# Effects-Driven IT Development

Specifying, Realizing, and Assessing Usage Effects

Hertzum, Morten; Simonsen, Jesper

*Published in:*
Scandinavian Journal of Information Systems

*Publication date:*
2011

*Document Version*
Publisher's PDF, also known as Version of record

*Citation for published version (APA):*
Hertzum, M., & Simonsen, J. (2011). Effects-Driven IT Development: Specifying, Realizing, and Assessing Usage Effects. *Scandinavian Journal of Information Systems*, *23*(1), 3-28.

# Effects-Driven IT Development

## Specifying, realizing, and assessing usage effect

Morten Hertzum
User-Driven IT Innovation, Department of Communication, Business and Information Technologies, Roskilde University, Denmark
*mhz@ruc.dk*

Jesper Simonsen
User-Driven IT Innovation, Department of Communication, Business and Information Technologies, Roskilde University, Denmark
*simonsen@ruc.dk*

**Abstract.** For customers information technology (IT) is a means to an end. This tight association between IT systems and their use is, however, often absent during their development and implementation, resulting in systems that may fail to produce desired ends. Effects-driven IT development aims to avoid the chasm between development and implementation through a sustained focus on the effects to be achieved by users through their adoption and use of a system. This involves iteratively (a) specifying the purpose of the system in terms of effects, (b) developing an IT system and associated organizational change that realize the specified effects, and (c) measuring the absence or presence of the specified effects during pilot use of the system while also remaining alert to the emergence of beneficial but hitherto unspecified effects. In this paper we explore effects-driven IT development and discuss the possibilities and challenges involved in making it an instrument for managing IT projects. Two main challenges are that effects must be measured while development is still ongoing, making pilot implementations a central activity, and that vendor and customer must extend their collaboration, particularly concerning organizational implementation. Empirically, the paper is based on three cases from the Danish healthcare sector.

Keywords: effects-driven IT development, usage effects, pilot implementation, real-use evaluation

Accepting editor: Margunn Aanestad

# 1 Introduction

Approaches to the management of information technology (IT) projects tend to focus on the challenges of completing the technical development on time and within budget and to pay comparatively less attention to the risk that people will not adopt and use the IT systems and associated work practices (Alter 2001; Markus 2004). Conversely, approaches to organizational-change management tend to focus exclusively on an organization's adoption, assimilation, and use of new IT systems and work practices, thereby taking the systems and their technical development as given (Alter 2001; Markus 2004). The resulting chasm between technical development and organizational implementation is a problem for three interrelated reasons: First, important opportunities and consequences of a system do not emerge until the system is in real use. Experiences from using a system in its intended environment create a deeper understanding of how the system can support the users in their work and, thereby, lead to new and revised requirements, which could not be foreseen up front or upon seeing an early prototype (e.g., Boehm 2000; Orlikowski 1996). Second, the process of starting to use a system often results in merely partial adoption (e.g., Fichman and Kemerer 1999; Häkkinen and Hilmola 2008). Users may lack the information, motivation, or time to embrace a new system and the accompanying new ways of working, or they may contest that the new system is an improvement over existing practices. Third, the actual usage of IT systems appears to be central to the effect—in particular, the long-term effect—of IT investment on organizational performance (e.g., Brynjolfsson and Hitt 2000; Devaraj and Kohli 2003). A development-implementation chasm therefore entails a risk of bad system-organization fit and partial adoption of systems, which consequently do not produce intended benefits (Landauer 1995; Markus 2004; Ward and Daniel 2006).

Effects-driven IT development is our attempt to address the development-implementation chasm and work systematically toward capturing the benefits of new IT systems. Effects-driven IT development entails a sustained focus on the effects to be achieved by users through their adoption and use of a system. The overall idea is that specification and assessment of the effects desired from a system can provide customer and vendor with an instrument for managing IT projects. This idea resembles benefits management (Ward and Daniel 2006), which starts from the premise that most IT projects are organizational-change projects that involve IT. By initially specifying investment objectives and then refining these objectives into benefits, changes, and finally IT functionality, benefits management links IT systems with organizational change. Benefits management has gained recognition as a way of maintaining a focus on deriving business benefit from IT projects. Effects-driven IT development pursues the similar aim of enhancing extant systems-development methods with improved support for succeeding in the combined activity of development and implementation. The main difference between effects-driven IT development and benefits management is twofold. First, effects-driven IT development is an inherently iterative approach in which assessment of whether effects are achieved during pilot use is fed back to the ongoing development activities; that is, effects assessment is formative. In contrast, benefits management postpones assessment until after project completion and mainly sees the assessment as input for future projects (Ward and Daniel 2006, pp. 113-118); that is, benefits assessment is mainly summative. Second, effects-driven IT development involves measurement of whether the specified effects are achieved—combined with assessment of whether additional desirable effects are emerging. In contrast, benefits management relies mostly on review meetings and other informal means of assessment (Ward and Daniel 2006, pp. 264-268). Whereas benefits management focuses mainly on benefits specification, pays less attention to their assessment, and is quite vague about what happens after assessment, we see effects as an instrument for iterating back and forth across the development-implementation chasm in a managed way. We pursue effects-driven IT development to supplement approaches such as benefits management with an instrument for managing IT projects on the basis of iterative measurements of the absence or presence of usage effects.

The aim of this article is to explore effects-driven IT development and discuss the possibilities and challenges involved in making it an instrument for managing IT projects. We will do this by relating effects-driven IT development to other techniques in systems development and by drawing on our experiences from three empirical cases of effects-driven IT development in the Danish healthcare sector, conducted in the period of 2004-2009. In the following, we first describe effects-driven IT development, which is an iterative process with three elements: effects specification, effects realization, and effects assessment (Section 2). In describing the three elements we also discuss related work. To illustrate and refine effects-driven IT development we, then, report from three empirical cases that focus on the specification of effects, on assessing planned and emergent effects, and on working with effects that initially slipped (Section 3). Finally, we discuss our experiences from the empirical cases and the possibilities and challenges we see for effects-driven IT development (Section 4). An important challenge concerns how the availability of information about whether intended effects are actually achieved may transform the customer-vendor relationship.

# 2 Effects-driven IT development

Effects-driven IT development is an instrument for managing IT projects. It aims to integrate technical development with organizational implementation through a focus on the effects of a system on organizational work. Simply put, the overall idea is to capture the purpose of a system in terms of effects that are both measurable and meaningful to the customer, to measure whether these effects are attained during pilot use of the system, and to use these measurements as guidance for subsequent development and implementation activities. Such a sustained focus on effects accentuates that the functionality of a system is merely a means to an end and that the progress of an IT project should therefore be judged by the outcome of actual use of the functionality. Figure 1, to be elaborated below, summarizes the overall process of effects-driven IT development.

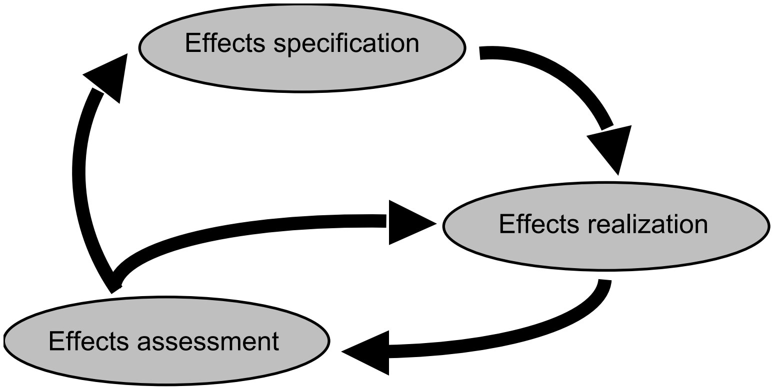


Figure 1. Effects-driven IT development.

## 2.1 Effects specification

Effects are the result of changes. Apart from the benefit associated with an effect, there are costs involved in accomplishing an effect because the required change depends on equipment purchases, IT development, revisions of work practices, new staff competences, or other organizational efforts. As different stakeholder groups have different tasks and responsibilities, any assessment of the cost-benefit relation associated with a change, such as the introduction of an IT system, may be specific to stakeholder groups. This is amplified by the way in which the change redistributes tasks, opportunities, power, and status among stakeholder groups (and individual staff members). Thus, the effects desired by different stakeholder groups will likely differ, and may be in conflict. In addition, there is a risk that the specified effects are not achieved because the changes are not consistently implemented or turn out to have unanticipated outcomes that annul the effects or render them unbeneficial. This suggests the presence of implementation barriers, including a lack of time, knowledge, and training, disproval of the IT system resulting in counter-implementation strategies, and a bad system-organization fit. It also shows the need for measures that make it possible to ascertain whether the specified effects have been achieved or follow-up activities are called for. Thus, apart from the effects as such there is a need for identifying their costs, stakeholders, barriers, and measures.

Effects-driven IT development assumes that it is possible to specify effects, not just in principle but also in practice. This involves identifying, formulating, and prioritizing effects. We suggest that this is done in collaboration with users following a participatory-design approach (e.g., Bødker et al. 2004). To support this process it appears useful to think of the effects as forming a hierarchy, see Figure 2. The effects hierarchy distinguishes five levels: *environment*, which includes legislation, political demands, standards, market considerations, competitors and so forth; *business strategy*, which states the organization-wide strategy adopted by an organization to attain good performance; *work tasks*, by which staff collaboratively organize their work to achieve the business strategy; *work processes*, through which work tasks are concretely attained with the available tools such as IT systems; and *IT systems*, which support staff in accomplishing work processes. Effects at higher levels specify why effects at lower levels are desirable, and lower-level effects specify how effects at higher levels can be attained. For example, national healthcare policies may state general, agenda-setting effects, which influence individual hospitals' choice of strategic effects, which in turn are reflected in effects directly concerning the clinical work. Figure 2 provides such an example and additional detail for the work-task level.

The effects hierarchy is a tool for working with the interrelations between effects. The presence of a higher-level effect indicates a need for specifying effects that spell out this effect at the lower levels. Conversely, it should be possible to link a lower-level effect to an effect at a higher level; otherwise the lower-level effect is probably superfluous. In addition, some of the specified effects may be hard to reconcile and may, therefore, require prioritization. The alignment of effects is central to effects specification and particularly important for systems with multiple stakeholders performing tightly coupled tasks. Typically, most of the time will be spent on the effects at the levels of work tasks, work processes, and IT systems because these effects directly affect the users' work. Effects at the strategic level are more indirect; these effects mostly set the context. A supplementary reason for focusing primarily on the effects that directly affect the users in their work is that system success is critically dependent on the users' support of and attitude toward the system and, thereby, on whether they can relate the sought-for effects to their work.

| *Effect hierarchy* | |
|---|---|
| *Environment* | The right medication is given to the right patient at the right time |
| *Business strategy* | All information about medication is recorded in the EMR |
| *Work tasks* | The nurses record their dispensing and administration of medication in the EMR |
| *Work processes* | Delegated medication orders replace recording in the nursing kardex |
| *IT systems* | EMR facility enabling nurses to make delegated medication orders |

➔

| *Effects details* | |
|---|---|
| *Description* | The nurses record their dispensing and administration of medication in the EMR |
| *Stakeholder* | Nurses |
| *Measure* | Number of violations of the requirement that all medication information is recorded in the EMR, established by record audits |
| *Target level* | Zero violations for delegated medication |
| *Present level* | Six (at second audit) |
| *Barriers* | Many activities compete for the nurses' attention. User interface for making delegated medication orders is somewhat unintuitive |

Figure 2. The specification of effects by means of an effects hierarchy and details about the individual effects. For more information about the example (in italics), see Section 3.3.

The approach most similar to effects-driven IT development is benefits management (Ward and Daniel 2006; Ward and Elvin 1999; Ward et al. 1996), which is about managing IT-enabled business change and thereby getting value from IT investments. Ward and Daniel (2006) define a benefit as "an advantage on behalf of a particular stakeholder or group of stakeholders" (p. 107) and benefits management as "the process of organizing and managing such that the potential benefits arising from the use of IS/IT are actually realized" (p. 36). The first step in their five-step process model of benefits management is identifying and structuring benefits. This step resembles effects specification and involves identifying potential benefits, understanding how they can be realized through a combination of technology and organizational change, determining how the benefits can be measured, identifying barriers to benefits realization, and producing a business case for the project. In effects-driven IT development the effects tend to serve two different functions depending on their level in the effects hierarchy: ensuring alignment across effects and thereby among stakeholders (the environment and business-strategy levels) and enabling measurement (the three lower levels). With respect to the effects that are to be measured, Ward and Daniel (2006, pp. 172-188) distinguish four degrees of explicitness with which benefits can be measured: (1) *observable*, where agreed criteria exist for specific people to assess the benefit based on their experience or judgement, (2) *measurable*, where current performance can be measured but no performance targets exist, (3) *quantifiable*, where targets can be set for the improvement to be achieved, and (4) *financial*, where a formula exists to convert the improvement to a financial value. It is our experience that the desired level of explicitness tends to depend on the stakeholder group as well as on the level of the effect in the effects hierarchy. Financial effects are most common at the upper levels.

The effects hierarchy is inspired by the abstraction hierarchy of cognitive systems engineering (Rasmussen et al. 1994; Vicente 1999) and by the strategic analysis of Bødker et al. (2004). Like a number of other methods for work analysis the abstraction hierarchy distinguishes different levels of analysis. The five levels of the abstraction hierarchy are interrelated in the sense that any level can be considered a function (what), a goal (why) for a function at a lower level, and a means (how) for a function at a higher level. During an IT project, designers repeatedly move up and down in the abstraction hierarchy to clarify the whys, whats, and hows associated with the system. Vicente (1999) holds that the purposes at the top level of the abstraction hierarchy are relatively permanent properties of work domains and thereby provide a stable framework within which different designs can be explored. This has, for example, been exploited by Leveson (2000) in her intent specifications. It is our contention that specified effects, particularly at the upper levels of the effects hierarchy, tend to be quite stable but also that they do not exhaust the set of desired effects. That is, additional effects will likely emerge.

## 2.2 Effects realization

Effects realization involves the technical development of an IT system and its organizational implementation. Numerous methods exist for performing these two activities; for example, Crystal (Cockburn 2007) and Business Process Improvement (Harrington 1991). Effects-driven IT development is not a substitute for these methods. Rather, effects-driven IT development aims to enhance such methods with an instrument for managing the activities required to integrate technical development and organizational implementation. To make use of this enhancement and, thereby, maintain a sustained focus on effects, methods for technical development and organizational implementation must allow for pilot implementations, which are central to effects assessment.

A pilot implementation is a limited implementation of a system under development in its intended environment and using real data (Hertzum et al. 2011). One or a few sites are selected for the pilot implementation, and the experiences gained at these sites inform the subsequent development and implementation activities. That is, pilot implementations are conducted while a system is being developed, not after it has been completed. We contend that this can be accomplished by, for example, configuring systems based on flexible development platforms and using Wizard-of-Oz techniques to simulate system facilities that have not yet been developed (Maulsby et al. 1993). A pilot implementation should be long enough for the effects of the use of the system to materialize, but at the same time a pilot implementation must be short enough to fit within an IT project. This tension may necessitate brief pilot implementations in some situations. While a system that is being pilot implemented is by definition not fully developed, it must—in contrast to a prototype—have considerable functionality and be sufficiently robust, reliable, and efficient to enable its implementation and use in a real work environment. This makes pilot implementations challenging to conduct.

Pilot implementations are central to effects-driven IT development because they cross the chasm between development and implementation. In contrast, IT projects appear frequently to focus on technical development and under-recognize the importance and complexity of organizational implementation. For example, Alter (2001, p. 27) concludes about the life-cycle model that represents "a typical project that might confront a software development manager" that "this model ends before implementation in the organization begins". However, unless both technical development and organizational implementation succeed customers will not reap the intended effects from their investment. This is emphasized in benefits management and in Markus' (2004) technochange model, which embeds IT projects in an overarching effort of organizational change. In the technochange model the technical development of an IT system is followed by two additional stages: shakedown and benefits capture. During shakedown the organization starts operating in a new way with the system and troubleshoots the associated problems; during benefits capture the organization systematically derives benefit from the new way of working. While technical development may produce a high-quality system, customers do not benefit until after successful adoption of the system in their organization. Thus, unless shakedown and benefits capture are included in IT projects there is a risk that shakedown will be unsystematic or partial and, consequently, that customers will not achieve the effects that motivated their decision to invest in a new system.

While the changes performed during shakedown are directed toward achieving the specified effects, it is well-recognized that the introduction of a new IT system may also produce unanticipated changes. This creates a need for improvisational change management, which in effects-driven IT development is addressed through the integrated approach to development and implementation and through the use of pilot implementation. Orlikowski and Hofman (1997) characterize improvisational change management by distinguishing three kinds of change: *Anticipated change* is planned ahead and occurs as intended by the originators of the change. *Emergent change* is defined as local and spontaneous changes, not originally anticipated or intended. Such changes do not involve deliberate action but grow out of practice. *Opportunity-based changes* are purposefully introduced changes resulting from unexpected opportunities, events, or breakdowns that arise after the introduction of a new IT system. Opportunity-based change cannot influence the functionality of an IT system unless the IT project contains iteration from organizational implementation back to technical development. In the absence of such iteration opportunity-based change is restricted to organizational implementation. Surprisingly, a systematic approach to improvisational change management is uncommon in practice. In a survey of large IT projects, 86% of respondents agreed that it is impossible to predict all the benefits of a system in advance, yet only 19% of the respondents' organizations had any process in place to identify further benefits after the system had been deployed (Ward et al. 1996). Effects-driven IT development aims to provide such a process through pilot implementation and the associated assessment and possible revision of effects.

## 2.3 Effects assessment

Effects-driven IT development presupposes that it is feasible to use the presence or absence of effects as an active means of managing IT projects. For this to work it must be possible to demonstrate effects within the timeframe of IT projects. This involves setting up and conducting pilot implementations to assess whether specified effects are achieved and allow emergent effects to surface. Effects at the upper levels of the effects hierarchy will often take longer to materialize than those at the middle and lower levels, and it may be difficult to measure upper-level effects within the timeframe of an IT project. It may, instead, be possible to regard the accomplishment of middle-level effects as indicators that the upper-level effects are well on their way. In addition to measurement of at least some of the specified effects, effects assessment is an opportunity to evaluate the completeness of the set of specified effects and should involve an awareness toward new, hitherto unspecified effects.

Effects assessment is essential because effects specification and realization are imperfect processes. Effects specification will likely lead to the identification of only a subset of the ultimately desirable effects. Some of the unspecified effects may emerge during effects realization, but it is also likely that some specified effects will not be realized (slipped effects) and that some unspecified effects will remain absent (missed effects). Finally, effects may, during effects assessment, turn out to be either beneficial or adverse. The resulting effects taxonomy distinguishes eight types of effect, see Figure 3. Two of the effect types (*planned* and *emergent*) are realized and beneficial. These effects should be sustained during subsequent project activities, and for emergent effects this may involve some opportunity-based activity. Two other effect types (*adverse* and *adverse emergent*) are also realized, but adverse. These effects should be undone. While specified effects may infrequently turn out to be adverse, adverse emergent effects are not uncommon (e.g., Ash et al. 2004; Han et al. 2005; Scott et al. 2005). Two effect types (*slipped* and *missed*) are beneficial but unrealized. These effects are unrealized for different reasons and, therefore, call for different follow-up activities. Slipped effects require further work on effects realization, and because these effects were specified users will be aware of the need for further work. Missed effects require renewed work on effects specification, but because the users have hitherto been unaware of these effects they may remain missed opportunities. Finally, two effect types (both labelled *avoided adverse*) are neither realized nor beneficial. These (non) effects require no further action, except the removal of any such effects from the set of specified effects.

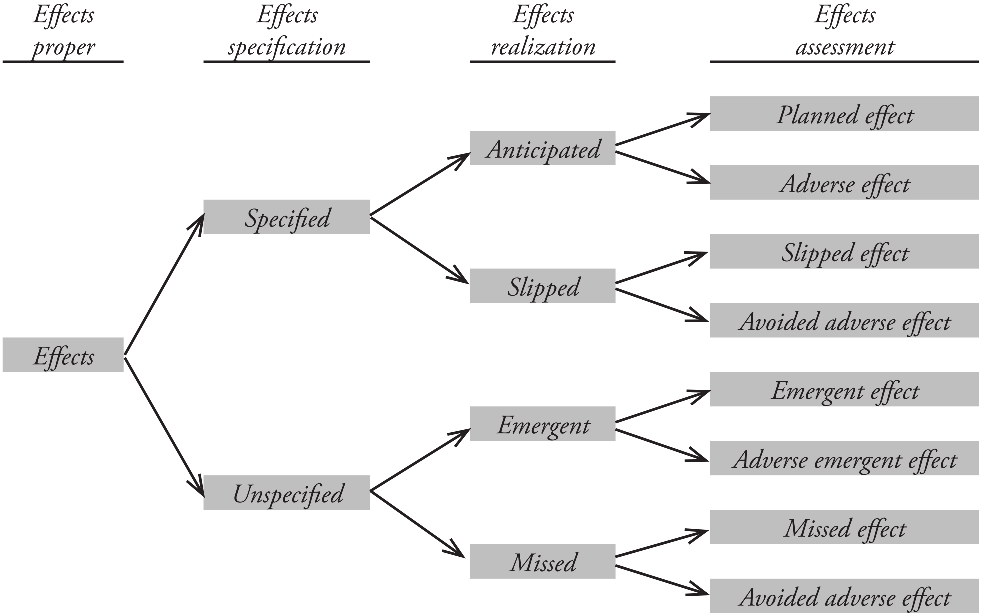


Figure 3. Effects taxonomy

Ward and Daniel (2006, p. 107) state that "All business performance improvements are measurable in some way and so are all of the benefits delivered by information systems." Reliable measures may however be difficult to obtain, particularly for effects that involve the establishment of new organizational procedures and collaborative practices (Farbey et al. 1999; Hamilton and Chervany 1981). Thus, while effects assessment is possible in principle, it may be challenging in practice. The challenges are emphasized by Love et al. (2005), who find that evaluation of IT systems is fraught with misconception and suffers from a lack of appropriate evaluation methods. A technique related to effects assessment is usability specifications (Good et al. 1986; Whiteside et al. 1988). A usability specification defines the worst, planned, and best levels of user performance for a specified set of tasks and reports a measurement of the users' present level of performance. This provides for a process alternating between realization and assessment until the planned level of performance has been attained. For the somewhat narrowly scoped tasks mostly associated with usability specifications it has not been considered a problem to obtain precise performance measurements. For effects that are to result from IT systems accompanied by substantial organizational change it is, however, difficult to determine the contribution of the IT system to the effects (Hamilton and Chervany 1981). Following Markus (2004) we consider attempts at linking effects to either technological or organizational causes dubious. Effects-driven IT development insists on the primacy of the effects and thereby on ensuring that IT projects do not become dissociated from the process of organizationally implementing the systems. Thus, effects assessment is an evaluation of the IT system with its associated organizational changes.

While effects specification, realization, and assessment individually resemble other research, their *integration into an instrument for managing IT projects* is unique to effects-driven IT development. For example, benefits management (Ward and Daniel 2006; Ward et al. 1996) defines assessment as a review performed at the end of a project mainly to inform future investments and projects, rather than as a means for guiding a project as it evolves. The abstraction hierarchy (Rasmussen et al. 1994; Vicente 1999) is a device for work analysis and not coupled to measurement of whether specified purposes are achieved after development and implementation. Prototyping (Beynon-Davies et al. 1999) tends to introduce iteration within the technical-development stage of IT projects, precluding the assessment of effects, which result from real work and thereby require (pilot) implementation. Conversely, results-driven incrementalism (Fichman and Moses 1999) and post-development pilot studies (Glass 1997) bypass technical development in favour of a focus on organisational implementation, precluding that effects assessment informs technical development. Agile systems development (Cockburn 2007; Dybå and Dingsøyr 2008) focuses on the technical-development stage and involves feedback from organizational implementation to technical development as part of an incremental approach to systems development. While this allows for the incorporation of emergent and opportunity-based change, agile methods do not normally involve the specification and assessment of effects as a means of systematically evaluating whether a system provides desired effects. Finally, our work on effects-driven IT development has been inspired by performance-based procurement (Connell et al. 1995), though it is an approach to procurement, rather than systems development, and focuses exclusively on financial benefits.

# 3 Three empirical cases

To illustrate effects-driven IT development we provide three empirical cases from our work with gradually devising and trying it out. The cases concern information systems for different parts of the healthcare sector. The healthcare sector is organizationally complex, healthcare information systems must be robust to avoid risk to patient health, and many efforts to introduce such systems have failed in the past (Haux et al. 2004; LeRouge et al. 2007; Sobol et al. 1999). We, therefore, consider the healthcare sector an appropriate test bed for effects-driven IT development. The first case focuses on the specification of effects, the second case on assessing planned and emergent effects, and the third case on working with slipped effects. Collectively, the cases cover most of our effects taxonomy and illustrate main aspects of effects-driven IT development.

## 3.1 Specifying effects: workspace system for healthcare centres

Municipal healthcare centres were established in Denmark in 2007 with a special focus on chronic and lifestyle diseases such as chronic obstructive pulmonary disease, diabetes, obesity, and certain forms of cancer. As part of the establishment of the centres a healthcare centre workspace system (HCWS) was developed to support the healthcare-centre clinicians, who comprise nurses and therapists, in devising, documenting, and keeping track of their work with their patients. The specification of the HCWS took place in collaboration with selected clinicians and consisted of specifying the effects to be achieved by the clinicians when the HCWS supported them in their work. We investigated the effects specification, which spanned a period of four months, by means of an action-research study in three healthcare centres (Barlach and Simonsen 2008).

Overall, the effects-specification process was organized as a series of five workshops in each of the three healthcare centres. At the first workshop in a series, the customer representatives were introduced to effects as a vehicle for specifying requirements and driving the systems-development process, and they discussed their requirements toward the HCWS. The outcome of the workshop was an initial effects specification, structured into the five levels of the effects hierarchy. For the next two workshops, the vendor prepared early prototypes based on the effects specification. These workshops unfolded around a walkthrough of the prototype and effects specification. While the prototype provided a concrete basis for the discussion, the main focus at the workshops remained at the users' work and at effects specification. This was deliberate, and it was easy to achieve because the customer representatives had little background for entering into discussions of technical and design issues. Instead, the customer representatives discussed the HCWS in terms of what might constitute desirable IT-supported patient trajectories for different types of disease. After each workshop the vendor updated and refined the effects specification and the prototype. For the fourth and fifth workshop the prototypes became more mature, and the format of the workshops changed accordingly. During these workshops the customer representatives got hands-on experience with the prototype in situations that increasingly resembled their everyday tasks and data. On this basis the customer representatives provided feedback on the prototype and refined their understanding of the effects they wanted from the HCWS. The vendor incorporated the refinements into the effects specification after each workshop, but at this stage the changes were small, and the focus of the vendor's work moved toward the prototypes.

The customer representatives participating in the workshops were the customer project manager and a few prospective users of the HCWS. They appreciated that specifying with effects meant that they should state their requirements in terms of their work and how they wanted it to change. The customer representatives were knowledgeable about their work and had a vocabulary for talking about it. This was directly useful to them in specifying the effects they wanted from the HCWS. Specifically, the customer representatives were not required to transform desired effects into requirements about system functionality and were, therefore, not dependent on knowledge about how to express requirements in technical terms. As a consequence the transformation of the effects into system functionality was performed by the vendor staff, who in this way got increased freedom to exercise their design skills and an increased responsibility for understanding the users' work sufficiently well to be able to transform the effects into relevant functionality. While the result of the transformation was manifested in the prototypes and presented to the customer representatives for feedback at the next workshop, the vendor's increased freedom during the design process presupposes that the customer is prepared to trust the vendor this freedom. The limited size of the HCWS project contributed to initially reaching the necessary level of trust, but a central factor in maintaining it during the effects-specification process was the configurator, who participated in the workshops and made the prototypes. The configurator was one of two vendor participants in the workshops; the other was the vendor project manager. Thus, the configurator took part in the discussions through which the customer representatives specified and refined the effects and in that way developed a good understanding of the specified effects. At the same time, the customer representatives got a direct impression of whether they succeeded in conveying the meaning of the effects to the configurator during the discussions, and they knew that he was the person responsible for transforming the specified effects into prototype functionality for the next workshop. In the HCWS project it was therefore clear whom the customer representatives needed to trust (in a larger project the configurator may become more of a mediator between the customer and the vendor's developers, and this may necessitate additional work to build a trusting relationship).

An important finding from the effects-specification workshops was that the specified effects at the upper levels of the effects hierarchy were quite stable. Most of the upper-level effects were specified during the first workshop, and they remained valid and unchanged during the remainder of the project. This is promising because it suggests that upper-level effects may guide subsequent efforts by providing a relatively stable framing within which to specify effects at the middle and lower levels. Specification of effects at the middle and lower level of the effects hierarchy required experimentation. One reason for this was that upper-level effects could be implemented in multiple ways at lower levels, and the customer representatives needed time to discuss different possibilities and reach agreement on the approach they wanted to pursue. For example, the upper-level effect "to contribute to an increase in patient motivation" for ceasing to smoke, losing weight, or complying with other preventive or disease-management recommendations could be pursued in multiple ways. Importantly, such effects could not be achieved through system functionality alone but required an integrated approach to technical and organizational development. Another reason for the larger need for experimentation in the specification of effects at the middle and lower levels was that the implications of a middle-level effect in some situations did not become apparent until it was later turned into low-level effects or transformed into system functionality. In this way the prototypes informed the effects specification by revealing unforeseen interactions between effects. For example, it turned out that a large part of the different patient trajectories was generic and that the effects relating to this part of the patient trajectories (approximately 25 effects) could be incorporated in one generic system design. A final reason for the experimentation involved in specifying effects at the middle and lower levels might have been that different stakeholders tended to focus on effects at different levels. The effects specified by customer representatives with managerial positions tended toward the upper levels of the effects hierarchy, whereas customer representatives with operational responsibilities mostly specified middle-level effects. As a consequence, discussion and experimentation were needed to ensure the alignment of upper-level and middle-level effects. The tendency to adjust the middle-level effects might, in part, arise from the organizational hierarchy between managers and operational staff.

## 3.2 Assessing planned and emergent effects: electronic patient record for a stroke unit

As part of the activities involved in the project tender and bid for a strategically important electronic patient record (EPR) contract, effects were specified, realized, and assessed for the use of a clinical-process module of the EPR at the stroke unit of a hospital. This process, which we investigated by means of an action-research study, lasted five months and culminated in a five-day period of pilot use (Simonsen and Hertzum 2008). Effects were specified during a series of workshops, after which the EPR was technically developed and organizationally implemented at the stroke unit. Data about five years of patients at the hospital were migrated to the EPR to achieve a realistic data load. To simulate a fully integrated EPR, a 'back office' was established and staffed around the clock. Record entries that involved paper transactions with other hospital wards were simulated by having the back office continuously monitor the EPR, identify such entries, mail them in the conventional fashion, wait for the results, and immediately type them into the EPR. This way, the EPR replaced all paper records in the stroke unit. The specified effects concerned the clinicians' formation of an overview of the status of each patient and their coordination of their work. These activities were particularly prominent during three clinical activities, which became the focus of the effects assessment: cross-disciplinary team conferences, medical ward rounds, and nursing handovers.

The initial focus of the effects assessment was on specified effects, which were measured as differences between the prior use of paper records and the use of the EPR during the period of pilot use. The established practice of using paper records formed the baseline for measuring the effects of the EPR. Baseline measurements were performed about a month before the pilot period and involved six team conferences, four ward rounds, and five nursing handovers. During the pilot period all clinicians at the stroke unit used the EPR, which was available on all computers and projected on the wall during team conferences and nursing handovers. To safeguard against misunderstandings, which might have entailed risk to patient health, the clinicians had access to supporters who were present around the clock. Measurements similar to those performed during the use of paper records were performed at five team conferences, three ward rounds, and five nursing handovers. The effects measurements included, among others, mental workload, which was measured by the NASA task load index (TLX, Hart and Staveland 1988). TLX ratings were made by each clinician participating in a team conference, ward round, or nursing handover and consisted of assigning a 0-to-100 rating to each of the six TLX subscales: mental demand, physical demand, temporal demand, effort, performance, and frustration.

The assessment yielded planned effects for all three clinical activities involved in the measurements (Hertzum and Simonsen 2008). Most prominently, improvements in mental workload when using the EPR instead of paper records were obtained for two of the three clinical activities, see Figure 4. For the team conferences mental workload was significantly lower on five of the six TLX subscales. For the ward rounds the chief physician's mental workload was significantly reduced, corroborating the results from the team conferences. For the nursing handovers mental workload neither decreased nor increased, but the EPR gave rise to planned decreases in the number of missing pieces of information and in the number of messages to pass on to other clinicians after the nursing handovers.

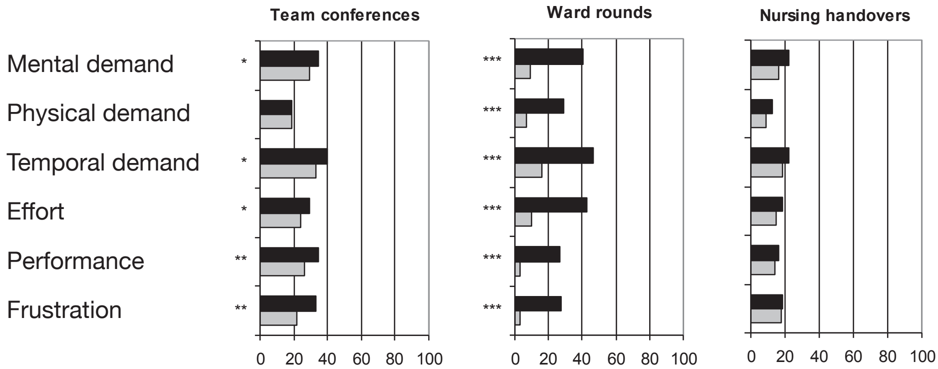


Figure 4. Mental workload for team conferences, ward rounds, and nursing handovers. Black bars give TLX ratings for the use of paper records, grey bars for the use of the EPR. Significant differences—all in favour of the EPR—are indicated with asterisks: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

The changes that occurred during the period of pilot use were, however, not restricted to the specified effects. Some effects emerged spontaneously as a result of the ways in which the clinicians changed their work practices in face of the EPR (Simonsen and Hertzum 2010). A prominent example concerned the nursing handovers and became visible because it differed from the work practices we had encountered when observing the clinicians' use of paper records.

A nursing handover consists of a walkthrough of the admitted patients, led by the nurse team leader and based entirely on reading the patient records; no nurses from the previous shift are present. During the observations of nursing handovers prior to the pilot period, the nurse team leader gave an oral overview of each patient based on the patient records, which were seldom seen by clinicians other than the nurse team leader. Rather, the nurse team leader scanned a patient's paper record and read key information out loud; the other nurses listened. Such oral reporting was an established practice but implied that the nurse team leader constituted a gatekeeper controlling access to the information in the paper record. In contrast, the electronic records were visible to everybody during the pilot period because the screens of the EPR were projected on the wall during the nursing handovers. As a result the nurses engaged in a process of collective reading. The contents of the electronic records were inspected by the group of nurses and they collectively participated in interpreting the status and condition of the patients, guided by the nurse team leader. The nurse team leader navigated the EPR and read selected passages aloud to draw attention to them as well as to set a shared flow in their reading, enabling her to negotiate when to wait for a moment, when to scroll down, and when to open windows with more detailed information. This change in the nurses' work practice emerged during the pilot period, and the nurses explained in follow-up interviews that they were positive toward this new way of working.

The collective reading and interpretation exemplifies that unspecified but beneficial effects may grow out of practice when systems are pilot implemented. Therefore, observation and interview are important elements of effects assessment in order to discover such emergent effects that should be included among the specified effects, which will be more formally assessed during subsequent iterations. Such inclusion of new specified effects resembles opportunity-based change (Orlikowski and Hofman 1997). A decision to include some emergent effects in the set of specified effects indicates that they are considered beneficial, and their existence indicates that it is impossible to specify all desired effects ahead of pilot use.

## 3.3 Working with slipped effects: electronic medication record

To help ensure that the right medication was given to the right patients at the right time an electronic medication record (EMR) had been deployed at the hospitals throughout a healthcare region in Denmark. However, after the EMR had been in use for 2 to 4 years at the individual hospital wards, none of eight main EMR facilities were used consistently by more than 67% of the wards and none of eight mandated work procedures involving the EMR were followed consistently by more than 48% of the wards (Granlien et al. 2008). A medical ward took an effects-driven approach to improve its adoption and use of the EMR. We investigated this process, which lasted nine months, by means of an action-research study (Granlien and Hertzum 2009).

Effects were specified at a workshop in collaboration with clinicians. The workshop participants also identified possible interventions, methods for assessing the effect of the interventions, barriers to their success, and the targeted group of clinicians. On this basis, the participants selected one focal effect, namely that all information about medication was recorded in the EMR. This effect was considered imperative because it provides for a better overview of patients' medication and eliminates the risk of discrepancies between multiple recordings with maltreatment of patients as a possible result. Effects realization focused on the nurses, who sometimes recorded the dispensing and administration of medication in the nursing kardex rather than the EMR. This, for example, happened when a physician had ordered medication orally but not recorded the order in the EMR. In such situations the nurses could not record the dispensing and administration of the medication in the EMR because this was only possible when the medication had previously been ordered in the EMR by a physician. The intervention devised to achieve the effect consisted of introducing delegated medication orders, which enabled all permanently employed, registered nurses at the ward to order selected medication such as light painkillers. With the delegated medication orders the nurses could always record the dispensing and administration of delegated medication in the EMR. Effects assessment consisted of medical record audits, of which two were performed prior to the interventions, two during the intervention period, and two after the interventions had ended. Each audit involved 28 patient records and consisted of reading through all nursing-kardex entries in the selected records to identify any recordings about medication. If any discrepancy existed between such a recording and the content of the EMR, it was considered a violation of the requirement to record all medication information in the EMR.

The intervention period lasted two months and started when the nurses were enabled to make delegated medication orders. A supplementary intervention was performed to inform the nurses about these medication orders. After three weeks all nurses at the ward had received training in the use of delegated medication orders. The nurses were positive about the delegated medication orders, and they started sharing insights about how to use these medication orders. At the first of the two audits performed during the intervention period (in April) it was, therefore, surprising that the number of violations concerning delegated medication had not decreased, see Figure 5. Our follow-up observations at the ward indicated that the nurses sometimes forgot about delegated medication and that the labelling of the steps involved in making delegated medication orders in the EMR was somewhat unintuitive.

As the specified effect had slipped, additional interventions were necessary. Two new organizational interventions were devised as it was not possible to implement quick revisions of the EMR interface: a how-to pocket guide and a box of candy. All nurses received a one-page pocket guide containing two screen dumps annotated with instructions about how to perform delegated medication orders. A copy of the pocket guide was also posted next to the computer in the room where nurses dispensed medication. In addition, a box of candy was placed in the staff room. The lid of the box and each individual bag of wine gums in the box carried a label saying: "The medication out of the nursing kardex and into the EMR". Contrary to the other interventions, the box of candy was purely motivational. The two next audits (in May and June) showed that apart from one violation all delegated medication was recorded in the EMR. This was a statistically significant improvement. Recording all medication information in the EMR made the nurses' work easier, as explained by one nurse: "Now I can stay in the medication room and look in the EMR. I do not have to go back to the office, find the patient's paper record, and look in the kardex". This implies that the nurses benefited from their change of work practice. Consequently, the positive effect of delegated medication orders was not restricted to the physicians, who do not consult the nursing kardex. It appeared as if the planned effect had been achieved. However, at the last audit (in September), three months after the intervention period, the number of violations was not different from the average number of violations at the five earlier audits, suggesting that the effect may again be slipping.

Without effects assessment, the nurses' positive reception of delegated medication orders might have led the clinicians to the incorrect conclusions that the planned effect had been achieved after the first pair of interventions or sustained after the second pair of interventions. Slipped effects call for renewed interventions that may reiterate previous attempts at realizing the effects or seek different ways of doing it. In the present case, an alternative to delegated medication orders could have been to target the physicians. Effects assessment may also reveal that an effect is adverse and should be abandoned. In the present case the effect of having all medication information recorded in the EMR remained valid to the clinicians, but we acknowledge that other studies have advocated the value of certain kinds of redundancy in hospital records (Cabitza et al. 2005).

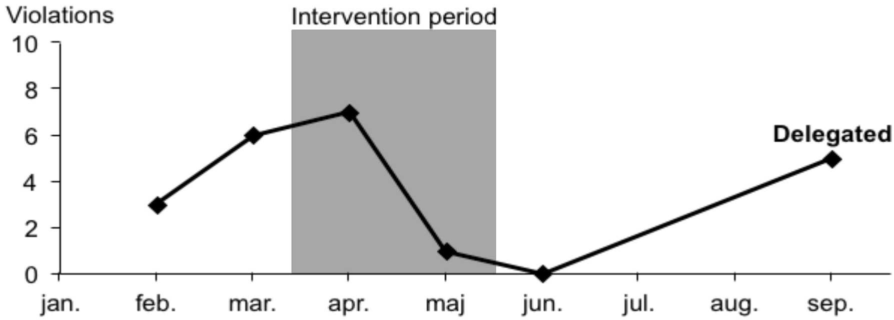


Figure 5. The number of audited records violating the requirement that all information about delegated medication is recorded in the EMR. Only at the May and June audits was the number of violations significantly ($p < 0.05$) lower than the average number of violations at earlier audits

# 4 Discussion

Effects-driven IT development aims to provide vendors and customers with an instrument for managing IT projects. We have described this instrument in terms of the specification, realization, and assessment of effects. In the following, we discuss the promises and challenges involved in working with effects in IT projects and in creating and maintaining the necessary partnership between customer and vendor.

## 4.1 Working with effects in IT projects

Our empirical cases showed that customers and vendors were able to specify, realize, and assess effects and these activities appeared to support them in working toward systems that provided benefit to users. In the HCWS and EPR cases the process of effects specification overlapped the configuration of the systems. For systems being configured on the basis of flexible development tools, this appeared to be a workable approach, and it provided for deriving prototype functionality from specified effects as well as for deriving effects from the ongoing work on prototype functionality. During effects realization, the developers' task was defined by the effects but they had considerable freedom in their decisions about how to transform the effects into functionality. In all three cases the customer was responsible for the organizational changes associated with the systems, but in the EPR case they were implemented in collaboration with the vendor. During effects assessment the absence or presence of specified effects was successfully measured. In the EPR case such measurement of specified effects was combined with an awareness of emergent effects, and in the EMR case measurements were iterated to work systematically toward achieving an effect that initially slipped. In combination, the activities of effects specification, realization, and assessment maintained a sustained focus on effects. Vendors and, particularly, customers appeared to find it easy to understand and relate to the notion of effects, and while effects at the lower levels of the effects hierarchy were subject to experimentation and change, upper-level effects remained stable. This stability is in contrast to the common experience of a discrepancy between a functional specification of requirements and users' actual requirements, as stated by for example Coble et al. (1997, p. 175): 'they [developers] argue that since the customer agreed to the functional specifications, the valid acceptance test is "meets specs", not "fulfills user requirements"'.

To summarize, we consider three aspects of the work with effects promising:

- Effects appear to be a 'language' that customers easily pick up and with which they can effectively express what they want from a system. Vendors, on their part, appear to be able to transform an effects specification to tentative system functionality. In combination, effects specifications and prototypes that illustrate system functionality have proved an effective means of communication between vendor and customer in the HCWS and EPR cases.
- The effects hierarchy provides a tool for aligning and working with effects at different levels of abstraction. It should be possible to link any effect to higher-level effects that provide the rationale for the effect and to lower-level effects that specify its concrete implementation. The HCWS case shows that different stakeholder groups may focus on effects at different levels. This way the linking of effects across levels also becomes a process of creating and checking the alignment of effects across stakeholder groups.
- Specified effects can be measured and emergent effects identified during effects assessment. Repeated measurements provide a means of countering that effects slip. This entails that effects-driven IT development, as exemplified by the EPR and EMR cases, extends benefits management (Ward and Daniel 2006) by detailing how benefits can concretely be assessed and how such assessment can be incorporated in IT projects through formative evaluations, rather than postponed until after project completion.

While we consider these promising aspects of working with effects to be important, there are at the same time multiple challenges involved in specifying, realizing, and assessing effects in the context of IT projects. The empirical cases suggest that the challenges can be overcome but they also illustrate that the IT-project context constrains, in particular, effects assessment. Four challenges stand out:

First, none of our cases have investigated the evolution of a set of effects over several iterations of effects specification, realization, and assessment. This, for example, leaves open how much up-front effects specification is required to strike the balance between setting the direction and scope of a project and allowing for gradually reaching an understanding of what is desirable and possible. In our three cases the effects have been specified through a series of up to five workshops, but the HCWS case concerned a system of limited size and the EPR and EMR cases focused more on specifying a small set of pertinent effects than on specifying an exhaustive set of effects. The low number of up-front effects-specification workshops in our cases points toward an iterative approach where the first effects assessments may lead to considerable revision of the set of specified effects, particularly at the middle and lower levels of the effects hierarchy. Compared to effects-driven IT development, benefits management (Ward and Daniel 2006) puts more emphasis on up-front specification because the assessment of whether benefits are achieved is not made until after the end of the project where the possibilities for iteration are limited in that they involve either reopening the project or starting a new project. The EMR case shows that after multiple iterations even an effect that was specified up front may still be slipping, and the effects taxonomy suggests that other types of effect may also require iteration. Thus, vendors and customers appear to need an instrument for working systematically with more types of effect than the specified effects that are realized as anticipated and lead to planned benefits. While our cases show that pertinent effects can be specified up front with modest effort and subsequently measured, we so far have insufficient experience with the exhaustiveness of the sets of specified effects.

Second, effects assessments must balance measurement of specified effects and identification of emergent effects. Measurement points toward counting of, for example, the number of violations of the requirement to record all information about medication in the EMR or toward ratings of, for example, mental workload. In the EPR case we experienced that standardized measurement instruments such as the TLX for mental workload were more readily accepted than tailor-made measurement instruments, which gave rise to discussion about how to interpret the measurements. Conversely, identification of emergent effects points toward observation, interview, and attention to unexpected ways of working with the system. Emergent effects will often be tailor-made responses to possibilities provided by the system in specific work situations, and this situatedness is generally perceived as an indication of relevance and importance rather than as uncertainty about how to interpret the emergent effects. Activities aiming to measure specified effects normally cannot substitute for activities aiming to identify emergent effects, and vice versa, because methods as well as criteria tend to differ. In striking a balance between the two kinds of activity it may be difficult to ensure sufficient attention to the identification of emergent effects, which have yet to be recognized and prioritized relative to the already specified effects (Farbey et al. 1999). This difficulty is not specific to effects-driven IT development but central to the division between plan-driven and iterative approaches to systems development (Boehm and Turner 2004; Larman and Basili 2003).

Third, the timing of effects assessments is a trade-off between, on the one hand, assessing after short periods of use to acknowledge project deadlines, save resources, and reduce diffusion of ineffective systems and, on the other hand, assessing after longer periods of use to allow system errors to be corrected, users to gain proficiency, work practices to stabilize, and the long-term implications of system use to surface. The EPR case exemplifies that effects-driven IT development may, for reasons external to the IT project itself, be confined to short pilot implementations. While it is encouraging that improvements could be measured after using the EPR for only five days, the EMR case illustrates that repeated effects assessments over a longer period of time may be necessary to achieve specified effects and possibly also to get beyond the goodwill that can be invested in trying something new for a restricted period of time. Little research has examined the learning curves in, for example, healthcare technologies (Ramsay et al. 2000) and it is, therefore, difficult to estimate when the learning effects associated with different kinds of system and organizational change have worn off. Jurison (1996) estimates that effects at the level of individual users can be observed within 6-8 months whereas effects at the organizational level may take a year to materialize. Our empirical cases indicate that these estimates are overly pessimistic, but we also acknowledge the risk of agreeing to overly brief pilot implementations in order to accommodate IT-project schedules.

Fourth, the effects that are assessed during pilot implementations are a result of multiple, interrelated factors. In starting to use a new IT system, users are not simply replacing one tool with another while everything else remains unchanged. Systems are accompanied by changes in individual users' tasks, in collective work practices, and in required competences, status, and organizational structures. It may also be necessary to introduce special precautions, such as proactive support staff, during pilot implementations because it is too risky to leave users of safety-critical systems to unassisted trial and error when they encounter situations not covered by training. On the one hand, effects-driven IT development aims to ensure that IT projects do not become dissociated from the process of organizationally implementing the systems; on the other hand, the factors affecting organizational implementation may appear too diverse and too fuzzy to enable reliable effects assessment. The effects assessment must be carefully set up, and for both vendor and customer this implies attention to organizational implementation and agreement about how to create the conditions for the adoption of the system and associated work practices. Customer, vendor, or both may be reluctant to extend their collaboration to also include these activities. A reason for especially vendor reluctance could be uncertainty about whether they will have sufficient influence on the factors that determine whether specified effects are achieved and, thereby, whether projects are successful. In our cases such reluctance from the vendor has, however, not been prominent.

## 4.2 A partnership between customer and vendor

Effects-driven IT development requires that customer and vendor trust each other sufficiently to enter into a collaboration that blends technical development and organizational implementation. This may make effects-driven IT development most relevant in situations where customer and vendor have, or see an interest in developing, a close, long-term partnership. Basing such partnerships on the specification, realization, and assessment of effects holds, we contend, promise for both vendor and customer:

- Vendors can enhance their business area from IT systems to complete business solutions including organizational implementation and change management. Thus, a broader range of vendors' expertise may be appreciated and valued. In addition, documentation of the usage effects obtained from a vendor's solutions may strengthen the marketing effect toward other customers.

- Customers can focus on conceptual proposals defining the problem and on desired outcomes in terms of specified effects, as opposed to more narrowly conceived usability issues or a detailed functional specification. This does not require detailed insight into technical issues. In addition, a partnership with the vendor can support long-term efforts to accomplish substantial change in an incremental manner.

A partnership with specialists among the users will likely support vendors in devising solutions that deliver desired effects, and users may more readily adopt associated changes in their work practice if the changes can be presented along with descriptions of the effects they seek to produce. There are, however, several unresolved challenges involved in establishing a partnership in which customer and vendor share the responsibility of providing IT systems and associated work practices that yield specified usage effects:

First, the roles of vendor and customer need to be renegotiated because effects-driven IT development entails a more permeable boundary between development and implementation. In addition to shifts in the allocation of existing tasks between vendor and customer, new competences are required to specify effects, conduct pilot implementations, and assess effects. These new competences must be acquired by vendor staff or they may, in large customer organizations with many IT projects, be acquired by customer staff. Our responsibility as action researchers in our empirical cases has in particular been effects assessment, which is central to effects-driven IT development but also difficult, for example because many factors may confound measurements (Hamilton and Chervany 1981). Assigning the measurement of the specified effects to a third party, external to both vendor and customer, may be a way of improving the credibility of the measurements and easing the customer-vendor partnership about the remaining activities.

Second, effects-driven IT development is most relevant in situations where the organizational context is complex but these are also the situations in which effects specification and organizational implementation are likely to be most difficult. When the organizational context is complex, a development-implementation chasm incurs more risk because less of the understanding pertinent to the development of a successful system can be acquired without trying out pilot versions of the system in real contexts (Hertzum et al. 2011; Markus 2004). However, a complex organizational context also entails that many groups may have a stake in the system, the effects they want to pursue may be in conflict, pilot implementations may be difficult to scope, organizational implementation may face multiple barriers, and effects assessment may become politicized (see, also, Farbey et al. 1999). In the healthcare sector, the contextual complexity, for example, includes that the organizational implementation and use of a system often involve multiple groups of collaborating actor, only some of which under the management of the customer that introduces the new system. As a consequence, vendors and customers may consider a partnership based on specifying, realizing, and assessing effects to be too fuzzy or too risky in complex organizational contexts. While the limited size of the system in the HCWS case partly reflected such considerations on the vendor's part, the same vendor participated in the EPR case, which concerned a more complex organizational context.

Third, using effects-driven IT development as an instrument for managing IT projects suggests that contract fulfilment could involve whether specified effects are achieved. Making contract fulfilment dependent on achieving specified effects would ensure a sustained focus on the specification, realization, and assessment of effects. However, this presupposes a hardening of the work with effects in that they change from an instrument for informing and guiding IT-project activities to an instrument for contractually regulating the customer-vendor relationship. In our cases we have so far explored effects-driven IT development as an instrument for informing and guiding IT-project activities, but our long-term goal is to explore ways of inscribing effects in IT-project contracts. We do, however, heed Behn and Kant's (1999) warning that performance contracting may inhibit experimentation. It is crucial to maintain a balance between achieving the specified effects that are subject to measurement and remaining receptive to emergent effects, which are beneficial but not subject to measurement. Mechling (1999) find that whereas practitioners, at least on the customer side, are very optimistic about the potential of performance contracting there is at the same time little real-world experience to learn from and great uncertainty about how to proceed. While we share this uncertainty, we also envisage that it will be difficult to obtain a sustained focus on effects without inscribing them in project contracts.

# 5 Conclusion

Often, IT projects do not produce the effects customers are aiming to achieve. A somewhat surprising reason for this is that few systems-development approaches advocate a sustained focus on effects throughout technical development and organizational implementation. With effects-driven IT development we propose an instrument for avoiding a chasm between development and implementation by making specification of effects and assessment of whether they are achieved during pilot use central activities of IT projects. We have explored effects-driven IT development through three empirical cases in the healthcare domain, but we believe the approach is more generally applicable to IT projects that seek to develop systems and change organizations. Effects-driven IT development can be summarized in three activities that form an iterative process:

- Effects specification, which consists of identifying and aligning effects across multiple stakeholders and hierarchical levels. This part of effects-driven IT development resembles other approaches to systems development, particularly benefits management.
- Effects realization, which consists of making IT systems and organizational change in ways that allow for pilot implementations of the system while it is not yet finalized. Pilot implementations are particularly valuable when effects must be realized in complex organizational settings.
- *Effects assessment*, which consists of measuring the absence or presence of specified effects while also remaining alert to emergent effects. By assessing effects during pilot implementations the assessment involves the system as well as its adoption and use, and the results of the assessment can provide guidance for the ongoing project activities.

Based on our empirical cases we find that several aspects of effects-driven IT development are promising. These promising aspects include that effects appear to be a language customers easily pick up and that using effects as an instrument for managing IT projects may provide for appreciating and valuing a broader range of a vendor's expertise. There are, however, also challenges. One set of challenges concerns the concrete work with effects and, for example, includes balancing the measurement of specified effects against the identification of emergent effects over several iterations of effects specification, realization, and assessment. In addition, the measurement of specified effects involves difficult decisions about the timing of measurements, the scope of pilot implementations, and the need for special precautions to safeguard against critical errors. Another set of challenges concerns the customer-vendor relationship, which must be renegotiated to reach a mutual understanding about how to transform effects specifications into system functionality, how to facilitate the adoption of the system and associated organizational change, and how to proceed in the face of adverse or slipped effects. Part of the renegotiation of the relationship between customer and vendor could involve that contract fulfilment is made dependent on achieving specified effects. This would ensure a sustained focus on effects, but previous experiences with performance contracting, mainly in areas other than systems development, indicate that considerable future work is required to inscribe effects in IT-project contracts.

# 6 Acknowledgements

This article is an important milestone for the authors because it gives a coherent account of an extensive research program we have jointly conducted since 2004. The authors have contributed equally to the establishment, consolidation, and management of the research program, to the execution of the empirical work, and to the development of theory and tools. Morten Hertzum has, however, written most of this article. For a description of the research program on effects-driven IT development and the six empirical projects conducted during 2004-2011, we refer to Hertzum and Simonsen (2011).

We are grateful for the inspiring collaboration we have had with Anders Barlach and Maren Granlien during our work with effects-driven IT development in the period 2006-2009. We especially acknowledge Anders Barlach's principal role in the HCWS case and Maren Granlien's principal role in the EMR case. We also thank CSC Scandihealth and Region Zealand for their participation in and co-funding of this research. Special thanks are due to the clinicians and developers who were involved in the three empirical cases.

# 7 References

Alter, S., (2001). Which life cycle—Work system, information system, or software? *Communications of the Association for Information Systems*, (7):17:01-17:54.

Ash, J. S., Berg, M., and Coiera, E., (2004). Some unintended consequences of information technology in health care: The nature of patient care information system-related errors. *Journal of the American Medical Informatics Association*, (11:2): 104-112.

Barlach, A., and Simonsen, J., (2008). Effect specifications as an alternative to use cases. In: *IRIS31: Proceedings of 31st Information Systems Research Seminar in Scandinavia*, A. Asproth, K. Axelsson, S. C. Holmberg, C. Ihlström and B. Sundgren (eds.), Åre, Sweden.

Behn, R. D., and Kant, P. A., (1999). Strategies for avoiding the pitfalls of performance contracting. *Public Productivity & Management Review*, (22:4): 470-489.

Beynon-Davies, P., Tudhope, D., and Mackay, H., (1999). Information systems prototyping in practice. *Journal of Information Technology*, (14:1): 107-120.

Boehm, B., (2000). Requirements that handle IKIWISI, COTS, and rapid change. *IEEE Computer*, (33:7): 99-102.

Boehm, B., and Turner, R., (2004). *Balancing agility and discipline: A guide for the perplexed*, Pearson, Boston.

Brynjolfsson, E., and Hitt, L. M., (2000). Beyond computation: Information technology, organizational transformation and business performance. *Journal of Economic Perspectives*, (14:4): 23-48.

Bødker, K., Kensing, F., and Simonsen, J., (2004). *Participatory IT design: Designing for business and workplace realities*, MIT Press, Cambridge.

Cabitza, F., Sarini, M., Simone, C., and Telaro, M., (2005). When once is not enough: The role of redundancy in a hospital ward setting. In: *Proceedings of the GROUP '05 Conference on Supporting Group Work*, K. Schmidt, M. Pendergast, M. Ackerman and G. Mark (eds.), ACM Press, New York, pp. 158-167.

Coble, J. M., Karat, J., and Kahn, M. G., (1997). Maintaining a focus on user requirements throughout the development of clinical workstation software. In: *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems*, ACM Press, New York, pp. 170-177.

Cockburn, A., (2007). *Agile software development: The cooperative game*, (2nd ed.), Addison-Wesley, Upper Saddle River.

Connell, K., Andal, D., and Brown, C. L., (1995). *Performance based procurement. Another model for California*. California Franchise Tax Board, Sacramento.

Devaraj, S., and Kohli, R., (2003). Performance impacts of information technology: Is actual usage the missing link? *Management Science*, (49:3): 273-289.

Dybå, T., and Dingsøyr, T., (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, (50:9&10): 833-859.

Farbey, B., Land, F., and Targett, D., (1999). The moving staircase: Problems of appraisal and evaluation in a turbulent environment. *Information Technology & People*, (12:3): 238-252.

Fichman, R. G., and Kemerer, C. F., (1999). The illusory diffusion of innovation: An examination of assimilation gaps. *Information Systems Research*, (10:3): 255-275.

Fichman, R. G., and Moses, S. A., (1999). An incremental process for software implementation. *Sloan Management Review*, (40:2): 39-52.

Glass, R. L., (1997). Pilot studies: What, why, and how. *Journal of Systems and Software*, (36:1): 85-97.

Good, M., Spine, T. M., Whiteside, J., and George, P., (1986). User-derived impact analysis as a tool for usability engineering. In: *Proceedings of the CHI'86 Conference on Human Factors in Computing Systems*, ACM Press, New York, pp. 241-246.

Granlien, M. F., Hertzum, M., and Gudmundsen, J., (2008). The gap between actual and mandated use of an electronic medication record three years after deployment. In: *MIE2008: Proceedings of the XXIst International Congress of the European Federation for Medical Informatics*, S. K. Andersen, G. O. Klein, S. Schulz, J. Arts and M. C. Mazzoleni (eds.), IOS Press, Amsterdam, pp. 419-424.

Granlien, M. S., and Hertzum, M., (2009). Implementing new ways of working: Interventions and their effect on the use of an electronic medication record. In: *Proceedings of the GROUP 2009 Conference on Supporting Group Work*, ACM Press, New York, pp. 321-330.

Hamilton, S., and Chervany, N. L., (1981). Evaluating information system effectiveness - Part I: Comparing evaluation approaches. *MIS Quarterly*, (5:3): 55-69.

Han, Y. Y., Carcillo, J. A., Venkataraman, S. T., Clark, R. S. B., Watson, R. S., Nguyen, T. C., Bayir, H., and Orr, R. A., (2005). Unexpected increased mortality after implementation of a commercially sold computerized physician order entry system. *Pediatrics*, (116:6): 1506-1512.

Harrington, H. J., (1991). *Business process improvement: The breakthrough strategy for total quality, productivity, and competitiveness*, McGraw-Hill, New York.

Hart, S. G., and Staveland, L. E., (1988). Development of NASA-TLX (task load index): Results of empirical and theoretical research. In: *Human Mental Workload*, P. A. Hancock and N. Meshkati (eds.), North-Holland, Amsterdam, pp. 139-183.

Haux, R., Winter, A., Ammenwerth, E., and Brigl, B., (2004). *Strategic information management in hospitals: An introduction to hospital information systems*, Springer, Heidelberg.

Hertzum, M., Bansler, J. P., Havn, E., and Simonsen, J., (2011). Pilot implementation: Learning from field experiments in IS development, submitted for publication.

Hertzum, M., and Simonsen, J., (2008). Positive effects of electronic patient records on three clinical activities. *International Journal of Medical Informatics*, (77:12): 809-817.

Hertzum, M., and Simonsen, J., (2011). Effects-driven IT development: Status 2004-2011. In: *Balancing Sourcing and Innovation in Information-Systems Development*, M. Hertzum and C. Jørgensen (eds.), Tapir Academic Publishers, Trondheim, pp. 165-192.

Häkkinen, L., and Hilmola, O.-P., (2008). ERP evaluation during the shakedown phase: Lessons from an after-sales division. *Information Systems Journal*, (18:1): 73-100.

Jurison, J., (1996). The temporal nature of IS benefits: A longitudinal study. *Information & Management*, (30:2): 75-79.

Landauer, T. K., (1995). *The trouble with computers: Usefulness, usability, and productivity*, MIT Press, Cambridge.

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

LeRouge, C., Mantzana, V., and Wilson, E. V., (2007). Special section on healthcare information systems research, revelations and visions. *European Journal of Information Systems*, (16:6): 669-760.

Leveson, N. G., (2000). Intent specifications: An approach to building human-centered specifications. *IEEE Transactions on Software Engineering*, (26:1): 15-35.

Love, P. E. D., Irani, Z., Standing, C., Lin, C., and Burn, J. M., (2005). The enigma of evaluation: Benefits, costs and risks of IT in Australian small-medium-sized enterprises. *Information & Management*, (42:7): 947-964.

Markus, M. L., (2004). Technochange management: Using IT to drive organizational change. *Journal of Information Technology*, (19:1): 4-20.

Maulsby, D., Greenberg, S., and Mander, R., (1993). Prototyping an intelligent agent through Wizard of Oz. In: *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, ACM Press, New York, pp. 277-284.

Mechling, J., (1999). Better funding for government IT: Views from the front line. *Journal of the American Society for Information Science*, (50:4): 305-313.

Orlikowski, W. J., (1996). Improvising organizational transformation over time: A situated change perspective. *Information Systems Research*, (7:1): 63-92.

Orlikowski, W. J., and Hofman, J. D., (1997). An improvisational model for change management: The case of groupware technologies. *Sloan Management Review*, (38:2): 11-22.

Ramsay, C. R., Grant, A. M., Wallace, S. A., Garthwaite, P. H., Monk, A. F., and Russell, I. T., (2000). Assessment of the learning curve in health technologies: A systematic review. *International Journal of Technology Assessment in Health Care*, (16:4): 1095-1108.

Rasmussen, J., Pejtersen, A. M., and Goodstein, L. P., (1994). *Cognitive Systems Engineering*, Wiley, New York.

Scott, J. T., Rundall, T. G., Vogt, T. M., and Hsu, J., (2005). Kaiser Permanente's experience of implementing an electronic medical record: A qualitative study. *BMJ*, (331:7528): 1313-1316.

Simonsen, J., and Hertzum, M., (2008). Participative design and the challenges of large-scale systems: Extending the iterative PD approach. In: *PDC2008: Proceedings of the Tenth Anniversary Conference on Participatory Design*, J. Simonsen, T. Robertson and D. Hakken (eds.), ACM Press, New York, pp. 1-10.

Simonsen, J., and Hertzum, M., (2010). Iterative participatory design. In: *Design Research: Synergies from Interdisciplinary Perspectives*, J. Simonsen, J. O. Bærenholdt, M. Büscher and J. D. Scheuer (eds.), Routledge, London, pp. 16-32.

Sobol, M. G., Alverson, M., and Lei, D., (1999). Barriers to the adoption of computerized technology in health care systems. *Topics in Health Information Management*, (19:4): 1-19.

Vicente, K. J., (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*, Erlbaum, Mahwah.

Ward, J., and Daniel, E., (2006). *Benefits management: Delivering value from IS & IT investments*, Wiley, Chichester.

Ward, J., and Elvin, R., (1999). A new framework for managing IT-enabled business change. *Information Systems Journal*, (9:3): 197-221.

Ward, J., Taylor, P., and Bond, P., (1996). Evaluation and realisation of IS/IT benefits: An empirical study of current practice. *European Journal of Information Systems*, (4:4): 214-225.

Whiteside, J., Bennett, J., and Holtzblatt, K., (1988). Usability engineering: Our experience and evolution. In: *Handbook of Human-Computer Interaction*, M. Helander (ed.), Elsevier, Amsterdam, pp. 791-817.