# Design Study Methodology: Reflections from the Trenches and the Stacks

Michael Sedlmair, *Member, IEEE*, Miriah Meyer, *Member, IEEE*, and Tamara Munzner, *Member, IEEE*

**Abstract**—Design studies are an increasingly popular form of problem-driven visualization research, yet there is little guidance available about how to do them effectively. In this paper we reflect on our combined experience of conducting twenty-one design studies, as well as reading and reviewing many more, and on an extensive literature review of other field work methods and methodologies. Based on this foundation we provide definitions, propose a methodological framework, and provide practical guidance for conducting design studies. We define a design study as a project in which visualization researchers analyze a specific real-world problem faced by domain experts, design a visualization system that supports solving this problem, validate the design, and reflect about lessons learned in order to refine visualization design guidelines. We characterize two axes—a *task clarity axis* from fuzzy to crisp and an *information location axis* from the domain expert's head to the computer—and use these axes to reason about design study contributions, their suitability, and uniqueness from other approaches. The proposed methodological framework consists of 9 stages: *learn*, *winnow*, *cast*, *discover*, *design*, *implement*, *deploy*, *reflect*, and *write*. For each stage we provide practical guidance and outline potential pitfalls. We also conducted an extensive literature survey of related methodological approaches that involve a significant amount of qualitative field work, and compare design study methodology to that of ethnography, grounded theory, and action research.

**Index Terms**—Design study, methodology, visualization, framework.

## 1 INTRODUCTION

Over the last decade design studies have become an increasingly popular approach for conducting problem-driven visualization research. Design study papers are explicitly welcomed at several visualization venues as a way to explore the choices made when applying visualization techniques to a particular application area [55], and many exemplary design studies now exist [17, 34, 35, 56, 94]. A careful reading of these papers reveals multiple steps in the process of conducting a design study, including analyzing the problem, abstracting data and tasks, designing and implementing a visualization solution, evaluating the solution with real users, and writing up the findings.

And yet there is a lack of specific guidance in the visualization literature that describes holistic methodological approaches for conducting design studies—currently only three paragraphs exist [49, 55]. The relevant literature instead focuses on methods for designing [1, 42, 66, 79, 82, 90, 91] and evaluating [13, 33, 39, 50, 68, 69, 76, 80, 85, 86, 95] visualization tools. We distinguish between methods and methodology with the analogy of cooking; *methods* are like ingredients, whereas *methodology* is like a recipe. More formally, we use Crotty's definitions that methods are "techniques or procedures" and a methodology is the "strategy, plan of action, process, or design lying behind the choice and use of particular methods" [18].

From our personal experience we know that the process of conducting a design study is hard to do well and contains many potential pitfalls. We make this statement after reflecting on our own design studies, in total 21 between the 3 authors, and our experiences of reviewing many more design study papers. We consider at least 3 of our own design study attempts to be failures [51, 54, 72]; the other 18 were more successful [4, 5, 10, 40, 43, 44, 45, 46, 52, 53, 67, 70, 71, 73, 74, 75, 77, 78].

In the process of conducting these design studies we grappled with many recurring questions: What are the steps you should perform, and in what order? Which methods work, and which do not? What are the potential research contributions of a design study? When is the use of visualization a good idea at all? How should we go about collaborating with experts from other domains? What are pitfalls to avoid? How and when should we write a design study paper? These questions motivated and guided our methodological work and we present a set of answers in this paper.

We conducted an extensive literature review in the fields of human computer interaction (HCI) [7, 8, 9, 12, 16, 19, 20, 21, 22, 25, 26, 27, 28, 29, 30, 31, 38, 47, 57, 63, 64, 65, 83] and social science [6, 14, 18, 24, 32, 62, 81, 87, 93] in hopes of finding methodologies that we could apply directly to design study research. Instead, we found an intellectual territory full of quagmires where the very issues we ourselves struggled with were active subjects of nuanced debate. We did not find any off-the-shelf answers that we consider suitable for wholesale assimilation; after careful gleaning we have synthesized a framing of how the concerns of visualization design studies both align with and differ from several other qualitative approaches.

This paper is the result of a careful analysis of both our experiences in the "trenches" while doing our own work, and our foray into the library "stacks" to investigate the ideas of others. We provide, for the first time, a discussion about design study methodology, including a clear definition of design studies as well as practical guidance for conducting them effectively. We articulate two axes, *task clarity* and *information location*, to reason about what contributions design studies can make, when they are an appropriate research device, and how they are unique from other approaches. For practical guidance we propose a process for conducting design studies, called the *nine-stage framework*, consisting of the following stages: *learn*, *winnow*, *cast*, *discover*, *design*, *implement*, *deploy*, *reflect*, and *write*. At each stage we identify pitfalls that can endanger the success of a design study, as well as strategies and methods to help avoid them. Finally, we contrast design study methodology to related research methodologies used in other fields, in particular those used or discussed in HCI, and elaborate on similarities and differences. In summary, the main contributions of this paper are:

- definitions for design study methodology, including articulation of the task clarity and information location axes;
- a nine-stage framework for practical guidance in conducting design studies and collaborating with domain experts;
- 32 identified pitfalls occurring throughout the framework;
- a comparison of design study methodology to that of ethnography, grounded theory and action research.

We anticipate that a wide range of readers will find this paper useful, including people new to visualization research, researchers expe-

- *Michael Sedlmair and Tamara Munzner are with the University of British Columbia, e-mail: [msedl,tmm]@cs.ubc.ca.*
- *Miriah Meyer is with the University of Utah, e-mail: miriah@cs.utah.edu.*

*Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.*
*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*

rienced in technique-driven visualization work who are transitioning to problem-driven work, experienced design-study researchers seeking comparison with other methodologies, reviewers of design study papers, and readers outside of the visualization community who are considering when to employ visualization versus full automation.

## 2 RELATED WORK

Only two sources discuss design study methodology, and both are brief. The original call for design study papers [55] contains only a paragraph about expectations, while Munzner [49] elaborates slightly further by defining the critical parts of a design study. Neither of these sources provide specific methodological and practical guidance on how to conduct design studies.

There is, however, a rich source of papers elaborating on models and methods, particularly evaluation methods, that pertain to design studies. Some of the most relevant for design studies include the investigation of Lloyd and Dykes into the early steps of problem analysis and paper prototyping in a longitudinal geovisualization design study, providing interesting insights into which human-centered methods work and which do not [42]; van Wijk's model for understanding and reasoning about the "value of visualization" [88] that provides a lens on the interplay between data, user, and visualization; Amar and Stasko's guidance for problem-driven visualization research by identifying and articulating gaps between the representation and the analysis of data, and provide precepts for bridging these gaps [3]; and Pretorius and van Wijk's arguments for the importance of considering not just the needs of the user, but also the structure and semantics of the data when designing a visualization tool [61].

The majority of other related work on methods deals with the question of how to evaluate visualization designs and tools in real-world settings. Carpendale provides an overview of relevant validation methods in visualization [13] while Munzner provides guidance on when to use which method [50]. Lam et al. conduct a broad literature survey of more than 800 visualization papers and derive seven guiding scenarios describing visualization evaluation [39]. Sedlmair et al. provide practical advice of how to validate visualizations in large company settings, one of many settings in which a design study may be conducted [76]. Finally, many proposed evaluation methods address the specific needs of validating the usefulness of visualization tools such as the multidimensional in-depth long-term case study approach [80], the insight-based method [68, 69], and grounded evaluation [33].

While these papers are excellent resources for specific methods applicable to design studies, the goal of this paper is a higher level articulation of a methodology for conducting design studies.

## 3 CHARACTERIZING DESIGN STUDIES

This section defines key terms, proposes two axes that clarify the potential contributions of design studies, and uses these axes to characterize their contributions and suitability.

### 3.1 Definitions

Design studies are one particular form of the more general category of **problem-driven** research, where the goal is to work with real users to solve their real-world problems. At the other end of the spectrum is **technique-driven** research, where the goal is to develop new and better techniques without necessarily establishing a strong connection to a particular documented user need. The focus in this paper is on problem-driven research, but in the larger picture we argue that the field of visualization benefits from a mix of both to maintain vitality. We define a **design study** as follows:

> *A design study is a project in which visualization researchers analyze a specific real-world problem faced by domain experts, design a visualization system that supports solving this problem, validate the design, and reflect about lessons learned in order to refine visualization design guidelines.*

Our definition implies the following:

- *analysis:* Design studies require analysis to translate tasks and data from domain-specific form into abstractions that a user can address through visualization.
- *real-world problem:* At the heart of a design study is a contribution toward solving a real-world problem: real users and real data are mandatory.
- *design:* Our definition of design is the creative process of searching through a vast space of possibilities to select one of many possible good choices from the backdrop of the far larger set of bad choices. Successful design typically requires the explicit consideration of multiple alternatives and a thorough knowledge of the space of possibilities.
- *validation:* A crucial aspect of our definition is the validation of the problem analysis and the visualization design in the broad sense of Munzner's nested model [50]. We advocate choosing from a wide variety of methods according to their suitability for evaluating the different framework stages, including justification according to known principles, qualitative analysis of results, informal expert feedback, and post-deployment field studies.
- *reflection:* Design becomes research when reflection leads to improving the process of design itself, by confirming, refining, rejecting, or proposing guidelines.

In this paper, we propose a specific process for conducting design studies, the nine-stage framework, described in detail in Section 4. We offer it as a scaffold to provide guidance for those who wish to begin conducting design studies, and as a starting point for further methodological discussion; we do not imply that our framework is the only possible effective approach.

Collaboration between *visualization researchers* and *domain experts* is a fundamental and mandatory part of the nine-stage framework; in the rest of the paper we distinguish between these roles. While strong problem-driven work can result from situations where the same person holds both of these roles, we do not address this case further here. The domain expert role is crucial; attempts to simply apply techniques without a thorough understanding of the domain context can fail dramatically [92].

Conducting a design study using the nine-stage framework can lead to three types of **design study research contributions**, the first of which is a *problem characterization and abstraction*. Characterizing a domain problem through an abstraction into tasks and data has multiple potential benefits. First, this characterization is a crucial step in achieving shared understanding between visualization researchers and domain experts. Second, it establishes the requirements against which a design proposal should be judged. It can thus be used not only by the researchers conducting the design study, but also by subsequent researchers who might propose a different solution to the same problem. Finally, it can enable progress towards a fully automatic approach that does not require a human in the loop by causing relevant domain knowledge to be articulated and externalized. We thus argue for considering this characterization and abstraction as a first-class contribution of a design study.

A *validated visualization design* is the second type of possible contribution. A visualization tool is a common outcome of a design study project. Our definition of design study requires that the tool must be appropriately validated with evidence that it does in fact help solve the target domain problem and is useful to the experts. The validated design of a visualization tool or system is currently the most common form of design study contribution claim.

The third type of contribution is the *reflection* on the design study and its retrospective analysis in comparison to other related work. Lessons learned can improve current guidelines, for example visualization and interaction design guidelines, evaluation guidelines, or process guidelines.

A **design study paper** is a paper about a design study. Reviewers of design study papers should consider the sum of contributions of all three types described above, rather than expecting that a single design study paper have strong contributions of all three. For instance, a design study with only a moderate visual encoding design contribution might have an interesting and strong problem characterization and abstraction, and a decent reflection on guidelines. On the other hand, a very thorough design and evaluation might counterbalance a moderate problem characterization or reflection. Our definitions imply that a design study paper does **not** require a novel algorithm or technique contribution. Instead, a proposed visualization design is often a well-justified combination of existing techniques. While a design study paper is the most common outcome of a design study, other types of research papers are also possible such as technique or algorithm, evaluation, system, or even a pure problem characterization paper [50].

### 3.2 Task Clarity and Information Location Axes

We introduce two axes, task clarity and information location, as shown in Figure 1. The two axes can be used as a way to think and reason about problem characterization and abstraction contributions which, although common in design studies, are often difficult to capture and communicate.

The **task clarity** axis depicts how precisely a task is defined, with `fuzzy` on the one side and `crisp` on the other. An example of a crisp task is "buy a train ticket". This task has a clearly defined goal with a known set of steps. For such crisp tasks it is relatively straightforward to design and evaluate solutions. Although similarly crisp low-level visualization tasks exist, such as correlate, cluster or find outliers [2], reducing a real-world problem to these tasks is challenging and time consuming. Most often, visualization researchers are confronted with complex and fuzzy domain tasks. Data analysts might, for instance, be interested in understanding the evolutionary relationship between genomes [45], comparing the jaw movement between pigs [34], or the relationship between voting behavior and ballot design [94]. These domain tasks are inherently ill-defined and exploratory in nature. The challenge of evaluating solutions against such fuzzy tasks is well-understood in the information visualization community [59].

Task clarity could be considered the combination of many other factors; we have identified two in particular. The **scope** of the task is one: the goal in a design study is to decompose high-level domain tasks of *broad* scope into a set of more *narrow* and low-level abstract tasks. The **stability** of the task is another: the task might change over the course of the design study collaboration. It is common, and in fact a sign of success, for the tasks of the experts to change after the researcher introduces visualization tools, or after new abstractions cause them to re-conceptualize their work. Changes from external factors, however, such as strategic priority changes in a company setting or research focus changes in an academic setting, can be dangerous.

The second axis is the **information location**, characterizing how much information is only available in the `head` of the expert versus what has been made explicit in the `computer`. In other words, when considering all the information required to carry out a specific task, this axis characterizes how much of the information and context surrounding the domain problem remains as implicit knowledge in the expert's head, versus how much data or metadata is available in a digital form that can be incorporated into the visualization.

We define moving forward along either of these axes as a design study contribution. Note that movement along one axis often causes movement along the other: increased task clarity can facilitate a better understanding of derived data needs, while increased information articulation can facilitate a better understanding of analysis needs [61].

### 3.3 Design Study Methodology Suitability

The two axes characterize the range of situations in which design study methodology is a suitable choice. This rough characterization is not intended to define precise boundaries, but rather for guiding the understanding of when, and when not, to use design studies for approaching certain domain problems.

Figure 1 shows how design studies fall along a two-dimensional space spanned by the task clarity and the information location axes. The red and the blue areas at the periphery represent situations for which design studies may be the wrong methodological choice. The red vertical area on the left indicates situations where no or very little data is available. This area is a dangerous territory because an effective visualization design is not likely to be possible; we provide

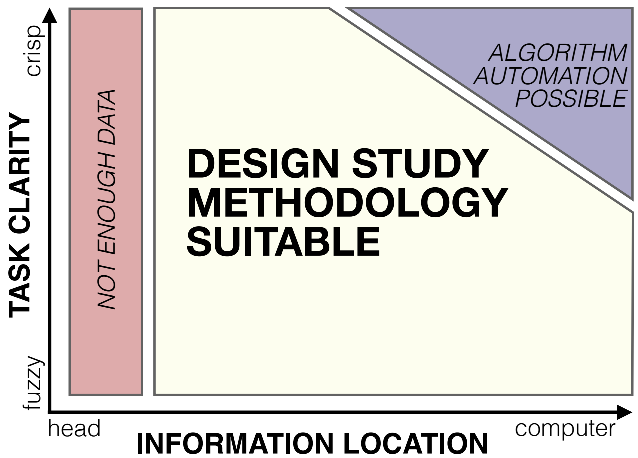


Fig. 1. The task clarity and information location axes as a way to analyze the suitability of design study methodology. Red and blue areas mark regions where design studies may be the wrong methodological choice.

ways to identify this region when winnowing potential collaborations in Section 4.1.2.

The blue triangular area on the top right is also dangerous territory, but for the opposite reason. Visualization might be the wrong approach here because the task is crisply defined and enough information is computerized for the design of an automatic solution. Conversely, we can use this area to define when an automatic solution is *not* possible; automatic algorithmic solutions such as machine learning techniques make strong assumptions about crisp task clarity and availability of all necessary information. Because many real-world data analysis problems have not yet progressed to the crisp/computer ends of the axes, we argue that design studies can be a useful step towards a final goal of a fully automatic solution.

The remaining white area indicates situations where design studies are a good approach. This area is large, hinting that different design studies will have different characteristics. For example, the regions towards the top left at the beginning of both axes require significant problem characterization and data abstraction before a visualization can be designed—a paper about such a project is likely to have a significant contribution of this type. Design studies that are farther along both axes will have a stronger focus on visual encoding and design aspects, with a more modest emphasis on the other contribution types. These studies may also make use of combined automatic and visual solutions, a common approach in visual analytics [84].

The axes can also associate visualization with, and differentiate it from other fields. While research in some subfields of HCI, such as human factors, deal with crisply defined tasks, several other subfields, such as computer supported cooperative work and ubiquitous computing, face similar challenges in terms of ill-defined and fuzzy tasks. They differ from visualization, however, because they do not require significant data analysis on the part of the target users. Conversely, fields such as machine learning and statistics focus on data analysis, but assume crisply defined tasks.

## 4 Nine-Stage Framework

Figure 2 shows an overview of our nine-stage framework with the stages organized into three categories: a **precondition** phase that describes what must be done before starting a design study; a **core** phase presenting the main steps of conducting a design study; and an **analysis** phase depicting the analytical reasoning at the end. For each stage we provide practical advice based on our own experience, and outline pitfalls that point to common mistakes. Table 1 at the end of this section summarizes all 32 pitfalls (*PF*).

The general layout of the framework is linear to suggest that one stage follows another. Certain actions rely on artifacts from earlier stages—deploying a system is, for instance, not possible without some

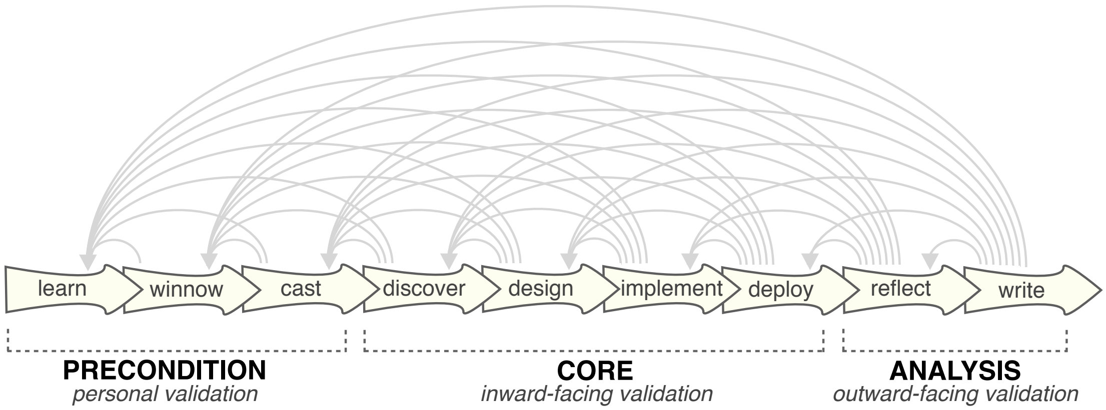


Fig. 2. Nine-stage design study methodology framework classified into three top-level categories. While outlined as a linear process, the overlapping stages and gray arrows imply the iterative dynamics of this process.

kind of implementation—and it is all too common to jump forward over stages without even considering or starting them. This forward jumping is the first pitfall that we identify (*PF-1*). A typical example of this pitfall is to start implementing a system before talking to the domain experts, usually resulting in a tool that does not meet their specific needs. We have reviewed many papers that have fatal flaws due to this pitfall.

The linearity of the diagram, however, does not mean that previous stages must be fully completed before advancing to the next. Many of the stages often overlap and the process is highly iterative. In fact, jumping backwards to previous stages is the common case in order to gradually refine preliminary ideas and understanding. For example, we inevitably always find ourselves jumping backwards to refine the abstractions while writing a design study paper. The overlapping stages and gray arrows in Figure 2 imply these dynamics.

Validation crosscuts the framework; that is, validation is important for every stage, but the appropriate validation is different for each. We categorize validation following the three framework phases. In the precondition stage, validation is **personal**: it hinges on the preparation of the researcher for the project, including due diligence before committing to a collaboration. In the core phase, validation is **inward-facing**: it emphasizes evaluating findings and artifacts with domain experts. In the analysis phases, validation is **outward-facing**: it focuses on justifying the results of a design study to the outside world, including the readers and reviewers of a paper. Munzner's nested model elaborates further on how to choose appropriate methods at each stage [50].

## 4.1 Precondition Phase

The precondition stages of *learn*, *winnow*, and *cast* focus on preparing the visualization researcher for the work, and finding and filtering synergistic collaborations with domain experts.

### 4.1.1 *Learn*: Visualization Literature

A crucial precondition for conducting an effective design study is a solid knowledge of the visualization literature, including visual encoding and interaction techniques, design guidelines, and evaluation methods. This visualization knowledge will inform all later stages: in the *winnow* stage it guides the selection of collaborators with interesting problems relevant to visualization; in the *discover* stage it focuses the problem analysis and informs the data and task abstraction; in the *design* stage it helps to broaden the consideration space of possible solutions, and to select good solutions over bad ones; in the *implement* stage knowledge about visualization toolkits and algorithms allows fast development of stable tool releases; in the *deploy* stage it assists in knowing how to properly evaluate the tool in the field; in the *reflect* stage, knowledge of the current state-of-the-art is crucial for comparing and contrasting findings; and in the *write* stage, effective framing of contributions relies on knowledge of previous work.

Of course, a researcher's knowledge will gradually grow over time and encyclopedic knowledge of the field is not a requirement before conducting a first design study. Nevertheless, starting a design study without enough prior knowledge of the visualization literature is a pitfall (*PF-2*). This pitfall is particularly common when researchers who are expert in other fields make their first foray into visualization [37]; we have seen many examples of this as reviewers.

### 4.1.2 *Winnow*: Select Promising Collaborations

The goal of this stage is to identify the most promising collaborations. We name this strategy *winnowing*, suggesting a lengthy process of separating the good from the bad and implying that careful selection is necessary: not all potential collaborations are a good match. Premature commitment to a collaboration is a very common pitfall that can result in much unprofitable time and effort (*PF-3*).

We suggest talking to a broad set of people in initial meetings, and then gradually narrowing down this set to a small number of actual collaborations based on the considerations that we discuss in detail below. Because this process takes considerable calendar time, it should begin well before the intended start date of the *implement* stage. Initial meetings last only a few hours, and thus can easily occur in parallel with other projects. Only some of these initial meetings will lead to further discussions, and only a fraction of these will continue with a closer collaboration in the form of developing requirements in the *discover* stage. Finally, these closer collaborations should only continue on into the *design* stage if there is a clear match between the interests of the domain experts and the visualization researcher. We recommend committing to a collaboration only after this due diligence is conducted; in particular, decisions to seek grant funding for a collaborative project after only a single meeting with a domain expert are often premature. We also suggest maintaining a steady stream of initial meetings at all times. In short, our strategy is: talk with many but stay with few, start early, and always keep looking.

The questions to ask during the *winnow* stage are framed as reasons to decide against, rather than for, a potential collaboration. We choose this framing because continued investigation has a high time cost for both parties, so the decision to pull out is best made as early as possible. Two of our failure cases underline the cost of late decision-making: the PowerSetViewer [54] design study lasted two years with four researchers, and WikeVis [72] half a year with two researchers. Both projects fell victim to several pitfalls in the *winnow* and *cast* stages, as we describe below; if we had known what questions to consider at these early stages we could have avoided much wasted effort.

The questions are categorized into *practical*, *intellectual*, and *interpersonal* considerations. We use the pronouns *I* for the visualization researcher, and *they* for the domain experts.

PRACTICAL CONSIDERATIONS: These questions can be easily checked in initial meetings.

***Data:*** *Does real data exist, is it enough, and can I have it?*
Some potential collaborators will try to initiate a project before real data is available. They may promise to have the data "soon", or "next

week/month/term"; these promises should be considered as a red flag for design studies (*PF-4*). Data gathering and generation is prone to delays, and the over-optimistic ambitions of potential collaborators can entice visualization researchers to move forward using inappropriate "toy" or synthetic data as a stopgap until real data becomes available. Other aspects of this pitfall are that not enough of the data exists in digital form to adequately solve the problem, or that the researcher cannot gain access to the data.

In our failed PowerSetViewer [54] design study, for instance, real data from collaborators did not materialize until after the *design* and *implement* phases were already completed. While waiting for real data, we invested major resources into developing an elegant and highly scalable algorithm. Unfortunately, we did not realize that this algorithm was targeted at the wrong abstraction until we tested it on real rather than synthetic data.

***Engagement:*** *How much time do they have for the project, and how much time do I have? How much time can I spend in their environment?*
Design studies require significant time commitments from both domain experts and visualization researchers. Although there are ways to reduce the demands on domain experts [76], if there is not enough time available for activities such as problem analysis, design discussions, and field evaluations, then success is unlikely (*PF-5*). Some of these activities also strongly benefit when they can be conducted in situ at the workplace of the collaborators, as we discuss with RelEx [74].

INTELLECTUAL CONSIDERATIONS: These important questions can be hard to conclusively answer early on in a design study, but they should be kept in mind during initial meetings. It is also useful to refer back to these questions later when monitoring progress; if a negative answer is discovered, it might be wise to pull out of the collaboration.

***Problem:*** *Is there an interesting visualization research question in this problem?*
This question points to three possible pitfalls. First, the researcher might be faced with a problem that can be automated (*PF-6*). Second, the problem, or its solution, may not interest the researcher (*PF-7*). Or third, the problem requires engineering, not research, to solve (*PF-8*). In one of our projects, we identified this latter pitfall after several months of requirements analysis in the *discovery* stage. We provided the domain experts with a concise list of suggestions for an engineering solution to their problem, and both sides parted ways satisfied.

***Need:*** *Is there a real need or are existing approaches good enough?*
If current approaches are sufficient then domain experts are unlikely to go to the effort of changing their workflow to adopt a new tool, making validation of the benefits of a proposed design difficult to acquire (*PF-9*).

***Task:*** *Am I addressing a real task? How long will the need persist? How central is the task, and to how many people?*
It is risky to devote major resources to designing for a task of only peripheral relevance to the domain experts, especially if there are only a few of them. Full validation of the design's effectiveness will be difficult or impossible if they spend only a small fraction of their time performing the task, or if the task becomes moot before the design process is complete (*PF-10*). We encountered this pitfall with Constellation when the computational linguists moved on to other research questions, away from the task the tool was designed to support, before the implementation was complete. We were able to salvage the project by focusing on the contributions of our work in terms of the abstractions developed, the techniques proposed, and the lessons learned [52, 48]. We also brushed against this pitfall with MizBee when the first domain expert finished the targeted parts of her biological analysis before the tool was ready for use; finding a second domain expert who was just beginning that analysis phase, however, yielded strong validation results for the design study [45]. These examples also point to how a design study resulting in a tool aimed at a small group of domain experts can still lead to strong contributions. In this sense, the value of design studies differs from van Wijk's definition of the value of visualization which advocates for targeting larger numbers of users [88].

INTERPERSONAL CONSIDERATIONS: Interpersonal considerations, although easy to overlook, play an important role in the success or failure of a design study (*PF-11*). In anthropology and ethnography, the establishment of rapport between a researcher and study participants is seen as a core factor for successful field work [24, 63]. While this factor is less crucial in design studies, we have found that rapport and project success do go hand in hand.

#### 4.1.3 *Cast*: Identify Collaborator Roles

The awareness of different roles in collaborations is a common theme in other research areas: the user-centered design literature, for instance, distinguishes many user, stakeholder and researcher roles [7, 9, 38, 60], while the anthropology literature distinguishes key actors who connect researchers with other key people and key informants who researchers can easily learn from [6]. Informed by these ideas, we define roles that we repeatedly observed in design studies.

There are two critical roles in a design study collaboration. The **front-line analyst** is the domain expert end user doing the actual data analysis, and is the person who will use the new visualization tool. The **gatekeeper** is the person with the power to approve or block the project, including authorizing people to spend time on the project and release of the data. In an academic environment, the front-line analysts are often graduate students or postdocs, with the faculty member who is the principal investigator of a lab serving as the gatekeeper. While it is common to identify additional front-line analysts over the course of a project, starting a design study before contact is established with at least one front-line analyst and approval is obtained from the central gatekeeper is a major pitfall (*PF-12*).

We distinguish roles from people; that is, a single person might hold multiple roles at once. However, the distribution of roles to people can be different for different design studies—expecting them to be same for each project is another pitfall (*PF-13*). After several projects where the front-line analyst was also the gatekeeper, we were surprised by a situation where a gatekeeper objected to central parts of the validation in a design study paper extremely late in the publication process, despite the approval from several front-line analysts [46]. The situation was resolved to everyone's satisfaction by anonymizing the data, but sharper awareness of the split between these roles on our part would have led us to consult directly with the gatekeeper much earlier.

Several additional roles are useful, but not crucial, and thus do not need to be filled before starting a project. **Connectors** are people who connect the visualization researcher to other interesting people, usually front-line analysts. **Translators** are people who are very good in abstracting their domain problems into a more generic form, and relating them to larger-context domain goals. **Co-authors** are part of the paper writing process; often it is not obvious until the very end of the project which, if any, collaborators might be appropriate for this role.

We have identified one role that should be treated with great care: **fellow tool builders**. Fellow tool builders often want to augment a tool they have designed with visualization capabilities. They may not have had direct contact with front-line analysts themselves, however, and thus may not have correctly characterized the visualization needs. Mistaking fellow tool builders for front-line analysts is thus a pitfall (*PF-14*); it was also a major contributing factor in the PowerSetViewer failure case [54].

At its worst, this pitfall can cascade into triggering most of the other *winnow*-stage pitfalls. In one of our other failure cases, WikeVis [72], we prematurely committed to a collaboration with a fellow tool builder (PF-3, PF-12, PF-14). Excited about visualization, he promised to connect us "promptly" to front-line analysts with data. When we met the gatekeeper, however, we discovered that no real data was available yet (PF-4), and that we would not be allowed to meet with the extremely busy front-line analysts until we had a visualization tool ready for them to use (PF-5). We tried to rescue the project by immediately implementing a software prototype based on the vague task description of the fellow tool builder and using synthetic data we generated ourselves, skipping over our planned problem analysis (PF-1). The resulting ineffective prototype coupled with a continued unavailability of real data led us to pull out of the project after months of work.

## 4.2 Core Phase

The core of a design study contains four stages: *discover*, *design*, *implement*, and *deploy*.

### 4.2.1 *Discover*: Problem Characterization & Abstraction

In a design study it is essential to learn about the target domain and the practices, needs, problems, and requirements of the domain experts in order to discover if and how visualization can enable insight and discovery. It is a pitfall to focus only on the problematic parts of the workflow, while ignoring the successful aspects that work well (*PF-15*). RelEx is an example design study where these both sides of this coin were crucial in the visualization design [74].

The discovery stage is related to what is called *requirements analysis* in software engineering [36], which is directly linked to talking with and observing domain experts. The process of problem characterization and abstraction in a design study is iterative and cyclic: the expert speaks and the researcher listens, the researcher abstracts, then elicits feedback from the expert on the abstraction. While the refinement of the abstraction begins in the *discover* stage, it continues through all subsequent stages through the final *write* stage.

Problem characterization and abstraction are critical for design studies. In particular, a pithy abstraction supports the transferability of a design study's specific results and findings to other domains, and also allows for an understandable and straightforward description of a domain problem to a visualization audience. Abstraction should happen very early in the discovery stage and should frequently be validated by checking back with the expert to ensure correctness. One design study where we put significant effort into articulating the abstractions clearly and concisely was MizBee [45]; ABySS-Explorer is an excellent example from other authors [56].

One difficulty of the *discover* stage is that *just talking* to users is necessary but typically not sufficient. The phenomenon that what a target user says they do in retrospect is only an incomplete match with their actual activities is well-known in psychology [23] and HCI [60]; most users do not accurately introspect about their own data analysis and visualization needs, making these needs invisible to them. Thus, the standard practice in user-centered design is a combination of methods including interviews and observations [7, 60]. A common observation technique in ethnography is *fly-on-the-wall* [8], a passive method where the researcher silently and unobtrusively observes users with the goal of fading into the background to obtain an objective picture of normal activities. We tried this method in one of our projects [78] but found it ineffective in a design study context as the complex cognitive tasks addressed by design studies are difficult to silently observe [76]: in data analysis many things happen within the head of the user. While the methods of *just talking* and *fly-on-the-wall* provide some interesting information, expecting them to work alone is a pitfall (*PF-16*). We have found good results with contextual inquiries [28], where the researcher observes users working in their real-world context and interrupts to ask questions when clarification is needed [71, 73, 74, 77, 78]. Reading domain literature, whether as general background or based on specific suggestions from the experts, is an additional way to gather information.

In discussions with domain experts, another pitfall is allowing them to focus initial conversations on their vision of possible visualization solutions rather than on explaining their problems (*PF-17*). Researchers often need to actively push domain experts to discuss problems, not solutions, until the expert needs are well understood.

The ability to quickly and effectively characterize and abstract a problem is a key skill for design study researchers. We argue for learning *just enough* to abstract, rather than attempting to understand all details as in approaches such as ethnography [6] or grounded theory [14, 25]. Gathering a full-blown understanding of a target domain to the point of becoming an expert is problematic both because it is very time-consuming, taking years or even decades, and because it also carries the inherent danger that visualization needs will become invisible at some point. We argue that a sweet spot exists for how much domain knowledge to acquire—erring in either direction, with not enough or too much, will result in an ineffective design study (*PF-18*). Working with experts in many different domains allows a design study researcher to gain experience in identifying this sweet spot and hence become more effective at rapid problem characterization and abstraction. We note that this stance aligns well with newer theoretical approaches in HCI [64], and we will return to this debate in Section 5.

### 4.2.2 *Design*: Data Abstraction, Visual Encoding & Interaction

After reaching a shared understanding of a problem with domain experts in the *discover* phase, the visualization researcher can begin designing a visualization solution. Beginning this stage does not mean that changes to the problem characterization and task abstraction are finished; further refinements of understanding problems and tasks are almost inevitable as work continues through subsequent stages.

Our definition of *design* at this stage is the generation and validation of data abstractions, visual encodings, and interaction mechanisms. We include data abstraction as an active design component because many decisions made about the visualization design include transforming and deriving data; the task abstraction in not included because it is inherently about what the experts need to accomplish. A task abstraction will either be a good or a bad reflection of the actual domain, while a data abstraction proposed by the researcher will be appropriate or inappropriate for the specific problem at hand. An interesting example is the ABySS-Explorer project [56] where the computational algorithms that assemble a genome from sequence data rely on a graph with unique sequence strands represented as nodes and overlapping strands as edges. The visualization researchers observed, however, that the experts often used a dual representation (swapping nodes and edges) when sketching their ideas on paper, and found that building their visualization tool around this swapped representation allowed the domain scientists to reason about the output of the algorithms very effectively. In contrast, in one of our own projects the experts reacted with shock and horror to an early proposal to swap nodes and edges [74]. Examples like these point to the importance and difficulty of finding the right abstraction; too little attention to abstraction is a common pitfall (*PF-19*).

The basic design cycle, as articulated in fields such as industrial design, includes identifying requirements, generating multiple solutions, and selecting the best one [88]. The previous *discover* stage of our framework covered the identifying requirements step. This *design* stage covers the generation and selection parts of the cycle. A common pitfall is to consider too few solutions and to prematurely commit to a selected solution (*PF-20*).

To avoid this pitfall, we suggest that researchers strive to have a **broad consideration space** of possible solutions. The consideration space consists of the set of valid visual encodings, interaction mechanisms, and data abstractions. While the size of the consideration space may not be a central concern in technique-driven work, its breadth is critical for problem-driven research. After considering broadly, the researcher should iteratively filter the consideration space down to a **narrow proposal space** based on design principles and guidelines. The suggested solutions in the proposal space should be brought up for discussion with domain experts, for instance, in the form of paper mockups, data sketches, or low-level prototypes [42]. The feedback of the experts can then be used to filter the proposal space to a selection of one or several design solutions that will be implemented in depth.

The goal of the design cycle is satisfying rather than optimizing: while there is usually not a *best* solution, there are many *good* and *ok* ones. The problem of a small consideration space is the higher probability of only considering ok solutions and missing a good one. One way to ensure that more than one possibility is considered is to generate multiple ideas in parallel—our AutobahnVis design study paper discusses parallel idea generation in early design stages [78]. Work in HCI shows that "parallel prototyping leads to better design results, more divergence, and increased self-efficacy" [22].

An alluring pitfall for researchers accustomed to technique-driven work is to assume that a particular algorithm or technique that they developed in a past project is the right match for a new problem, instead of considering multiple ideas. Although such a match is not impossible, it is rare: researchers should be very careful not to assert a specious need for their favorite technique (*PF-21*). We have seen this pitfall many times as reviewers, and fell into it ourselves in one failed design study [51].

#### 4.2.3 *Implement*: Prototypes, Tool & Usability

The implementation of software prototypes and tools is tightly interleaved with the design process. Choosing the right algorithms to meet scalability and other requirements, closely integrating new software with existing workflows [76], and creating software prototypes are all instances of design activities that involve coding. Here, we summarize some HCI and software engineering guidelines which are helpful in the process of implementing design prototypes and solutions.

Rapid software prototyping, with the goal of quickly developing throw-away code, is a crucial skill in design studies: non-rapid prototyping is another pitfall (*PF-22*). In particular, the more time spent coding a solution the harder it is to throw it away. The tendency is to tweak a given implementation rather than to start over from scratch, which is problematic in cases where a design turns out not to fit the identified needs and problem of the experts, or where the needs have changed. This pitfall was a major factor the failure of the PowerSetViewer design study [54].

This pitfall is common knowledge in software engineering and is addressed by approaches like extreme programming and agile software development [41]. Several tactics for design studies are: start simply, ideally with paper prototypes; quickly write code that can be thrown away; and close user feedback loops with methods such as design interviews and workshops [15], or deploying early versions of a tool as technology probes [31]. We have used rapid prototyping in many of our design studies [45, 46, 44, 43, 52, 70, 71, 73, 74, 75, 77, 78].

Usability engineering is another approach that crosscuts design and implementation; a pitfall is to focus either too much or too little on usability (*PF-23*). Too little usability is still the more common case in visualization, where a tool that might provide utility to domain experts does not succeed because it is too difficult to use. Too much usability can be a pitfall for more HCI-oriented researchers, where a sole focus on usability can obscure more interesting questions about the usefulness and utility of a novel approach [26]. If the domain experts have limited time for engagement, usability inspection methods can augment the deployment of a visualization tool. These methods include cognitive walkthroughs, heuristic evaluations, and other kinds of expert reviews from people trained in HCI [85, 86], as well as discount usability testing with non-experts such as students [57]. We used discount usability with RelEx [74] and expert reviews in Vismon [10].

#### 4.2.4 *Deploy*: Release & Gather Feedback

The final stage of the core phase involves deploying a tool and gathering feedback about its use *in the wild*. This stage is a central component of a successful design study, but a common pitfall is to not build enough deployment time into a project schedule (*PF-24*). We consider several weeks to be a bare minimum, and several months to be much safer.

The major goal in validating a deployed system is to find out whether domain experts are indeed helped by the new solution. This goal is typically confirmed by experts doing tasks faster, more correctly, or with less workload, or by experts doing things they were not able to do before. A different contribution is changing the way experts see the problem. While this goal is less tangible, and perhaps less obvious, we argue in Section 3.2 for considering such a change as a step towards automating a problem. The two axes of task clarity and information location allow for a more concrete reasoning about such findings, namely changing task clarity from fuzzy to crisp, and changing the location of critical information from inside the expert's head to digital form.

The visualization literature contains a multitude of proposed methods for evaluating visualization tools in the wild [13, 33, 69, 85, 95], as well as guidance on using them [39, 50]. The most common form of validation are *case studies* with real users, real problems, and real data, as featured in many strong design studies by others [35, 58], and many of our own [10, 40, 43, 44, 45, 46, 71, 74, 76, 77]. A common pitfall is to claim a case study where the users, tasks, or data are not in fact real (*PF-25*); we have seen many examples as reviewers. The common case is where the data is real but the findings were made by the developers of the tool themselves, rather than a front-line analyst. While these examples are a legitimate form of validation, they are weaker and less convincing than reporting from the wild; we suggest that they be clearly labelled as *usage scenarios* so that there is no implication that they are true case studies. We have used them as validation in several of our own design studies [52, 53], and a mix of usage scenarios and case studies in others [4].

Lab studies with domain experts are rare in design studies because their findings tend to be less rich than field studies. Sometimes, however, they can provide benefits by convincing gatekeepers to open the door for wide deployment of visualization, as in the design study of heart disease diagnosis by Borkin et al. [11] or in our own MostVis project [73, 76].

From a collaboration point of view, researchers should be aware of *experimental demand characteristic* effects [12, 32]—when researchers work closely with domain experts it is very likely that the experts will offer positive and favorable feedback. While having experts give positive feedback about a tool is necessary for showing efficacy, it is far from sufficient for a convincing validation (*PF-26*).

### 4.3 Analysis Phase

The final two stages are *reflect* and *write*. These are usually done retrospectively but could be started early in the process.

#### 4.3.1 *Reflect*: Confirm, Refine, Reject, Propose Guidelines

The importance of reflection and its value for research is well-recognized in other fields such as anthropology [63]. This need for critical reflection is also evident in design study research; reflection is where research emerges from engineering and we identify failing to do so as a pitfall (*PF-27*). Reflection on how a specific design study relates to the larger research area of visualization is crucial for adding to the body of knowledge and allowing other researchers to benefit from the work. It is particularly informative for improving currently available design guidelines: based on new findings, previously proposed guidelines can be either *confirmed*, by substantiating further evidence of their usefulness; *refined* or extended with new insights; *rejected* when they are applied but do not work; or new guidelines might be *proposed*. For example, the reflection contribution of the LiveRAC design study refines guidelines for the user-centered design process [43].

#### 4.3.2 *Write*: Design Study Paper

Writing about a design study is often done in parallel with reflection, but can be started at any point [83]. In our experience, writing a design study paper is harder and more time-consuming than writing other types of visualization papers because of the amount of reconsideration and reorganization necessary. Leaving insufficient time for writing is a common pitfall (*PF-28*). We have found that a few weeks is usually not enough, and a few months is more realistic. Paper writing is the time to revisit abstractions, to identify contributions, and to come up with a coherent and understandable line of argumentation. A common pitfall is to think that a paper without a technique contribution is equal to a design study paper (*PF-29*), a mistake we have seen many times as reviewers. We strongly argue against this mentality: a good design study paper is not simply a weak technique paper dressed up with a bit of retroactive introspection.

We find that no matter how well we think the abstractions have been defined in the design phase, the writing phase inevitably forces us to revisit the abstractions in an attempt to clearly articulate them. A common reason for this is that many additional insights have emerged in the intervening stages. Thus, we argue that retrospectively refining the abstractions is the rule, and not the exception. The axes we provide in Section 3.2 can help to frame some of the less obvious contributions related to problem abstraction.

A common pitfall in writing a design study paper is to include too much domain background (*PF-30*). While the enthusiasm of the visualization researcher to explain hard-won knowledge about the domain to the readers is understandable, it is usually a better choice to put writing effort into presenting extremely clear abstractions of the task and data. Design study papers should include only the bare minimum of domain knowledge that is required to understand these abstractions. We have seen many examples of this pitfall as reviewers, and we continue to be reminded of it by reviewers of our own paper submissions. We fell headfirst into it ourselves in a very early design study, which would have been stronger if more space had been devoted to the rationale of geography as a proxy for network topology, and less to the intricacies of overlay network configuration and the travails of mapping IP addresses to geographic locations [53].

Another challenge is to construct an interesting and useful story from the set of events that constitute a design study. First, the researcher must re-articulate what was unfamiliar at the start of the process but has since become internalized and implicit. Moreover, the order of presentation and argumentation in a paper should follow a logical thread that is rarely tied to the actual chronology of events due to the iterative and cyclical nature of arriving at full understanding of the problem (*PF-31*). A careful selection of decisions made, and their justification, is imperative for narrating a compelling story about a design study and are worth discussing as part of the reflections on lessons learned. In this spirit, writing a design study paper has much in common with writing for qualitative research in the social sciences. In that literature, the process of writing is seen as an important research component of sense-making from observations gathered in field work, above and beyond merely being a reporting method [62, 93].

In technique-driven work, the goal of novelty means that there is a rush to publish as soon as possible. In problem-driven work, attempting to publish too soon is a common mistake, leading to a submission that is shallow and lacks depth (*PF-32*). We have fallen prey to this pitfall ourselves more than once. In one case, a design study was rejected upon first submission, and was only published after significantly more work was completed [10]; in retrospect, the original submission was premature. In another case, work that we now consider preliminary was accepted for publication [78]. After publication we made further refinements of the tool and validated the design with a field evaluation, but these improvements and findings did not warrant a full second paper. We included this work as a secondary contribution in a later paper about lessons learned across many projects [76], but in retrospect we should have waited to submit until later in the project life cycle.

It is rare that another group is pursuing exactly the same goal given the enormous number of possible data and task combinations. Typically a design requires several iterations before it is as effective as possible, and the first version of a system most often does not constitute a conclusive contribution. Similarly, reflecting on lessons learned from the specific situation of study in order to derive new or refined general guidelines typically requires an iterative process of thinking and writing. A challenge for researchers who are familiar with technique-driven work and who want to expand into embracing design studies is that the mental reflexes of these two modes of working are nearly opposite. We offer a metaphor that technique-driven work is like running a footrace, while problem-driven work is like preparing for a violin concert: deciding when to perform is part of the challenge and the primary hazard is halting before one's full potential is reached, as opposed to the challenge of reaching a defined finish line first.

Table 1. Summary of the 32 design study pitfalls that we identified.

| | | |
|---|---|---|
| PF-1 | premature advance: jumping forward over stages | general |
| PF-2 | premature start: insufficient knowledge of vis literature | learn |
| PF-3 | premature commitment: collaboration with wrong people | winnow |
| PF-4 | no real data available (yet) | winnow |
| PF-5 | insufficient time available from potential collaborators | winnow |
| PF-6 | no need for visualization: problem can be automated | winnow |
| PF-7 | researcher expertise does not match domain problem | winnow |
| PF-8 | no need for research: engineering vs. research project | winnow |
| PF-9 | no need for change: existing tools are good enough | winnow |
| PF-10 | no real/important/recurring task | winnow |
| PF-11 | no rapport with collaborators | winnow |
| PF-12 | not identifying front line analyst and gatekeeper before start | cast |
| PF-13 | assuming every project will have the same role distribution | cast |
| PF-14 | mistaking fellow tool builders for real end users | cast |
| PF-15 | ignoring practices that currently work well | discover |
| PF-16 | expecting *just talking* or *fly on wall* to work | discover |
| PF-17 | experts focusing on visualization design vs. domain problem | discover |
| PF-18 | learning their problems/language: too little / too much | discover |
| PF-19 | abstraction: too little | design |
| PF-20 | premature design commitment: consideration space too small | design |
| PF-21 | mistaking technique-driven for problem-driven work | design |
| PF-22 | non-rapid prototyping | implement |
| PF-23 | usability: too little / too much | implement |
| PF-24 | premature end: insufficient deploy time built into schedule | deploy |
| PF-25 | usage scenario not case study: non-real task/data/user | deploy |
| PF-26 | *liking* necessary but not sufficient for validation | deploy |
| PF-27 | failing to improve guidelines: confirm, refine, reject, propose | reflect |
| PF-28 | insufficient writing time built into schedule | write |
| PF-29 | no technique contribution $\neq$ good design study | write |
| PF-30 | too much domain background in paper | write |
| PF-31 | story told chronologically vs. focus on final results | write |
| PF-32 | premature end: win race vs. practice music for debut | write |

## 5 COMPARING METHODOLOGIES

Design studies involve a significant amount of qualitative field work; we now compare design study methodology to influential methodologies in HCI with similar qualitative intentions. We also use the terminology from these methodologies to buttress a key claim on how to judge design studies: **transferability is the goal, not reproducibility**.

*Ethnography* is perhaps the most widely discussed qualitative research methodology in HCI [16, 29, 30]. Traditional ethnography in the fields of anthropology [6] and sociology [81] aims at building a rich picture of a culture. The researcher is typically immersed for many months or even years to build up a detailed understanding of life and practice within the culture using methods that include observation and interview; shedding preconceived notions is a tactic for reaching this goal. Some of these methods have been adapted for use in HCI, however under a very different methodological umbrella. In these fields the goal is to distill findings into implications for design, requiring methods that quickly build an understanding of how a technology intervention might improve workflows. While some sternly critique this approach [20, 21], we are firmly in the camp of authors such as Rogers [64, 65] who argues that goal-directed fieldwork is appropriate when it is neither feasible nor desirable to capture everything, and Millen who advocates *rapid ethnography* [47]. This stand implies that our observations will be specific to visualization and likely will not be helpful in other fields; conversely, we assert that an observer without a visualization background will not get the answers needed for abstracting the gathered information into visualization-compatible concepts.

The methodology of *grounded theory* emphasizes building an understanding from the ground up based on careful and detailed analysis [14]. As with ethnography, we differ by advocating that valid progress can be made with considerably less analysis time. Although early proponents [87] cautioned against beginning the analysis process with preconceived notions, our insistence that visualization researchers must have a solid foundation in visualization knowledge aligns better with more recent interpretations [25] that advocate bringing a prepared mind to the project, a call echoed by others [63].

Many aspects of the *action research* (AR) methodology [27] align with design study methodology. First is the idea of learning through action, where intervention in the existing activities of the collaborative research partner is an explicit aim of the research agenda, and prolonged engagement is required. A second resonance is the identification of *transferability rather than reproducibility* as the desired outcome, as the aim is to create a solution for a specific problem. Indeed, our emphasis on abstraction can be cast as a way to "share sufficient knowledge about a solution that it may potentially be transferred to other contexts" [27]. The third key idea is that personal involvement of the researcher is central and desirable, rather than being a dismaying incursion of subjectivity that is a threat to validity; van Wijk makes the same argument for the visualization community [88]. A related idea from AR is the importance of the researcher being skilled at opening up lines of communication rather than being a distant observer.

However, our perspective opposes the AR agenda in a few crucial ways. First, we explicitly advocate translation of participant concepts into the language of visualization as part of the data and task abstraction, whereas AR opposes the layering of scientific language on participant concepts. Second, we do advocate that the visualization researcher takes the lead on designing the solution because they know about the visualization design space while the domain expert does not; this stance directly opposes the AR attitude that the researcher should facilitate but not lead. Third, we do not necessarily advocate that the participants be full partners in the writing process, whereas that is an important goal for AR. We also note that some concerns of AR are simply orthogonal to ours, including the political agenda of being socially relevant and adversarial to the status quo, and the theoretical underpinning of postmodernity.

One theme shared between design study methodology and all of these other qualitative approaches is the conviction that *reproducibility is not the goal* because the design study process requires intrinsically subjective field work [12]. The measure of success is not that a different visualization researcher would design the same system, it is that this particular researcher has created something useful.

## 6 Conclusions

In this paper we propose a process for conducting design studies, the nine-stage framework, formed by reflecting on our personal experiences as well as an extensive literature review in other fields. The framework consists of nine stages with practical guidance and potential pitfalls for each stage. Along with the nine-stage framework, we provide relevant definitions of key terms, and define the two axes of task clarity and information location in order to describe the suitability and potential contributions of design studies. We also contrast design study methodology with other methodologies popular in HCI that use qualitative methods.

We offer our process as **one** potential way of conducting design studies, which should not be taken as set in stone but as a starting point for a vivid and creative discussion about alternative approaches and ideas. Our process focuses on design studies as conducted by visualization researchers. Interesting questions rising from this focus include: how does the process generalize to practitioners? How does it differ when applied to related fields such as data mining or machine learning [89]? We hope that our work will entice more visualization researchers into this fast-growing part of the field, and that it will inspire further methodological discussion.

## Acknowledgements

We thank all collaborators and co-researchers from our various design studies. We also thank NSERC and the CRA for funding this work, and Matthew Brehmer, Jessica Dawson, Stephen Ingram, Kristin Potter, Elizabeth Jurrus, and Torsten Möller for their feedback.

## References

[1] M. Agrawala, W. Li, and F. Berthouzoz. Design principles for visual communication. *Communications of the ACM*, pages 60–69, Apr. 2011.

[2] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. *IEEE Symp. Information Visualization (InfoVis)*, pages 111–117, 2005.

[3] R. Amar and J. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 11(4):432–442, 2005.

[4] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008)*, 14(6):1253–60, 2008.

[5] D. Baur, F. Seiffert, M. Sedlmair, and S. Boring. The streams of our lives: Visualizing listening histories in context. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2010)*, 16(6):1119–1128, 2010.

[6] H. R. Bernard. *Research Methods in Anthropology: Qualitative and Quantitative Approaches*. AltaMira Press, 5th edition, 2011.

[7] H. Beyer and K. Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[8] J. Blomberg, J. Giacomi, A. Mosher, and P. Swenton-Wall. Ethnographic field methods and their relation to design. *Participatory design: Principles and practices*, pages 123–155, 1993.

[9] K. Bødker, F. Kensing, and J. Simonsen. *Participatory IT design: Designing for business and workplace realities*. The MIT Press, 2004.

[10] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum (Proc. EuroVis 2012)*, 2012. in press.

[11] M. A. Borkin, K. Z. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. J. Rybicki, C. L. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2011)*, pages 2479–2488, 2011.

[12] B. Brown, S. Reeves, and S. Sherwood. Into the wild: Challenges and opportunities for field trial methods. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1657–1666, 2011.

[13] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 19–45. Springer LNCS, 2008.

[14] K. Charmaz. *Constructing grounded theory. A practical guide through qualitative analysis*. Sage Publications, Inc, 2006.

[15] A. Cockburn. *Agile Software Development: The Cooperative Game*. Addison-Wesley Professional, 2006.

[16] A. Crabtree, T. Rodden, P. Tolmie, and G. Button. Ethnography considered harmful. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 879–888, 2009.

[17] P. Craig and J. Kennedy. Coordinated graph and scatter-plot views for the visual exploration of microarray time-series data. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 173–180, 2003.

[18] M. Crotty. *The foundations of social research: Meaning and perspective in the research process - Chapter 1*. Sage Publications Ltd, 1998.

[19] A. Dix. Theoretical analysis and theory creation. In *Research Methods in Human-Computer Interaction*, pages 175–195. Cambridge Univ. Press, 2008.

[20] P. Dourish. Implications for design. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 541–550, 2006.

[21] P. Dourish. Responsibilities and implications: further thoughts on ethnography and design. In *Proc. ACM Conf. Designing for User eXperiences (DUX)*, pages 2–16, 2007.

[22] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Computer-Human Interaction (ToCHI)*, 17(4):1–24, 2010.

[23] K. A. Ericsson and H. A. Simon. Verbal reports as data. *Psychological Review*, 87(3):215–251, 1980.

[24] D. M. Fetterman. *Ethnography: Step by Step*. Sage Publications, 2nd edition, 1998.

[25] D. Furniss, A. Blandford, and P. Curzon. Confessions from a Grounded Theory PhD: Experiences and Lessons Learnt. In *ACM Trans. Computer-Human Interaction (ToCHI)*, pages 113–122, 2011.

[26] S. Greenberg and B. Buxton. Usability evaluation considered harmful (some of the time). In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 111–120, 2008.

[27] G. R. Hayes. The relationship of action research to human-computer interaction. *ACM Trans. Computer-Human Interaction (ToCHI)*, 18(3):1–20, 2011.

[28] K. Holtzblatt and S. Jones. Contextual inquiry: A participatory technique for system design. In *D. Schuler, A. Namioka (eds.): Participatory Design: Principles and Practices*, pages 177–210. Lawrence Erlbaum Associates, 1993.

[29] J. Hughes, V. King, and T. Rodden. Moving out from the control room: ethnography in system design. *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW)*, pages 429–439, 1994.

[30] J. Hughes, J. O'Brien, and T. Rodden. Designing with ethnography: a presentation framework for design. In *Proc. Conf. Designing Interactive Systems (DIS)*, pages 147–158, 1997.

[31] H. Hutchinson et al. Technology probes: inspiring design for and with families. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 17–24, 2003.

[32] M. J. Intons-Peterson. Imagery paradigms: How vulnerable are they to experimenters' expectations? *Journ. Experimental Psychology - Human Perception and Performance*, 9:394–412, 1983.

[33] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded evaluation of information visualizations. In *Proc. ACM CHI Wkshp. BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. 2008.

[34] D. F. Keefe, M. Ewert, W. Ribarsky, and R. Chang. Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Trans. Visualization and Computer Graphics (Proc. Vis 2009)*, 15(6):1383–1390, 2009.

[35] R. Kincaid. SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2010)*, 16(6):900–907, 2010.

[36] B. L. Kovitz. *Practical Sofware Requirements: A Manual of Content & Style*. Manning, 1999.

[37] J. Kruger and D. Dunning. Unskilled and unaware of it: How difficulties in recog-

nizing one's own incompetence lead to inflated self-assessments. *Journ. Personality and Social Psychology*, 77(6):1121–1134, 1999.

[38] M. Kuniavsky. *Observing the User Experience: A Practitioner's Guide To User Research*. Morgan Kaufmann, 2003.

[39] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 2012. Appeared online: 30 Nov. 2011.

[40] H. Lam, D. Russell, D. Tang, and T. Munzner. Session Viewer: Visual Exploratory Analysis of Web Session Logs. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 147–154, 2007.

[41] C. Larman and V. R. Basili. Iterative and incremental development: A brief history. *IEEE Computer*, 36(6):47–56, 2003.

[42] D. Lloyd and J. Dykes. Human-centered approaches in geovisualization design: investigating multiple methods through a long-term case study. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2011)*, 17(12):498–507, 2011.

[43] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: interactive visual exploration of system management time-series data. In *Proc. ACM Conf. on Human Factors in Computing Systems (CHI)*, pages 1483–1492, 2008.

[44] M. Meyer, T. Munzner, A. dePace, and H. Pfister. MulteeSum: A tool for exploring space-time expression data. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2010)*, 16(6):908–917, 2010.

[45] M. Meyer, T. Munzner, and H. Pfister. MizBee: A multiscale synteny browser. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009)*, 15(6):897–904, 2009.

[46] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister. Pathline: A tool for comparative functional genomics. *Computer Graphics Forum (Proc. EuroVis 2010)*, 29(3):1043–1052, 2010.

[47] D. Millen. Rapid ethnography: Time deepening strategies for HCI field research. In *Proc. ACM Conf. Designing Interactive Systems (DIS)*, pages 280–286, 2000.

[48] T. Munzner. *Interactive Visualization of Large Graphs and Networks*. PhD thesis, Stanford University Department of Computer Science, 2000.

[49] T. Munzner. Process and pitfalls in writing information visualization research papers. In *Information Visualization: Human-Centered Issues and Perspectives*, pages 134–153. Springer LNCS 4950, 2008.

[50] T. Munzner. A Nested Model for Visualization Design and Validation. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009)*, 15(6):921–928, 2009.

[51] T. Munzner, A. Barsky, and M. Williams. Reflections on QuestVis: A visualization system for an environmental sustainability model. In *Scientific Visualization: Interactions, Features, Metaphors*, volume 2, pages 240–259. Dagstuhl Follow-Ups, 2011.

[52] T. Munzner, F. Guimbretière, and G. Robertson. Constellation: a visualization tool for linguistic queries from MindNet. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 132–135, 1999.

[53] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. Visualizing the Global Topology of the MBone. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 85–92, 1996.

[54] T. Munzner, Q. Kong, R. T. Ng, J. Lee, J. Klawe, D. Radulovic, and C. K. Leung. Visual mining of power sets with large alphabets. Technical report, Univ. British Columbia Computer Science TR-2005-25, Dec. 2005.

[55] T. Munzner and S. North. InfoVis 2003 Call for Participation: Papers. `http://www.infovis.org/infovis/2003/CFP/#papers`.

[56] C. B. Nielsen, S. D. Jackman, I. Birol, and S. J. Jones. ABySS-Explorer: Visualizing genome sequence assemblies. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009)*, 15(6):881–8, 2009.

[57] J. Nielsen. Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. In *Cost-Justifying Usability*, pages 245–272. Academic Press, 1994.

[58] A. Perer, I. Guy, E. Uziel, I. Ronen, and M. Jacovi. Visual Social Network Analytics for Relationship Discovery in the Enterprise. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 71–79, 2011.

[59] C. Plaisant. The challenge of information visualization evaluation. In *Proc. Advanced Visual Interfaces (AVI)*, pages 109–116. ACM Press, 2004.

[60] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2002.

[61] A. J. Pretorius and J. J. van Wijk. What does the user want to see? what do the data want to be? *Information Visualization*, 8(3):153–166, 2009.

[62] L. Richardson and E. A. S. Pierre. Writing: A method for inquiry. In *Handbook of Qualitative Research*, pages 959–978. Sage, 3rd edition, 2005.

[63] J. Rode. Reflexivity in Digital Anthropology. In *ACM Trans. Computer-Human Interaction (ToCHI)*, pages 123–132, 2011.

[64] Y. Rogers. New theoretical approaches for HCI. *Annual review of information science and technology*, 38(1):87–143, 2004.

[65] Y. Rogers, K. Connelly, L. Tedesco, W. Hazlewood, A. Kurtz, R. Hall, J. Hursey, and T. Toscos. Why it's worth the hassle: the value of in-situ studies when designing Ubicomp. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)*, pages 336–353. Springer-Verlag, 2007.

[66] B. E. Rogowitz, L. A. Treinish, and S. Bryson. How not to lie with visualization. *Comput. Phys.*, 10(3):268–273, June 1996.

[67] K. Ruhland, M. Sedlmair, S. Bioletti, and C. O'Sullivan. LibViz: Data Visualisation of the Old Library. *Intl. Journ. Architectural Computing*, 7(1):177–192, 2009.

[68] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. Visualization and Computer Graphics*, 11(4):443–456, 2005.

[69] P. Saraiya, C. North, V. Lam, and K. Duca. An insight-based longitudinal study of visual analytics. *IEEE Trans. Visualization and Computer Graphics*, 12(6):1511–1522, 2006.

[70] M. Sedlmair. Car-x-ray: Closely Integrating Both Worlds. In *Visual Analysis of In-Car Communication Networks (PhD Thesis)*. LMU, 2010.

[71] M. Sedlmair. ProgSpy2010: Visualizing Flash Activities. In *Visual Analysis of In-Car Communication Networks (PhD Thesis)*. LMU, 2010.

[72] M. Sedlmair. WiKeVis: Visualizing Functional Networks. In *Visual Analysis of In-Car Communication Networks (PhD Thesis)*. LMU, 2010.

[73] M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, and A. Butz. MostVis: An Interactive Visualization Supporting Automotive Engineers in MOST Catalog Exploration. In *Proc. Intl. Conf. Information Visualization (IV)*, pages 173–182, 2009.

[74] M. Sedlmair, A. Frank, T. Munzner, and A. Butz. RelEx: A design study of visual network analysis in the automotive domain. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2012)*, 2012. In press.

[75] M. Sedlmair, W. Hintermaier, K. Stocker, T. Böring, and A. Butz. A dual-view visualization of in-car communication processes. In *Proc. Intl. Conf. Information Visualization (IV)*, pages 157–162, 2008.

[76] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information Visualization Evaluation in Large Companies: Challenges, Experiences and Recommendations. *Information Visualization*, 10(3):248–266, 2011.

[77] M. Sedlmair, P. Isenberg, D. Baur, M. Mauerer, C. Pigorsch, and A. Butz. Cardiogram: Visual Analytics for Automotive Engineers. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1727–1736, 2011.

[78] M. Sedlmair, B. Kunze, W. Hintermaier, and A. Butz. User-centered development of a visual exploration system for in-car communication. In *Proc. Intl. Symp. Smart Graphics (SG)*, pages 105–116, 2009.

[79] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages (VL)*, pages 336–343. IEEE, 1996.

[80] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proc. AVI Wkshp. on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)*. ACM, 2006.

[81] D. Silverman. *A Very Short, Fairly Interesting and Reasonably Cheap Book About Qualitative Research*. Sage Publications Ltd., 2007.

[82] A. Slingsby, J. Dykes, and J. Wood. Configuring hierarchical layouts to address research questions. *IEEE Trans. Vis. Comput. Graph.*, 15(6):977–984, 2009.

[83] H. Thimbleby. Write now! In *Research Methods in Human-Computer Interaction*, pages 196–211. Cambridge Univ. Press, 2008.

[84] J. Thomas and K. Cook. Illuminating the path: The research and development agenda for visual analytics. *National Visualization and Analytics Center*, 2005.

[85] M. Tory and T. Möller. Human factors in visualization research. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 10(1):72–84, 2004.

[86] M. Tory and T. Möller. Evaluating visualizations: do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, 2005.

[87] J. van Niekerk and J. Roode. Glaserian and Straussian grounded theory: Similar or completely different? In *Proc. South African Institute of Computer Scientists and Information Technologists (SAICSIT)*, pages 96–103, 2009.

[88] J. J. van Wijk. Views on visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 12(4):421–432, 2006.

[89] K. L. Wagstaff. Machine Learning that Matters. In *Proceedings of the 29th International Conference on Machine Learning*. Copyright 2012 California Institute of Technology, June 2012.

[90] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. ACM Advanced Visual Interfaces (AVI)*, pages 110–119, 2000.

[91] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.

[92] W. Willinger, D. Alderson, and J. C. Doyle. Mathematics and the Internet: A source of enormous confusion and great potential. *Notices of the American Mathemematical Society*, 56(5):586–599, 2009.

[93] H. Wolcott. *Writing up qualitative research*, volume 20 of *Qualitative Research Methods*. Sage, 3rd edition, 2009.

[94] J. Wood, D. Badawood, J. Dykes, and A. Slingsby. BallotMaps: Detecting name bias in alphabetically ordered ballot papers. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2011)*, 17(12):2384–2391, 2011.

[95] T. Zuk, L. Schlesier, P. Neumann, M. S. Hancock, and S. Carpendale. Heuristics for information visualization evaluation. In *Proc. ACM CHI Wkshp. on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. 2008.