate vendor software had feature sets that matched staff members' information needs. Picking the right software package was essential to the library since it wanted to minimize later software customization due to both cost constraints and lack of staff in IT Services. It appeared that the library wanted to avoid a problem commonly seen with purchasing vendor software packages: that is, that "too often, people select software based on factors such as price, current technology buzz or the system that is the flashiest but without a good fit. Such companies are left with expensive customization and bolted together solutions." (Schiff, 2014).

An EPSRC grant funded the research, and the library did not compensate us. The library's management was motivated to participate in the study since they were in the process of deciding on a software package in the midst of tensions among staff and did not have the finances or the IT staff to conduct a thorough analysis of staff information needs themselves. We focused on investigating the feasibility of their ideas in a complex, evolving situation that involved the selection of a library-wide commercial software package to take over from the technology that it had used for the last decade. We also explored the usefulness of the ideas in selecting a software package that would better fit users' needs and help reduce future software customization (and the associated costs). Our findings can help others alleviate the common problems of user dissatisfaction and resultant software customization that many companies that pick the buy (vs. build) software option face. We conducted the study between July 2006 and September 2007.

## 4.1 Framing the Field Research

Action research (AR) has many variations, such as canonical AR (Susman & Evered, 1978; Davison, Martinsons, & Ou, 2012), action learning (Revans, 1982) and action science (Argyris, 1995). The particular type of AR we chose for this research was that based on work at the University of Lancaster over a 30-year period (Checkland, 1999, pp. 151-154). We chose the Lancaster model since it adheres to the interpretive paradigm, advocates a cycle of learning, and provides a high level of rigor that assists external parties to see how one achieved results. As Checkland, (1995, pp. 2-3) notes, if one is to take action research seriously, rather than a set of anecdotal reflections, researchers should declare the intellectual framework of ideas in advance. In other words, it is important that the researcher sets out "the epistemology in terms of which the research findings will be expressed" (Checkland, 1995, p. 56). Researchers should also declare the method they intend to use. These steps make up the FMA model of action research in which F is the framework of ideas, M is the method used, and A is the area of interest (see Checkland & Holwell, 1998, pp. 23-26).

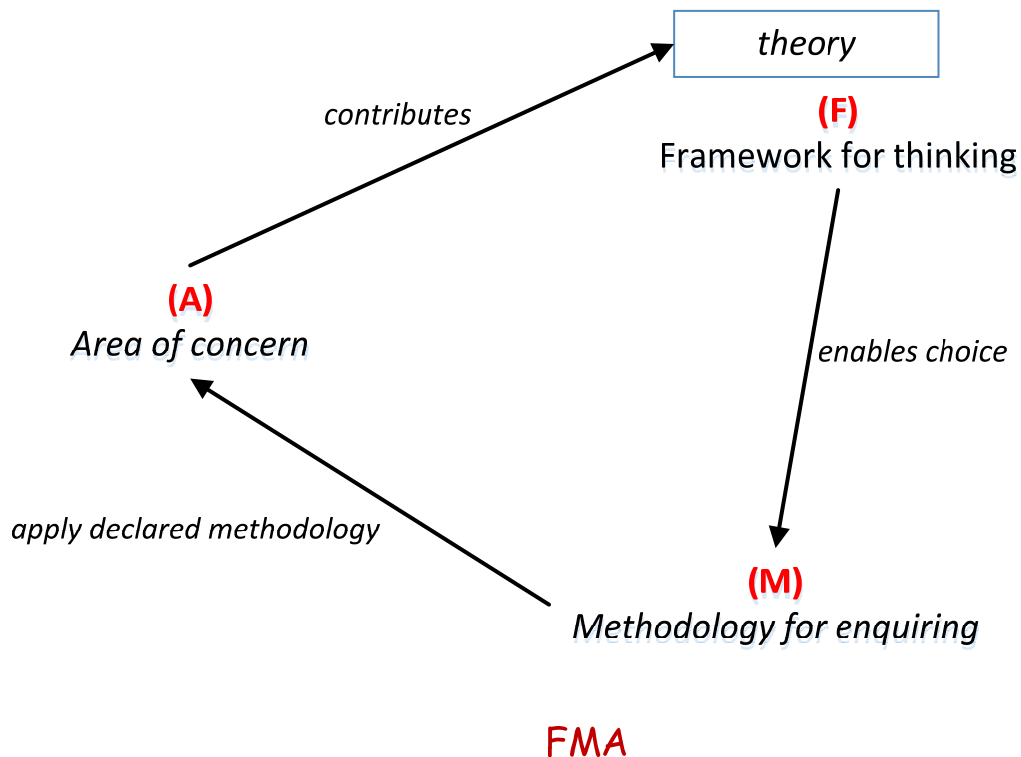
Before the start of the field study, we defined the following FMA model with the intention of revisiting the model after we completed the study to explore how the ideas evolved.

**Research theme:** we derived a research theme from literature that identified the process of requirements definition as being problematic and having implications to bodies of research relating to IS failure, misinformation, and the gap between clients and technicians.

**F:** the research's underpinning framework of ideas pays homage to Husserl's (2001) phenomenology Schutz's (1962-1966) sociology.

**M:** we were interested in developing a way of “bringing to life” the idea of navigation and client-driven requirements definition by applying Vickers’ (1983a) appreciation and the authentication tool PEARL in ways that researchers have not previously used (see Section 3.2).

**A:** we conducted the study at Portsmouth City Central Library (in the UK).



**Figure 3. FMA Model (Checkland & Scholes, 1990)**

Avison, Lau, Myers, and Nielsen (1999) point out that, in AR, “the researcher wants to try out a theory with practitioners in real situations, gain feedback from this experience, modify the theory as a result of the feedback, and try it again”. The field study discussed in this paper is part of an AR cycle of learning that began with a pilot study (see Cooray & Stowell, 2007) that tested an initial version of the theory. After reflecting on the findings of the pilot study, we modified the initial idea and tested it in the larger field study detailed in Section 4 above.

In their paper on managing risk in software process improvement Iverson, et al (2004) argue that one should articulate certain criteria to maintain relevance and rigor in a qualitative AR study. These criteria comprise:

1. Clarifying the roles that the actors involved play
2. Documenting how one collected data
3. Examining the usefulness of the implemented ideas
4. Considering the study’s theoretical contributions, and
5. Considering the conditions under which one can transfer or adapt the results to other contexts.

In this section, we consider the first two; in Section 5, we consider the remaining three.

**Roles:** although AR researchers cannot be objective observers, clarifying their roles can help establish their impartiality (Checkland, 1995). In this study, we documented our own interactions in each session (see Tables 3 and 4). We used the subsequent documents to help reflect on our roles, learn from them, and, if necessary, change our approach. Furthermore, participants “authenticated” and edited all models and documentation produced. The practitioner-participants in the study were subject librarians,

supervisors, library assistants, an IT supervisor, the chief librarian, the chief library manager, and the assistant library manager.

**Documentation:** Baskerville and Wood-Harper (1996) suggests that describing how one collected data in detail distinguishes research from consulting. To enable a third party to recover the process of inquiry, we provide appropriate records as follows: after each session with participants, we created two types of records. The first was a PEARL record to document our interpretation of the manner or atmosphere in which the session occurred (i.e., interactions between participants) so that an external party could understand how we reached the results even though they may not be able to reproduce the exact same results (Stowell & Welch, 2012) (See Table 2 below for an example). We created this type of record to mitigate the often-used criticism that qualitative social research cannot replicate social situations due to individuals' dynamic and unpredictable nature.

**Table 2. Example of a PEARL Record Describing the Manner/Atmosphere in which the Session Occurred**

PEARL element	Description for session 15 with subject librarians
P: participants	MM, PG, SC2, AH, Shavi Cooray (SC1)
E: engagement	AH engaged in the session confidently while SC2, PG, and MM exhibited hesitance.
A: authority	Librarians
r: informal relationships	AH appeared to want to portray herself as knowledgeable in the area, which enabled her to exert informal power over the others by using her dominant personality and vocal skills. For example, SC2 was not given the opportunity to talk much, although it appeared that he wanted to but could not do so due to AH. Also PG and SC2 seemed not to appear confrontational so rarely confronted AH. SC1's age and relative lack of experience may have contributed to AH's attempt to dominate the session.
L: learning	Learned about the difficulty of getting participants to see that they should first learn about their situation in a conceptual sense without purely starting from a technical perspective.

The second type of record (e.g., "analyst's role" column in Tables 3 and 4) was a personal record of our reflections on our own interactions during each session. We used this type of record to assist both an external party and ourselves to trace how we may have influenced the sessions. One can use these records to improve on the way in which one conducts future sessions. One can use both types of records to reflect on how one conducted each session, and external parties can use them to interpret and authenticate the study's results.

## 4.2 Testing the Ideas in a Field Study

We have developed the participative approach to IS development over several projects (e.g., Stowell & West, 1994; Champion & Stowell, 2001; Stowell & Cooray, 2016a), and the research we report here is one such project that has added some useful lessons for information system development.

We conducted the field study in three stages. In phase one (appreciate) and phase two (articulate), we drew on previous research in soft systems to produce an activity model that encapsulated "ideas for action". However, it is phase three (actuate) that sets our approach apart from previous research. In this phase, we introduced a step-by-step approach, grounded in a theory of learning and operationalized through the mnemonic PEARL, to enable clients to learn or navigate their way to a design specification (e.g., use-case diagram)). Although other approaches have used a multitude of tools to translate an activity model to an IT model (e.g., Stowell & West, 1994; Liang, West, & Stowell, 1998), they typically begin with a "subjective approach" that positivism later engulfs. This shift often arises due to IT developers' early intervention, lack of traceability, and use of non-interpretive tools. Our approach provides clients with a way of following the steps in a cycle of learning in a traceable manner using tools that result in a client design specification. In order to navigate from activity models to use-case diagrams, we put Vickers' (1983a) cycle of appreciation at the center of the process and selected tools that helped participants make reality, value, and action judgments that lead to learning

### 4.2.1 Phase One: Appreciate: Starting the Learning Cycle

The first stage of the study began with meeting individual participants and inviting them to reflect on the situation and their role in it. We encouraged each participant to think about how individual elements of the situation relate to each other and their influence on the wider environment. They could use a model of their choice (e.g., verbal description, diagrams, software) to describe the area of interest. We documented this in the form of minutes. We then invited participants to represent their thoughts, in this instance, as individual rich pictures (RP) (Checkland & Poulter, 2006, pp. 24-27). Aside from showing a general example of a RP, we provided no specific guidelines about format. Each participant drew their RP without models to influence them. On completion, we asked each individual to clarify elements in their pictures. We chose the RP as a tool in this stage since the situation under consideration was fuzzy and participants had not yet identified a specific problem theme to pursue. In instances where participants are more aware of a specific problem issue or research question, we suggest that the appreciative inquiry method (AIM) (Stowell & Welch, 2012) might better suit phase one.

Once the RPs were produced, we used questions corresponding to the elements in CATWOE from SSM (Checkland, 1995) and PEARL (Champion et al, 2005) to encourage participants to explain their thinking. Participants stated that this activity helped them to take a step back and truly “appreciate” their individual views and why they viewed the situation in that way. They stated that, in their busy work lives, they rarely took the time to reflect on and question their perceptions of the situation of interest and that instead they took their views as a given.

The participants’ responses to the activities of phase one varied slightly; Cooray (2010) records the notes of each individual session. As an example, we show the response of one particular participant (participant X) made during her session in Table 3.

Using rich pictures enabled each participant to focus on “what” they currently thought about the situation rather than “how” to address it. Importantly, in this phase, instead of the enquirer’s assuming that each participant already knows what “the problem” is, the enquirer invites them to reflect on their views of the situation, which means that each person learns about the situation as a whole before the group jointly decide what the problems are. In this stage, participants did not interact with each other. Each participant summarized their key findings at the end of the first stage and stated what actions they would take in relation to those findings before the next stage. Some examples of actions that participants decided on were as follows.

- Find out about official city council documents on governance of the city libraries and how much control the city council has
- Find out more information about technology to promote the sharing of information between the city council, senior managers, and librarians
- Conduct discussions on data transparency with senior managers, and
- Find out about the data sources the information librarians need to get information from and their cost.

Participants agreed to meet again for stage two after they finished stage one. The project was client driven, and the time the clients spent at each stage depended on their “acceptance” of the outcome before moving on to the next. For this reason, it is difficult to be precise about how long each stage should take or might take if it had been a traditional project with timescales set by the problem’s “owner”. In this case, the participants determined the time for each stage, which we found acceptable as the main point was to initiate a cycle of learning. However, as a means of gaining an approximate time for each stage, we estimate that on average each participant spent two hours during this stage. We note that, to be truly client driven, a study undertaken elsewhere might be shorter or longer depending on the participants. The key role of the researcher/facilitator here is to encourage progress but, importantly, without influencing outcomes.

**Table 3. An Example of a Session with a Single Participant**

Key activities of phase one	How participant X responded to activity	Analyst's role in activity	Additional remarks
Reflect and talk about the situation of interest using any desired method of description.	Participant X used the English language, diagrams, software and hard documents to describe her opinion confidently.	The researcher did not talk unless questions were asked by participants.	There was one interruption during the session by another staff member who invited herself to the session briefly.
For the researcher to use CATWOE (Checkland, 1981) and PEARL (Champion & Stowell, 2001) to ensure that sufficient information is gathered for the rich picture from participants.	The CATWOE and PEARL questions posed by the researcher in relation to the lending service were answered confidently and comfortably by participant X.	Once X had completed the previous activity <i>uninterrupted</i> , then the researcher used the elements in CATWOE and PEARL to see which aspects about the domain had not been spoken about by X. Questions corresponding to those missing elements were then posed to X.	The elements in CATWOE and PEARL were used <i>not to</i> structure the participant's descriptions but to help the researcher as a facilitator to gather a rich supply of information about the situation of interest.
For the participant to draw an individual rich picture of her opinion of the library functions.	Participants time constraints, shortage of staff, and lack of individual desks. The staff wanted the researcher to draw rich pictures.	The researcher drew the rich picture for participant X based on her perception of what X had said. The rich picture was then shown to X and she was invited to make any modifications if necessary.	The notes made by the researcher in the previous activity about the lending service with regards to the elements in CATWOE and PEARL) helped in understanding the wider situation within which the lending staff were located. It was useful in formulating the rich picture. The activity helped her to realize that she had a sense of the unease that could be traced back to the feeling that senior managers distrust librarians and the city council has too much influence on the everyday tasks at the library. She stated that prior to the next meeting she would find out about the official financial information concerning the library.

#### 4.2.2 Phase Two: Articulate: Defining Needs

After phase one finished, we combined the individual rich pictures into a single one. We took care to ensure that we included only the elements contained in individual rich pictures in the combined rich picture. We then presented the RP to two sets of participant groups to authenticate it (see Champion & Stowell, 2001) and to encourage group discussions. Participants used the picture to discuss different elements, ask questions to clarify issues, and correct or expand on them as necessary (see Stowell & Welch, 2012, Appendix C). The participants also used the opportunity to talk about the actions they had taken after stage one and their results. For instance, a participant stated that, after careful scrutiny of official city council records, she realized that “the city council had complete power over the running of the library while the people actually working there had very little”. Another librarian who had attempted to get access to certain financial information found that she did not have clearance to view that information.

While we conducted the first stage with individuals, the second stage allowed participants to recognize the myriad views that others held. This process resonates with Gadamer's (1998) idea of a “fusion of horizons”, where “experts” exchange views on a given topic and, in the process, each learn more about the others' views. One participant exemplified the argument that Stewart (2007) and Checkland (1995) make that RPs enable a holistic view of a situation to emerge by commenting that the RP enabled her to “see the whole picture and how other people were involved” instead of just her own role in it (Stowell & Welch, 2012, Appendix C). The discussions also helped reinforce certain views. For example, librarians realized that many of them had similar discomfort about the lack of data transparency. The senior

managers were, in turn, shocked about librarians' views since they believed that they were being transparent with the librarians. All participants agreed that the combined rich picture provided them with an opportunity to view the situation in a different way and exposed them to views of others.

Table 4 summarizes the response of the first participant group to the activities of phase two.

**Table 4. Response of First Participant Group to the Key Activities of Phase Two**

Key activities of phase two	How the participant group responded to activity	Analyst's role in activity	Additional remarks
For the participants to reflect on and discuss the composite rich picture.	The participants immediately focused on the idea that there was a lack of staff communication in the library.	Benign. Allowed the participants to absorb the contents of the rich picture. Only spoke when questioned.	The participants asked questions in order to clarify the contents of the rich picture but did not appear to challenge it. As a result, there was an atmosphere of reasonableness and fairness.

Once the participants agreed on the combined RP, we asked them to identify what they considered to be problem themes and decide which one they wished to develop further. This activity generated further discussion until they reached an agreement they were satisfied with. The participants identified the following problems:

1. Gaps in communication between ground staff and management
2. Gaps in communication between staff/management and the public
3. Information not transparent to all levels of staff
4. Restrictions on the development of the library by the city council
5. Staff perception about communication differs from management's perceptions, and
6. Individual staff members have insufficient information to fully satisfy their responsibilities.

We then asked participants to decide what they considered to be the most important problem theme that they wished to explore. They decided on: "Individual members of staff have insufficient information to fully satisfy their responsibilities".

Acting as facilitator, we then produced, with the participants' assistance, a "careful concise description of the purposeful activity" (Checkland & Holwell, 1998, p. 157). This description generated further discussion as it began to form. The process of writing the description provided another opportunity for the participants to explore what each person understood the problem theme to mean. After further discussion, the end result was: "A City Council owned system to provide members of the library's lending staff with relevant information to maintain the relationships needed to effectively fulfil their responsibilities to the public and suppliers".

We then used the above definition to produce an activity model (AM). The AM, jointly produced by participants with minimum guidance, depicted the activities the participants felt were necessary to implement the system represented in their definition. The participants used the AM as a sensemaking device to discuss the activities it contained.

The size of the AM and the way certain clusters had developed prompted participants to divide the activities in the AM into 4 subsystems in the whole to ease the discussion in the next phase. The four subsystems included:

1. A system to decide how to provide relevant information
2. A system to decide how to effectively fulfill responsibilities
3. A system to decide how to maintain relationships, and
4. A system to take action.

Figure 4 represents one subsystem they selected for further analysis in the final phase.

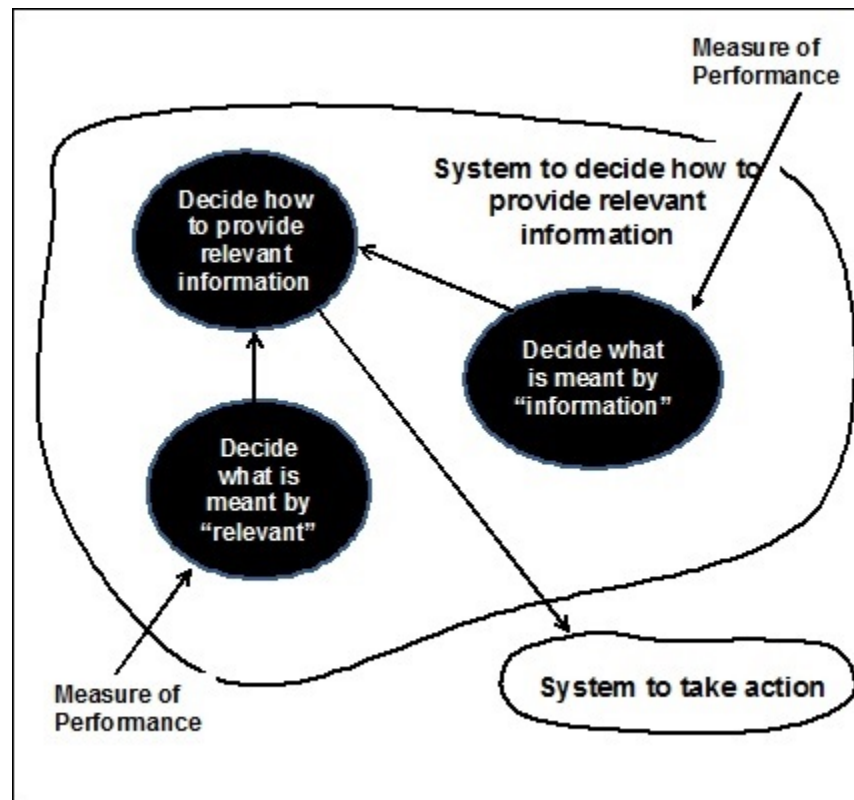


Figure 4. Extract from Activity Model

At the end of phase two, participants decided on a list of actions that they would take prior to the third phase:

1. Conduct a survey of customers to identify their concerns and expectations
2. Conduct a survey of suppliers (publishers and so on) to identify their current trends and best practices, and
3. Talk to colleagues in other libraries to identify best practices.

Participants agreed to meet for phase three once they had conducted the above tasks.

Each group session took on average three hours to complete, and, while we did not ask participants to document time spent on the above actions (after stage two and prior to stage three), we estimate that, on average, each participant spent two hours performing them.

#### 4.2.3 Phase Three: Actuate: Explicitly Using Vickers' (1983a, 1983b) Notion of Appreciation

We call this phase "actuate" by which we mean to stimulate, spur on, motivate. This phase is based on Vickers' (1983a) appreciative cycle. We have found that, by using this phase as a framework for thinking helps the participants learn their way through the feasibility of implementing the purposeful activities in the AM and translate what they learned into an initial IT specification (e.g., use-case diagram). Producing the IT specification arises out of the process of navigating the gap between clients and developers and by allowing the client to be in control of the requirements development process.

The phase began with our separately presenting the AM being to the two participant groups using Vickers' (1983a, pp. 39-41) appreciative cycle as a framework to guide the process. We invited participants to make reality, value, and action judgements in relation to the activities in the AM. We first invited them to reflect on "what is the case" (reality judgments; Vickers, 1983, p. 40) at the present time for each of the activities they considered to be important in the AM.

We used questions corresponding to each element in PEARL and CATWOE (Checkland & Poulter, 2006) to encourage participants to reflect on and discuss key issues relating to "what is the case" (reality judgements) for the important activities in the AM. Participants focused on what they considered to be

relevant information. The meeting was initially conflictual in nature since the managers' perception of relevant information and the librarians' views on relevant information differed. The librarians claimed that they did not have access to relevant financial data and that managers were actively withholding information from them. Based on the surveys and discussions that participants had with customers, suppliers, and colleagues from other libraries (prior to stage three), the librarians agreed that they now had a better understanding of the information that they felt the managers had withheld from them.

To identify a path towards a resolution, we then asked participants to reflect on the elements in PEARL in terms of "what ought to be the case" (value judgments; Vickers, 1983a, p. 40) for the same activities in an ideal setting of unlimited resources and no constraints. Again, debate on what should be considered as relevant information dominated the debate. The actions that participants undertook at the end of the previous phase assisted them in formulating a list of relevant information that they needed to perform their jobs effectively. A sample list included the following information:

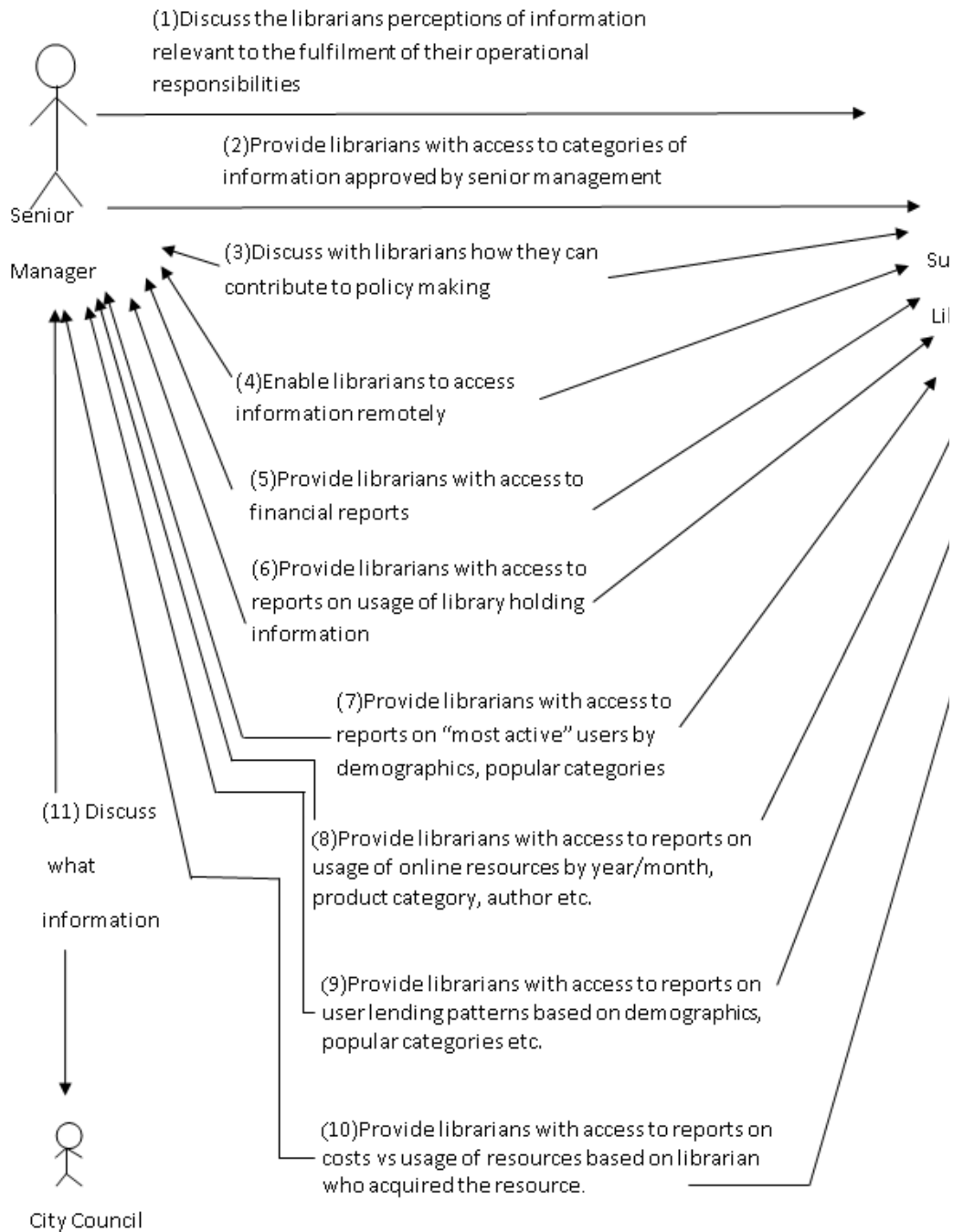
- a) Reports on financial data about costs/revenue by year/month and product category.
- b) Reports on usage of library holding information by year/month, product category, author, publisher, etc.
- c) Reports on "most active" users by demographics, popular categories etc.
- d) Reports on usage of online resources by year/month, product category, author, etc.
- e) Reports on user lending patterns based on demographics, popular categories, etc.
- f) Reports on costs vs. usage of resources based on librarian who acquired the resource.

Participants agreed that the lack of relevant information mainly affected their ability to measure their progress towards their goals. They stated that they could not evaluate the effectiveness of their work efforts if they could not measure the impact of those efforts on customer retention and expansion, costs vs. benefits, and so on. Librarians suspected that they did not have access to such information since the senior management preferred to keep that information to themselves. Senior management explained that they could only supply information that the city council approved.

The value of using PEARL as a sensemaking device became clear as the participants quickly grasped the difference between "what is the case" and "what ought to be the case". Participants then used the elements of PEARL to learn about what they understood by each activity. Although PEARL engendered a discussion about such things as informal power relations and "commodities", which other knowledge-elicitation tools ignore, it took some participants time before they felt comfortable enough to freely discuss element "r" in relation to their situation (we direct the reader to Stowell (2014) and Checkland (1999, p. A20) for further discussion of commodities and "r"). Participants agreed that PEARL and CATWOE (Checkland & Poulter, 2006) were useful as a means of structuring the debate because it focused them on the specifics and forced them to discuss their motivations. The process made it more difficult to retain hidden agendas. Using questions related to the elements of PEARL enabled participants to see the plethora of views on the activities in the AM and encouraged a systemic, holistic appreciation of the situation.

We also asked participants to reflect on the activities in the AM by using the T and W from CATWOE to help them think about why they viewed the situation in a particular way (W) and the transformation (T) that they hoped to achieve by getting access to relevant information. The exercise focused on exposing participants to each other's world views and associated motivation to generate shared learning about the situation and reducing conflicts (e.g., conflict between librarians and senior managers on their differing perceptions of relevant information). The discussions encouraged participants to explain their reasoning in a more explicit way, which helped others understand the thinking behind their contributions. Librarians confirmed that the deeper investigation of the issues using PEARL and CATWOE enabled them to understand the senior managers' plans and that surfacing and discussing the issues helped diffuse the situation. Participants acknowledged that the questions corresponding to the elements from PEARL and CATWOE helped them to organize their thoughts and explain their plans in a more structured and comprehensible fashion. This situation suggests that CATWOE and PEARL assisted in diffusing a potentially conflictual situation between librarians and senior managers by creating an environment of shared learning.





**Figure 5. Relationship Diagram for "Decide How to Provide Relevant Information" Activity**

In the next step, we produced a relationship diagram that served as a precursor to the initial IT specification (use-case diagram). We based the relationship diagram (see Figure 5 below) on the answers that participants gave to the PEARL questions in relation to "what ought to be the case" for the activities in the AM and helped document the value judgements participants made. Answers that participants gave to P in PEARL became roles in the relationship diagram, and the answers to E, A, and r in PEARL became relationships in the relationships diagram.

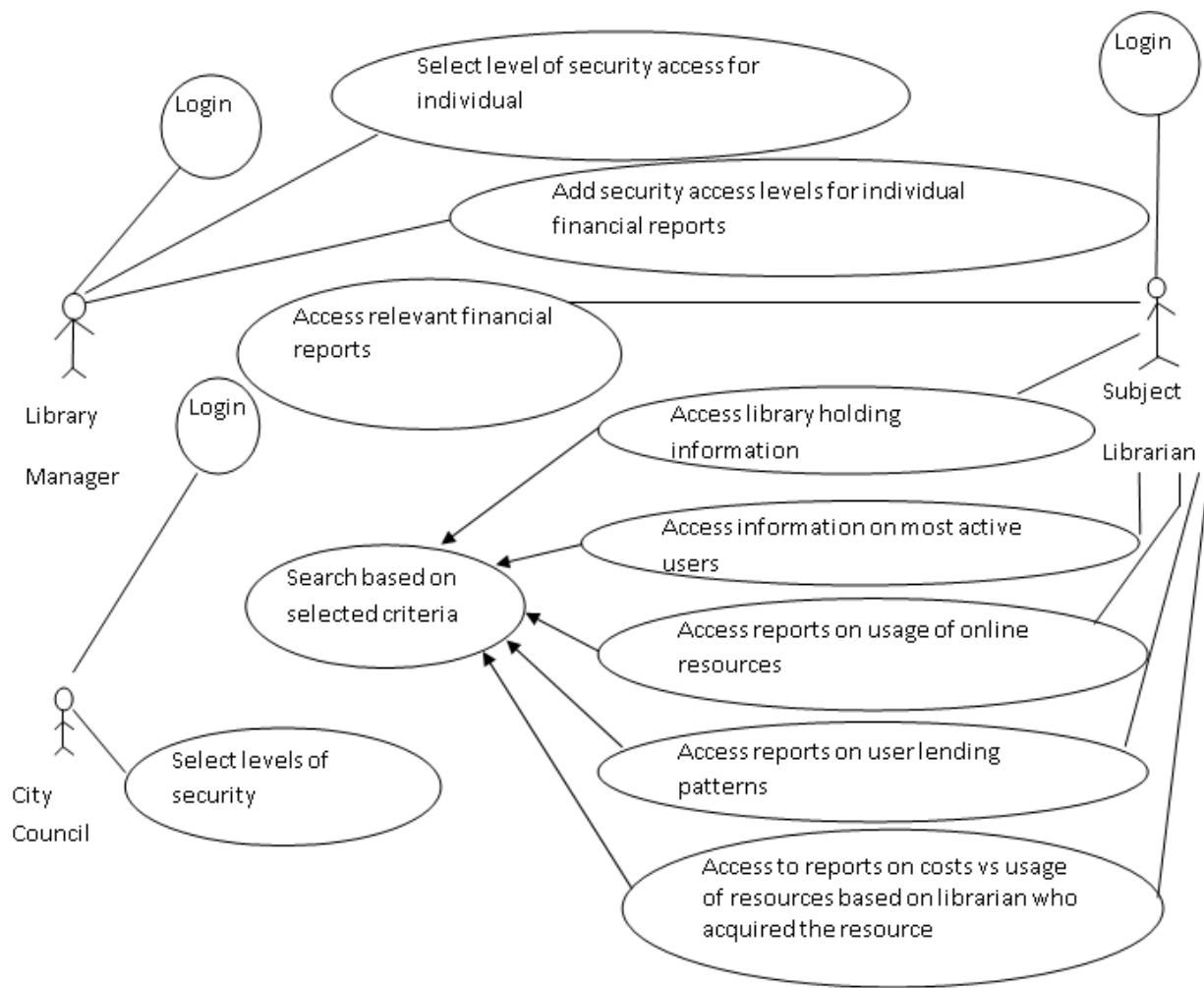
We then invited the participants to discuss the feasibility of implementing the relationships in the relationships diagram in the real world, which corresponds to Vickers' (1983b) action judgments. Action judgements address the mismatch between "what is the case" (reality judgements) and "what ought to be the case" (value judgements). We then asked participants to consider the relationships, processes, and attitudes that might be influenced if the relationships in the relationship diagram were implemented. Participants undertook this task to produce an agreed-on relationship diagram. Table 5 below summarizes the session with one group.

**Table 5. Key Activities of Phase Three**

<b>Key activities of phase three</b>	<b>How the participant group responded to activity</b>	<b>Analyst's role in activity</b>	<b>Additional remarks</b>
Participants think about the feasibility of implementing the relationships in the relationship diagram in the real world.	There appeared to be a shift away from the problem theme (selected in phase two) by one participant. All other participants confirmed the applicability of the problem theme that had emerged the study.	Introduced the relationship diagram and showed participants how it could be traced back to their answers given to the PEARL questions.	In order to help participants reflect on the implications of operationalizing the relationships in the relationships diagram, we invited participants to think about the relations, attitudes, and processes that may be influenced if the said relationships were implemented.
Participants modify the relationships diagram to include only the relationships that they consider as feasible to be implemented in the real world.	The participants deleted the relationships that they considered to be unfeasible from the relationship diagram.	Participants could modify the relationships diagram as they saw fit.	

Once the participants modified the relationships diagram to include only the selected feasible relationships, the participants and we considered if technology support would be needed to implement any of the relationships. The participants agreed that relationships 4-10 (see Figure 5) would need software support and 1-3 and 11 would not. At this stage, we showed participants an example of a use-case diagram and explained that its purpose was to show the desired behavior or interactions between users and potential software. We explained that developers can use use-case diagrams to ensure that software aligns well with clients' strategic goals and needs. We then proceeded to produce a use-case diagram using the relationship diagram that participants came up with previously. We did this task with the participants so they could easily provide input and feedback. Figure 6 shows the use-case diagram (IT specification) that represents the relationships 4-10 from the relationship diagram. The diagram shows how participants expect to interact with the software that will help them implement these relationships.

While the use-case diagram depicts how the participants expected the software to behave, it is but one in a portfolio of models that staff could use to advocate for their needs and IT services could use to pick a software package that better fits with staff members' needs. Other models in the portfolio include the rich pictures (both individual and composite), root definitions, activity models for the different definitions, relationship diagrams for the significant activities, a list of information needs, and a collection of use-case diagrams. While the tangible models provide a visual agenda for discussion that staff can use to advocate for themselves, one cannot understate the importance of the intangible benefits. Participants stated that they could take a step back from their current perceptions of the situation and collectively learn about the domain by becoming exposed to different views. This learning led to their identifying underlying problem themes and more accurate solutions. The learning also provided participants with the knowledge to discuss their needs with IT services in a more informed manner. The two members of IT services stated that the portfolio of models and subsequent discussions with participants lead to a better understanding of what the staff wanted. They also said that understanding the big picture (system) would be useful with integration issues common in the implementation of software packages. Since the library had already decided on buying a software package rather than developing software in house, the study ended when we delivered the portfolio of models. Stage three took approximately six hours to complete.



**Figure 6. Extract of IT Specification/ Use-case Diagram**

In the three months after we delivered the portfolio, IT services together with the staff decided on which software package to purchase. While the final outcome was not vastly different from what the library originally planned (IT services had narrowed the choice down to two packages at the time the study commenced), when contacted after the decision was made, both staff and IT services said that the conversations were easier and that the staff were more comfortable with the upcoming implementation. When we contacted the chief librarian two years later, she stated that the new technology seemed to be working well and that the customizations that the library needed were relatively minor. She said she believed that the reduced customization and relatively smooth transition was partly due to staff members' being able to voice and discuss their concerns, which led to a sense of increased ownership of the new software, which she said was evident when some staff stepped up as champions and vocal advocates of the software. She also noted that the library did not lay off any staff as a result of the new software. Some staff had left, but this was unlikely to have been related to the new computer system.

While the study provided insight on the use of our ideas in an environment where one decided to buy rather than build software, we acknowledge that the situation may differ when one builds software from scratch. Even in the case of customized software, the portfolio of models (including the use-case diagram that belongs to the UML standard) provide an initial specification that developers can use to develop software and is in a format that IT developers can understand and work with (Champion, Stowell, & O'Callaghan, 2005; Cooray 2010). Indeed, research demonstrates that use-case diagrams can lead to class diagrams and eventual software code (Ashrafi & Ashrafi, 2008). Clients can also use use-case diagrams to test if software provides the behavior they desire.

Figure 7 encapsulates the framework that we followed.

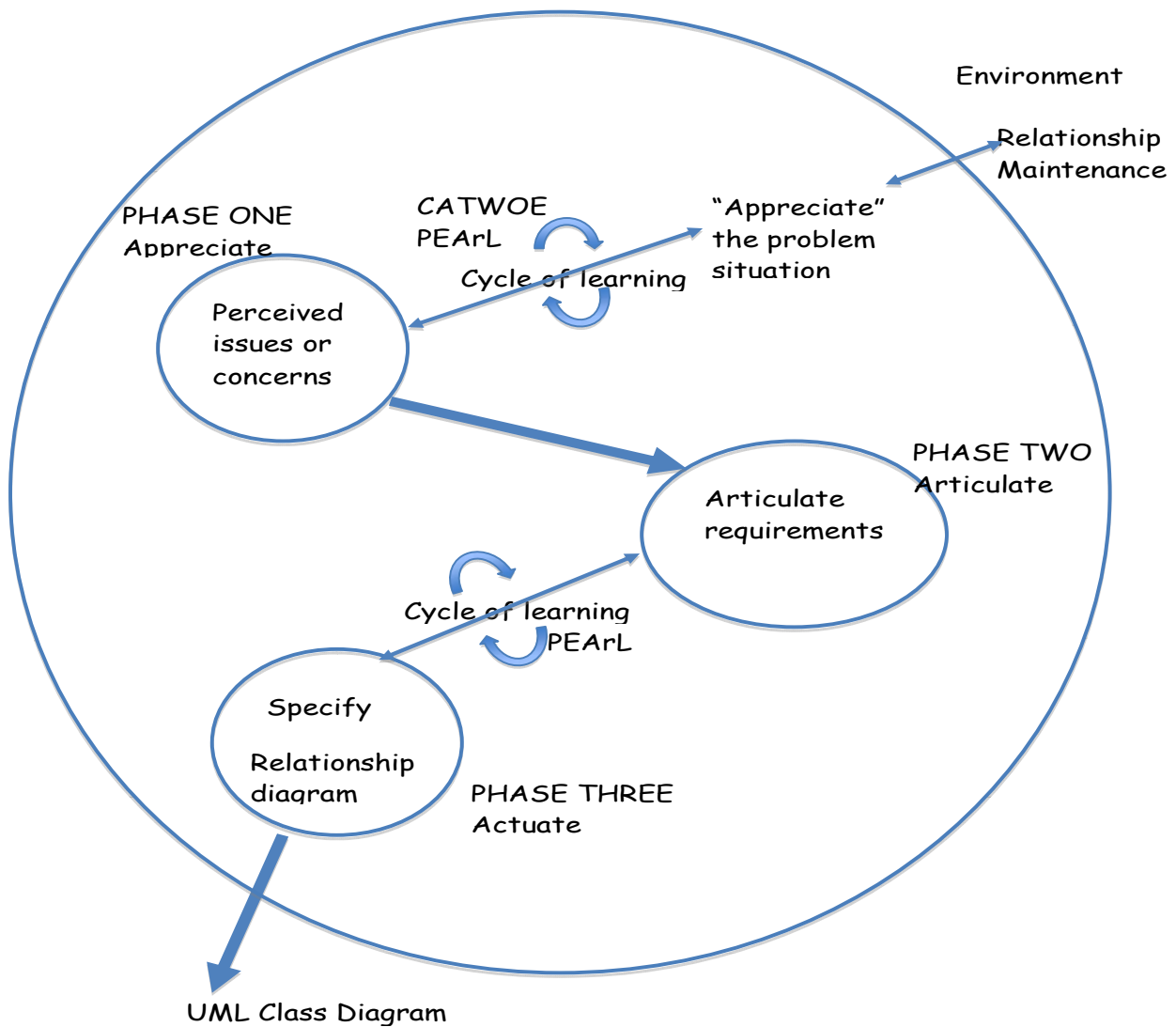


Figure 7. Appreciate, Articulate, and Actuate as a Cycle of Learning

## 5 Discussion

The outcomes and portfolio of models that participants produced during this study and subsequent studies evidenced the practical value of the appreciation cycle (e.g., Stowell & Cooray, 2016a; Cooray, 2012). These models showed that the clients could use Vickers' (1983b) appreciation cycle to navigate from their initial understanding of the problem situation through to a use-case diagram (Stowell & Welch, 2012, p. 141) that showed the software's desired behavior. One can consider the use-case diagram as an initial IT design model since research (Ashrafi & Ashrafi, 2008) has shown that one can convert use-case diagrams into class diagrams that form the basis of object-oriented IT design. Contrary to the industry standard of software engineers' producing design specifications, it seemed possible for the participants to encapsulate their requirements into a design specification such as a use-case diagram without involving software engineers. In the study, the participants could produce the models without prior training since the tools they used were simple and required no technical expertise and since each model mapped onto the next, which increased traceability. As a result, our minimal guidance was sufficient to achieve a successful outcome. Although software engineers were not involved in this study, we were well versed in action research and could provide participants with information about the models (see Stowell, 2012). We suggest that, when transferring these ideas to other settings, a manager or staff member be provided with basic training in AR and the models used so that this individual can provide the staff with information needed to produce a cycle of learning. We caution against using a software engineer or IT specialist for

this task and advocate involving a suitable member of staff to ensure that control lies with the business users. This project provided supporting evidence that, by using simple “tools”, the participants could learn their way to a use-case diagram without influence from a member of the IT domain (e.g., software engineer). The models clients produced would also provide an aide memoir at a later stage for them to evaluate the technical provision.

Our findings have implications for current IS development practices and challenges. Agile development methods that have recently gained much attention (Agile, 2004; Boehm, 2003), which include practices such as short iterations, frequent releases, and simple designs. In particular, agile methods advocate a minimal-requirements determination phase with minimal documentation. In agile methods such as XP, a customer representative is expected to be co-located with the IT team (Ramesh et al., 2010) and be available at all times to answer questions from the IT team so that requirements could be gathered throughout the development process. Although limited research on the requirements-elicitation aspect of agile methods exists (Erickson et al., 2005), research has shown that agile development may have a negative impact on the requirement principles of purposefulness, appropriateness, and truthfulness (Pineiro, 2002). For example, XP relies almost completely on oral communication with the customer representative.

We argue that the ideas discussed in this paper can provide insight on addressing some of the reported challenges of eliciting requirements in agile methods. For example Ramesh et al. (2010) found that, although a customer representative should be co-located with the IT team in theory, onsite customer representation is difficult to attain and many firms face limited access to customers in practice. In their study of 16 organizations that used agile methods, none of them had real onsite customers: they used only product managers as surrogates mostly on a part-time basis. Additionally, they found that establishing trust between client and developer can be challenging in agile development. One can link the reduced trust between clients and developers to the idea that only one on-site client representative acts as the intermediary between developers and other clients, which reduces the direct interaction that the IT team has with the majority of clients. One can also link it to the lack of documentation, which means clients cannot adequately verify the developer’s work, leading to reduced trust and increased the risk of inadequately inspected requirements (Ramesh et al., 2010); a situation that is exacerbated with high staff turnover.

We argue that our approach can provide insight on addressing the reported challenges of requirements-elicitation practices in agile methods. First, we suggest that all or most clients be given the opportunity to identify their requirements collectively and produce a portfolio of models in a format that developers can work with. We suggest that this group learning is important since it allows clients to become exposed to alternative views and see the situation is different ways. By relying on the views of an onsite client representative, agile methods risk a situation in which one ascertains requirements from a narrow, personal perspective. Second, we show that clients can produce a set of traceable, simple, models to document their learning process when eliciting their requirements. These models allow clients to verify that the IT team has met their requirements and provides a visual agenda for discussions with IT personnel. For example, clients can compare the behavior represented in the use-case diagram with the actual behavior that the software exhibits. Indeed, Paetsch, Eberlein, and Maurer’s (2003) research shows that agile RE should include more detailed requirements verification in the process. Although the additional documentation and shared learning in our approach may seem counter-intuitive to agile principles, we posit that the additional time and effort is justified when one considers the reduction in later software customization, increased client-developer trust, and increased client ownership of the final product.

The approach we report here suits IS projects where changes to the business environment happen at a low to moderate frequency (such as the Central Library). We have not yet explored the feasibility of the ideas in a rapidly changing business environment. However, one should not assume that managers in a city library do not experience many of the pressures that managers in commercial organizations do as they are subjected to similar demands as their commercial counterparts (e.g., stock control, economic pressures brought about by online bookstores). Part of the weakness in this process lies in the fact that the library staff had no control over purchasing the customer system, which was a decision taken by the city council and presented them was a fait accompli. However, and importantly, *the shaping* of the way in which it would be operated and staff members’ appreciation about how it could be used to benefit them were positive outcomes.

In this study, the approach demonstrated its usefulness even when it had been decided to buy commercial software rather than build it in house. In situations where an organization decides to buy a software package, its IT department typically has control over selecting it, and client interaction is kept to a minimum. We argue that, in such situations (as in our study), clients can identify their information requirements and how they expect the software to behave. We believe that doing so increases client ownership and helps IT staff to find a package that best fits business users' requirements. While we have not empirically studied our approach's suitability for a IS project that builds software in house, independent software engineers who reviewed the portfolio of models from this study maintained that they could understand the models and use them in further design to develop software. The outcome of this study supported the lessons learnt from a similar exercise that Champoin et al. (2005) undertook.

We conducted our study in a face-to-face environment, and the participants (less than 20) could meet individually and in groups during the process. In situations with more stakeholders in multiple business units, one might need to select multiple user representatives from each business unit to take part in the requirements-analysis phase. The user representatives can then use the simple non-technical models produced during the process as a visual agenda to keep other users informed and obtain their opinion. Another study (Stowell & Cooray, 2016a) shows that one can conduct virtual action research using synchronous virtual meeting software. In situations with geographically dispersed stakeholders, one should also consider such an approach (Stowell & Cooray, 2016a).

Of the tools used in the development process, the mnemonic PEARL that stands out, which is noteworthy because we tested it for the first time as a tool for eliciting knowledge and resolving conflict. Our results suggest that using PEARL as a tool to elicit knowledge was effective and worth exploring in further studies. We also found evidence that it prompted participants to reflect on and discuss "difficult" issues. Often, these issues are implicit and unspoken, such as informal authority ("r") (see Stowell, 2014; Checkland, 1999, p. A20). Other sense making tools used in ISD do not address issues such as these, which are often informal but usually significant. Our study suggests PEARL is a good candidate to fill this omission and help address the problem of misinformation. As such, we need to investigate it further (see Table 2 above and Stowell & Welch, 2012, pp. 145-147, 149-151, 155, xxxiv-xxxix).

Although most of the discussions between the participants in our study did not have conflict, phases two and three had several conflictual situations between librarians and senior managers. We found that, initially, participants discussed the conflictual issues in an existential or high-level context and rarely explained their frame of mind or motivations for their assertions, which led to more accusations and tension. To reduce tension and focus the discussion around the specifics, we used the strategy of exploring the problem issue in the context of the elements from PEARL and CATWOE (Checkland, 1981). The tools allowed participants to draw attentions away from how they felt emotionally and identify underlying personal motivations for their views of the conflict. They then used the elements in PEARL and CATWOE to structure their assertions and present their reasoning in a more organized fashion. Senior managers stated that the questions corresponding to the elements from PEARL and CATWOE helped them to organize their thoughts and explain their plans in a more structured and comprehensible fashion. All participants agreed that the process helped them to reduce tensions and arrive at an agreement. This provides some level of support to using AR (e.g., Vickers' (1983b) appreciation, PEARL, and CATWOE) to create shared understanding leading to reduced conflicts in teams.

Our documents throughout the study provide evidence that, in all three phases of the approach, participants changed their perceptions about how they viewed how the library functioned. Several participants confessed that they had not understood the wider picture at first but that listening to each person explain their views was a valuable learning process (see Stowell & Welch, 2012, pp. xvii). One can see as much when participants engaged in debate and then modified their models without our encouragement (Stowell & Welch, 2012, pp. xxxii). The approach we used fulfilled the criteria that we suggest above; namely: 1) the participants or clients should be able to learn about the wider "system" and not just their part in it, 2) they should be able to express their opinions without the inquiry method getting in the way (i.e., a method that does not have to satisfy the basic principles of its own design), and 3) the clients should be able to describe the requirements to the IT expert in a way relevant to them and to the expert. We argue that one can reduce the impact of analyst-introduced misinformation and IT expert bias (Sarker & Valacich, 2010) by allowing clients to learn their way through the "information systems design" process to a use-case diagram. By doing so, we can increase the chances that IS will align with clients' goals.

## 6 Conclusion

Research has identified the relationship between IS failure and poorly defined requirements, yet IS failures continues to rise (Dwivedi et al., 2015). One reason for this increasing failure rate could be because few methods of inquiry can address such issues as the misinformation effect and IT expert bias. Moreover, few methods explicitly focus on client learning. Researchers such as Majchrzak et al. (2005), Ciborra and Lanzara (1994), and Salaway (1987) have indicated that client learning can enhance client-developer dialogue and ISD in general especially when implemented at an early stage. Despite their suggestions, an approach that focuses on client learning that culminates in a design specification has yet to emerge. In this paper, we develop a method that addresses these issues. Our approach builds on an established idea, and we develop it experientially to support the process of learning that helps participants to make sense of their situation. The appreciative cycle was the guiding principle that enabled the clients to navigate the gap between an activity model and use-case diagram. By using PEARL to operationalize Vickers' (1983a, 1983b) appreciation, we expand its scope beyond the original notion. The results provide some evidence that it is an effective tool for eliciting knowledge and resolving conflict. In our study, participants could learn to a use-case diagram with our help and without IT developers' involvement. They could do so since the models they used were simple. More importantly, each model provided the framework for the next model. Participants could build a use-case diagram with researcher support since the framework for the use-case diagram was already built in the previous relationship diagram. The approach provided a pathway for clients to learn about their perceptions about the situation (i.e., why they perceived the situation in such a way) and about others' perceptions before thinking about requirements. The participants considered requirements only after participants engaged in this cycle of learning. This study provides useful insight into how to focus on client learning before eliciting requirements from them. The results provide the basis for further research into client-driven IS development by addressing challenges in requirements elicitation of agile methods, in reducing the misinformation effect and IT expert bias, and in eliciting requirements in organizations that choose to buy software packages rather than build software in house.

Our results suggest that that approaching ISD in this way, with the emphasis on learning, offers a way to address the problem of analyst-introduced misinformation. One can reduce analysts' influence, at least initially, during the ISD process by providing clients with the tools to navigate the gap and then to define requirements in an IT specification (see Figure 6) that all participants and IT specialists understand (Cooray, pp xlvi). The IT specialist can then engage with clients and discuss the feasibility of the technical options proposed. Importantly, the clients can still retain control by using the models that they created as an agenda for discussion and to ensure that the design stage accurately represents the requirements they identified during analysis.

The lessons learnt from the research adds to literature on information systems development (e.g., Stowell & West, 1974; Avison et al., 1998; Mingers, 1995; Langefors, 1995; Peppard, 2001; Avital et al., 2006; Sarker & Valacich, 2010) and add insight into the value of client learning to this process. The ideas used in the study may also be valuable for software-intensive development methods such as agile. Agile methods advocate a shorter analysis phase and minimal documentation, but they poorly develop requirements (Ashrafi & Ashrafi, 2008; Eberlin & Leite, 2007; Paetch, Eberlein, & Maurer, 2003). By embracing the notion of appreciation and navigation, we provide the means for agile clients to learn about their problem situation as a "whole" or "system" and define their requirements with minimal influence from developers. The models produced provide a means to easily document the stepwise learning process that clients go through without spending too much time or effort on formal text-based documentation. Agile developers could use the IT specification (and other models) that clients produce as a foundation when prioritizing requirements to implement (product backlog) and dividing implementation into time boxes. Although further research is warranted, results from the study shows that there is potential to use these ideas as a front end to agile methods in order to improve the requirements-development process.

While the results of the pilot study (Cooray, 2007) and the later Portsmouth City Council field study are encouraging, we still need further research. In this instance, the participants were used to learning new things and were keen to play a part in developing the new computer system. The lessons learnt about a new way of enabling the client to guide the development process will make a useful contribution to the literature on analyst-introduced misinformation, client learning in ISD, and agile software development.

## References

- Agile. (2004). *Principles behind the Agile Manifesto*. Retrieved from <http://agilemanifesto.org/principles.html>
- Appan, R., & Browne, G. (2012). The impact of analyst induced misinformation on the requirements elicitation process, *MIS Quarterly*, 36(1), 85-106.
- Argyris, C. (1995). Action science and organizational learning. *Journal of Managerial Psychology*, 10(6), 20-26.
- Armson, R. (2011). *Growing wings on the way: Systems thinking for messy situations*. Axminster, UK: Triarchy Press.
- Ashrafi, N., & Ashrafi, H (2008). *Object oriented systems analysis and design*. New York: Prentice Hall.
- Avison, D. E., Lau, F., Myers, D. M., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, 42(1), 94-97.
- Avison, D., Wood-Harper, A. T., Vidgen, R. T., & Wood, J. R. G. (1998). A further exploration into information systems development: The evolution of Multiview 2. *Information Technology and People*, 11(2), 124-139.
- Avital, M., Lyytinen, K., Boland, R. J., Butler, B., Dougherty, D., Fineout, M., Jansen, W., Levina, N., Rifkin, W., & Venable, J. (2006). Design with a positive lens: An affirmative approach to designing information and organizations, *Communications of the Association for Information Systems*, 18, 519-545.
- Baskerville, R. (2012). Reviving the IT in IS. *European Journal of Information Systems*, 21(6), 587-591.
- Baskerville, R., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology*, 11(3), 235-246.
- Bell, S., & Wood-Harper, T. (1998). *Rapid information systems development: A non-specialists guide to analysis and design in an imperfect world*. London: McGraw-Hill.
- Bridge. (n.d.). In *Oxford English Dictionary*. Retrieved from [http://oxforddictionaries.com/us/definition/american\\_english/bridge](http://oxforddictionaries.com/us/definition/american_english/bridge)
- Bullock, A., & Trombley, S. (2000). *The new Fontana dictionary of modern thought*. London: Harper Collins.
- Champion D., & Stowell F. A., & O'Callaghan, A. (2005). Client-led information systems creation: Navigating the gap. *Information Systems Journal*, 15(3), 213-231.
- Champion, D., & Stowell, F. A. (2001). PEARL: A systems approach to demonstrating authenticity in information system design. *Journal of Information Technology*, 16(1), 3-12.
- Checkland, P. B., & Poulter, J. (2006). *Learning for action*. Chichester: Wiley.
- Checkland, P. B. (1995). Model validation in soft systems practice. *Systems Research and Behavioural Practice*, 12(1), 47-54.
- Checkland, P. B. (1999). *Systems thinking, systems practice: A thirty year retrospective*. Chichester: Wiley.
- Checkland, P. B., & Holwell, S. (1998). *Information, systems and information systems*. Chichester: Wiley.
- Checkland, P. B., & Scholes, J. (1990). *Soft systems methodology in action*. Chichester: Wiley
- Checkland, P. B. (1981). *Systems thinking, systems practice*. Chichester: Wiley.
- Churchman, C. W. (1971). *The design of inquiring systems: Basic concepts of systems and organizations*. New York: Basic Books.
- Ciborra, C. U., & Lanzara, G. F. (1994). Formative contexts and information technology: Understanding the dynamics of innovation in organizations, accounting, management, and information technology, 4(2), 61-86.



- Cohn, C. (2014). Build vs. buy: How to know when you should build custom software over canned solutions. *Forbes*. Retrieved from <http://www.forbes.com/sites/chuckcohn/2014/09/15/build-vs-buy-how-to-know-when-you-should-build-custom-software-over-canned-solutions/#3c4c37044849>
- Cooray, S., & Stowell, F. (2007). User driven IS development: Navigating the gap. In *Proceedings of the United Kingdom Systems Society Conference*.
- Cooray, S. (2010). *End user driven development of information systems: Revisiting Vickers notion of appreciation* (Doctoral thesis). Portsmouth: University of Portsmouth.
- Stowell, F., & Welch, C. (2012). *The managers guide to systems practice: Making sense of complex problems*. Chichester: Wiley.
- Davison, R., Martinsons, M., & Ou, C. (2012). The roles of theory in canonical action research. *MIS Quarterly*, 36(3), 763-786.
- DeMarco, T., & Lister, T. (1987). *Peopleware: Productive projects and teams*. New York: Dorset House Publishing
- Doherty, N., & King, M (2005). From technical to socio technical change: Tacking the human and organizations aspects of systems development project. *European journal of Information Systems*, 14(1), 1-5.
- Dwivedi, Y. K., Wastell, D., Laumer, S., Zinner Henriksen, H., Myers, M. D., Bunker, D., & Srivastava, S. C. (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1), 143-157
- Eberlein, R., & Leite, J. (2007). *Agile requirements definition: A view from requirements engineering*. Retrieved from <http://www.enel.ucalgary.ca/tcre02/papers/Eberlein.pdf>
- Fortune, J., & Peters, G. (2005). *Information systems: Achieving success by avoiding failure*. Chichester: Wiley and Sons.
- Fowler, J., Horan, P., & Cope, C. (2007). How an "imperative" IS development was saved from a failing course of action—a case study. *Journal of Issues in Informing Science and Information Technology*, 4, 395-406.
- Gadamer, H. G. (1998). *Truth and method* (2nd ed.). New York: Continuum.
- Graham, I. (1994). *Object oriented methods*. Wokingham: Addison Wesley.
- Hardy-Vallee, B. (2012). The cost of bad project management. *Gallup*. Retrieved from <http://www.gallup.com/businessjournal/152429/cost-bad-project-management.aspx>
- Husserl, E. (2002). Pure phenomenology, its method and its field of investigation. In D. Moran & T. Mooney (Eds.), *The phenomenology reader* (pp. 124-133). London: Routledge.
- Iversen, J., Mathiassen, L., & Nielsen. P. A. (2004). Managing risk in software process improvement: An action research approach. *MIS Quarterly*, 28(3), 395-433.
- Ison, R. (2005). Guest editorial: Geoffrey Vickers 2004: Contemporary applications and changing appreciative settings. *Systems Research and Behavioural Science*, 22(4), 277-284.
- Jary, D., & Jary, J. K. (1995). *Collins dictionary of sociology*. London: Harper Collins.
- Kirsch, L. J., & Beath, C. M. (1996). The enactments and consequences of token, shared, and compliant participation in information systems development. *Accounting, Management, and Information Technology*, 6(4), 221- 254.
- Langefors, B. (1995). *Essays on infology* (B. Dahlbom, Ed.). Lund: Studentlitteratur.
- Ledeen, K. S. (n.d.). *Build v. buy: A decision paradigm for information technology applications*. Retrieved from <http://www.nevo.com/our-knowledge/whitepapers/BuildVsBuy.pdf>
- Lin, A., & Silva, A. (2005). The social and political construction of technological frames. *European Journal of Information Systems*, 14(1), 49-59.
- Liang, Y., West, D., & Stowell, F. A (1998). An Interpretivist Approach to IS definition using object modelling. *Information Systems Journal*, 8, 163-180.

- Majchrzak, A., Beath, C. M., Lim, R. A., & Chin, W. W. (2005). Managing client dialogues during information system design to facilitate client learning. *MIS Quarterly*, 29(4), 653-672.
- McGinn, M. (1997). *Wittgenstein and the Philosophical Investigations*. New York: Routledge.
- Mingers, J. (1995). Information and meaning: Foundations for an intersubjective account. *Information Systems Journal*, 5, 285-306.
- Morris, S., & Travis, A. (2001). Straw calls halt to £80m IT system. *The Guardian*. Retrieved from <http://www.theguardian.com/politics/2001/feb/16/Whitehall.uk>
- Mukherjee, I. (2008). Understanding information system failure from the complexity perspective. *Journal of Social Sciences*, 4(4), 309-319.
- Mumford, E., & Henshall, D. (1979). *A participative approach to computer system design*. London: Associated Business Press.
- Nelson, R. R. (2007). IT project management: Infamous failures, classic mistakes, and best practices. *MIS Quarterly Executive*, 6(2), 67-78.
- Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. In *Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 308-314). IEEE.
- Peppard, J. (2001). Bridging the gap between IS organisation and the rest of the business: Plotting a route. *Information Systems Journal*, 11(3), 249-270.
- Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 449-480.
- Revas, R. W. (1982). *The origin and growth of action learning*. Brickley, UK: Chartwell-Bratt.
- Rosenkranz, C., Charaf, M., & Holten, R. (2013). Language quality in requirements development: Tracing communication in the process of information systems development. *Journal of Information Technology*, 28(3), 198-223.
- Salaway, G. (1987). An organizational learning approach to information systems development. *MIS Quarterly*, 11(2), 245-264
- Sarker, S., & Valacich, J. (2010). An alternative to methodological individualism: A non-reductionist approach to studying technology adoption by group. *MIS Quarterly*, 34(4), 779-808.
- Schutz, A. (2011). *Collected papers V. Phenomenology and the social sciences* (L. Embree, Ed.). New York: Springer.
- Schutz, A. (1962-1966). *Collected papers* (Vols. I, II, & III). The Hague: Nijhoff.
- Schiff, J. (2014). 9 Tips for implementing an ERP Solution. *CIO*. Retrieved from <http://www.cio.com/article/2458889/enterprise-resource-planning/9-tips-for-selecting-and-implementing-an-erp-system.html>
- Shipp, K. (2005). *Systems thinking and practice: Diagramming*. Milton Keynes: Open University.
- Susman, G., & Evered, R. (1978). An Assessment of the merits of scientific action research. *Administrative Science Quarterly*, 23, 583-603.
- Stewart, R. (2007). *Change Management and Information Systems*. New York: Springer.
- Stowell, F. A. (2014). Organisational power and the metaphor commodity. *International Journal of Systems and Society*, 1(1), 12-20.
- Stowell, F. A. (2012). The appreciative inquiry method—a suitable framework for action research? *Systems Research and Behavioural Science*, 30(1), 15-30.
- Stowell, F. A., & Cooray, S. (2016a). Virtual action research for virtual organisations. *Systemic Practice and Action Research*, 29(3).
- Stowell, F. A., & Cooray, S. (2016b). Using soft systems ideas within virtual teams. In C. Graham (Ed.), *Strategic management and leadership for systems development in virtual spaces* (pp. 247-269). New York: Business Science.

- Stowell, F. A., & Welch, C. (2012). *The managers guide to systems practice: Making sense of complex problems*. Chichester: Wiley.
- Stowell, F. A., & West, D. (1994). *Client-led design: A systemic approach to information systems definition*. London: McGraw-Hill.
- Stowell, F. A. (1991). Towards client-led development of information systems. *Information Systems Journal*, 1, 173-189.
- Stowell, F. (1985). Experience with soft systems methodology and data analysis. *Information Technology Training*, 48-50.
- Vickers, G. (1983a). *The art of judgement*. London: Harper and Rowe.
- Vickers, G. (1983b). *Human systems are different*. London: Harper and Rowe
- Wastell, D. G. (1999). Learning dysfunctions in information systems development: Overcoming the social defenses with transitional objects. *MIS Quarterly*, 23(4), 581-600.
- Winter, M. C., Brown, D. H., & Checkland, P. B. (1995). A role for soft systems methodology in information systems development. *European Journal of Information Systems*, 4(3), 130-142.
- Wood-Harper, A. T., Antill, L., & Avison, D. E. (1985). *Information systems definition: The multiview approach*. Oxford: Blackwell Scientific Publications.

## About the Authors

**Frank Stowell** is a Professor of Systems and Information Systems at the School of Computing, University of Portsmouth, UK. He has a PhD in Organisational Change and is a past President of the UK Academy of Information Systems and the UK Systems Society. Presently he is the chair the Council of Information Systems Professors and is on the Board of the World Organisation of Systems and Cybernetics. He is the Editor in Chief of the *International Journal of Systems and Society* and previously Editor in Chief of the *International Journal of Information Technologies and Systems Approach*. He has previously published in the *Information Systems Journal*, *Journal of Information Technology*, *Systems Research and Behavioral Science*, *Systems Practice and Action Research*, and others.

**Shavindrie Cooray** is an Assistant Professor of Information Systems at the Department of Management, Curry College, USA. She has a PhD from the School of Computing, University of Portsmouth, UK. She has previously published in *Systems Practice and Action Research*, *International Journal of Information Technologies and Systems Approach*, and others. Her work on client led IS design and virtual teams have been presented at several international conferences.

Copyright © 2017 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints or via e-mail from [publications@aisnet.org](mailto:publications@aisnet.org).