

6-2015

# A Workaround Design System for Anticipating, Designing, and/or Preventing Workarounds

Steven Alter

*University of San Francisco, [alter@usfca.edu](mailto:alter@usfca.edu)*

Follow this and additional works at: <http://repository.usfca.edu/at>

 Part of the [Management Information Systems Commons](#)

---

## Recommended Citation

Alter, S. (2015) "A Workaround Design System for Anticipating, Designing, and/or Preventing Workarounds," EMMSAD 2015 (Exploring Modeling Methods for Systems Analysis and Design) a working conference associated with CAISE 2015 (Conference on Advanced Information System Engineering), June 8-12, Stockholm, Sweden

This Article is brought to you for free and open access by the School of Management at USF Scholarship: a digital repository @ Gleeson Library | Geschke Center. It has been accepted for inclusion in Business Analytics and Information Systems by an authorized administrator of USF Scholarship: a digital repository @ Gleeson Library | Geschke Center. For more information, please contact [repository@usfca.edu](mailto:repository@usfca.edu).

# A Workaround Design System for Anticipating, Designing, and/or Preventing Workarounds

Steven Alter

University of San Francisco  
alter@usfca.edu

**Abstract.** Idealized system design produces requirements reflecting management intentions and “best practices.” This paper proposes a workaround design system (WDS) for anticipating, designing, and/or preventing workarounds that bypass systems as designed. A WDS includes a process and an interactive “workaround design tool” (WDT) for identifying and evaluating foreseeable workarounds based on work system theory and a theory of workarounds. This paper summarizes the conceptual background and explains the form, use, and implications of the proposed WDS and WDT.

The idea of WDS addresses significant gaps in practice and research. Designers should have methods for identifying likely obstacles and anticipating and evaluating a non-trivial percentage of plausible workarounds. Methods for identifying workarounds might help in training work system participants. Researchers might use WDS to explore why specific responses to obstacles did or did not occur. The lack of methods related to anticipating, designing or preventing workarounds implies that WDS may prove fruitful even though it is impossible to anticipate all possible workarounds.

**Keywords:** workaround, systems analysis and design, business process management, emergent change

## 1 Augmenting Design by Placing Workarounds in the Foreground

As a contribution to EMMSAD 2015 (Exploring Modeling Methods for Systems Analysis and Design), this paper<sup>1</sup> introduces a way to highlight and discuss an important topic that is ignored or barely mentioned in most discussions of modeling methods and systems analysis and design. The topic is workarounds, a widely recognized phenomenon in everyday business life. Some workarounds attempt to overcome

---

<sup>1</sup> This paper will be presented at EMMSAD 2015 (Exploring Modeling Methods for Systems Analysis and Design) a working conference associated with CAISE 2015 (Conference on Advanced Information System Engineering), June 8-12, Stockholm, Sweden. An abbreviated version will appear in the Proceedings.

unanticipated obstacles; others bypass cumbersome processes or technologies; yet others involve taking personal advantage of incomplete management oversight. Some authors view workarounds as essential occurrences in everyday work and even as sources of innovation; others view them as inappropriate or hazardous activities, non-compliance, or opportunistic behavior that undermines management intentions.

Workarounds are important for systems analysis and design because ignoring them has negative consequences. First, it places systems at greater jeopardy by increasing the probability that inappropriate workarounds will not be anticipated and controlled, and therefore will undermine whatever the systems are trying to achieve. Ignoring foreseeable workarounds also increases the probability of creating cumbersome features that work system participants will view as sure indications that the original designers did not understand the nature and details of the work being done.

In effect, this paper calls into question the implicit or explicit assumptions that application software captures and enforces best practices. Frequent examples in the management, operations, and sociotechnical literature demonstrate that work system participants with even a modicum of behavioral discretion may perform activities in ways that were not prescribed by the software or by management, and may act in ways that conflict directly with officially sanctioned processes. Systems analysis and design and related modeling methods should address those issues if the goal is to build realistic systems that will achieve business goals.

**Augmenting established methods.** This paper's new idea is to augment the established analysis and design approach of determining requirements that reflect an idealized specification of a business process and of related usage patterns for software. Augmenting the traditional emphasis on best practices or on the sanctioned business process, the new idea is a workaround design system (WDS) focusing on what a work system participant should do or is likely to do when current or proposed specifications of routines, processes, best practices, or methods do not fit realities that they may encounter. This involves imagining exceptions or obstacles and identifying appropriate responses. Those responses may involve workarounds or may address exceptions and obstacles in other ways. Thus, the idea of a WDS is quite different from typical approaches such as identifying alternate paths in typical use case narratives.

The idea of a WDS starts with a broadly defined process that includes identifying the work system, identifying foreseeable exceptions or obstacles that might call for a workaround, identifying plausible workarounds, evaluating those workarounds, and deciding how to adjust the design, if necessary. The WDS uses a workaround design tool (WDT) that provides knowledge-based support for each step through templates and compilations of available knowledge in forms such as lists of typical workaround drivers, lists of typical design moves and characteristics that might change, and possibly even workaround design patterns that resemble design patterns for software.

A systematic approach for imagining and evaluating foreseeable workarounds could be applied in many types of situations, implying many possible uses of a WDS. The following potential uses of a WDS augment the established design approach of determining requirements that reflect an idealized specification of a business process or that look at a small number of alternative paths:

- An effective WDS might help managers and system designers identify circumstances under which an IT-enabled work system might be bypassed or undermined. Anticipation of exceptions or obstacles and the resulting workarounds might help in designing the work system (including related IS/IT support) to encourage appropriate behavioral discretion while also blocking inappropriate workarounds. Notice how this goes beyond typical IS risk analysis because it asks specifically about foreseeable workarounds that might be developed by work system participants.
- An effective WDS might lead to more complete and useful instructions about what to do when foreseeable exceptions and obstacles occur in real world practice.
- An effective WDS might be incorporated into training during and after implementation of new or improved work systems. Use of the WDS might help work system participants understand their own work system in a deeper way.
- An effective WDS might sensitize managers and designers to be more realistic about how work systems and software will be developed, implemented, and used.
- An effective WDS might encourage more effective participation in analysis and design efforts by giving work system participants a way to contribute that engages both their imagination and their knowledge of their own work settings.
- An effective WDS might make likely workarounds visible as a contribution to future improvements in the work system or information system that supports it.

**Source of ideas about workarounds.** This paper applies a theory of workarounds that was developed as part of research about unplanned change [1]. The theory attempted to encompass workarounds discussed in “300+ articles” that mentioned examples of workarounds or ideas about workarounds. The articles were found through Google Scholar searches such as “workaround + nursing” or “workaround + bureaucracy” or “workaround + hazard.” Those articles were from disciplines including information systems, medical informatics, operations management, organization behavior, management, ergonomics, and public administration.

**Related methods and research.** The idea of using a WDS for anticipating foreseeable workarounds is a new variation on a long established practice of enriching design and planning processes by identifying events whose occurrence might have a significant impact. Strategy studies have used many related methods (e.g., scenario analysis, cross impact analysis, and Delphi studies). At an operational level, various forms of risk analysis and crisis management planning have been used (e. g., failure modes and effects analysis (FMEA) in Six Sigma and Monte Carlo simulation).

The idea of WDS also overlaps with research areas such as the following:

- organizational routines as the basis of change (e.g., [2])
- emergent change (e.g., [3,4])
- extensions of BPM (e.g., [5]) to address exceptions and contingencies (e.g., [6])
- process aware information systems (e.g., [7,8])
- context aware information systems (e.g., [9])
- tailorable technology and secondary design (e.g., [10])
- adaptive case management and dynamic case management (e.g., [11,12])
- behavioral programming, (e.g. [13]).

**Organization.** This paper's explanation of the proposed WDS and WDT starts by summarizing the theory of workarounds that forms the basis of a WDS. It explains that a WDS is a temporary work system for devising workarounds, supported by a software-based tool called a workaround design tool (WDT). It explains some of the possible capabilities and sources of knowledge for a WDT. As a contribution to a conference on exploring modeling methods and systems analysis and design, it explains a new approach for taking workarounds seriously enough to incorporate the anticipation of workarounds into assumptions about how systems in organizations operate and how to analyze and design systems. A multi-year research project could build on these ideas by following the entire design science research cycle including the accumulation of relevant knowledge for the WDT, creation of a WDT, and testing the entire WDS/WDT approach in experimental or real world design situations.

## **2 Background about Workarounds**

The proposed WDS and WDT are based on a theory of workarounds that is explained in depth in [1], which includes a lengthy literature review covering previous discussions of workarounds along with many examples from various disciplines. This paper mentions ideas from the previous paper's literature review but due to page limitations cannot cite the many references cited in the previous paper. The following background about a theory of workarounds suffices for the explanation of WDS and WDT that appears later in the paper.

### **2.1 Definition of Workaround and Related Preconditions**

Workarounds occur for a variety of reasons. In some cases, people trying to do their work need to respond in some way to unanticipated obstacles or exceptions. In other cases, workarounds bypass cumbersome or inefficient process steps. Workarounds may bypass organizational routines that emerged over time without an explicit design and fail to consider important contingencies. Activities may deviate from expectations due to a lack of knowledge or training, personal opportunism, or other reasons. Regardless of the driver, workarounds are often a springboard for change, especially when they challenge the stability and coherence of processes and systems that no longer serve the organization, its employees, or its customers.

To accommodate all of the different types of situations in which workarounds occur, the theory of workarounds is based on the following definition of workaround:

A workaround is a goal-driven adaptation, improvisation, or other change to one or more aspects of an existing work system in order to overcome, bypass, or minimize the impact of obstacles, exceptions, anomalies, mishaps, established practices, management expectations, or structural constraints that are perceived as preventing that work system or its participants from achieving a desired level of efficiency, effectiveness, or other organizational or personal goals.

[1] explains how this definition is broader and more encompassing than 12 other definitions in the literature. For example, a comprehensive view of workarounds includes adaptations that may occur in any part of a work system rather than just in processes or technologies. According to [14,15], a work system is a system in which human participants and/or machines perform work using information, technology, and other resources to produce products/services for internal or external customers.

Workarounds affect details of a work system's operation, either temporarily or over an extended period, but do not change its overall identity, purpose, and high-level architecture. Aspects of WDS also apply to many workarounds in non-organizational settings that are peripheral to this paper's focus, such as using a substitute material in a home project, selecting an alternate driving route to avoid a parade, or selecting a non-preferred menu item in a restaurant because the preferred item is unavailable. Workarounds may be totally ethical, ethically questionable, or fraudulent. Decisions related to creating and executing workarounds may or may not consider ethics and legality along with many other factors.

**Preconditions.** With the above definition, preconditions for the occurrence of a workaround include the following:

- A specific process, policy, or set of practices within an existing work system
- Organizational and/or personal goals related to that situation
- An obstacle, exception, anomaly, mishap, established practice, management expectation, or structural constraint that might be perceived as something to bypass or overcome
- An ability to imagine and execute a workaround.

**Actions that are not workarounds.** To clarify the scope of this discussion, it is worthwhile to mention common types of goal-directed actions or activities in organizations that are not workarounds. The following are not considered workarounds:

- Reengineering projects or other formal projects designed to produce major work system changes. These are not workarounds because major changes would affect the work system's high level architecture.
- Events or work system changes that occur due to inattention, accidents, or mistakes of work system participants. These are not workarounds because they are not goal-driven adaptations, improvisations, or other activities that attempt to bypass or overcome obstacles or exceptions.
- Improvisation or bricolage not involved with overcoming obstacles, exceptions, anomalies, mishaps, or structural constraints in specific processes or practices within a work system.
- Criminal actions, sabotage, or other attacks by people who are neither work system participants nor their direct managers. These are not workarounds because they are not adaptations or improvisations by work system participants.

## 2.2 Goals and Effects of Workarounds

Table 1 summarizes goals of workarounds, effects of workarounds, and perspectives on workarounds that were found in the literature review of workarounds in [1]. Table 1 illustrates the wide diversity of views of workarounds. Each entry in Table 1 is based on several examples from that literature review.

**Table 1.** Goals of Workarounds, Direct Effects of Workarounds, and Perspectives on Workarounds

<i>Various Goals of Workarounds</i>	<i>Various Direct Effects of Workarounds</i>	<i>Various Perspectives on Workarounds</i>
<ul style="list-style-type: none"> <li>• Overcome inadequate IT functionality.</li> <li>• Bypass obstacles built into existing routines.</li> <li>• Bypass or overcome transient obstacles due to anomalies or mishaps.</li> <li>• Respond to mishaps with quick fixes.</li> <li>• Augment existing routines without developing new resources.</li> <li>• Substitute for unavailable or inadequate resources.</li> <li>• Design and implement new resources.</li> <li>• Prevent mishaps.</li> <li>• Pretend to comply.</li> <li>• Lie, cheat, steal for personal benefit.</li> <li>• Collude for mutual benefit.</li> </ul>	<ul style="list-style-type: none"> <li>• Continuation of work despite obstacles, mishaps, or anomalies.</li> <li>• Creation of hazards, inefficiencies or errors.</li> <li>• Impacts on subsequent activities.</li> <li>• Compliance or non-compliance with management intentions.</li> </ul>	<ul style="list-style-type: none"> <li>• Workarounds as necessary activities in everyday life.</li> <li>• Workarounds as creative acts.</li> <li>• Workarounds as sources of future improvements.</li> <li>• Workarounds as quick fixes that don't go away.</li> <li>• Workarounds as add-ons, shadow systems, feral systems.</li> <li>• Workarounds as inefficiencies or hazards.</li> <li>• Workarounds as a means for maintaining appearances.</li> <li>• Workarounds as resistance.</li> <li>• Workarounds as distortions or subterfuge.</li> </ul>

## 3 Theory of Workarounds

The theory of workarounds combines a process theory [16] (cited by [17]), with aspects of an influence diagram, thereby identifying steps in producing a workaround along with key factors that determine whether and how those steps will be undertaken. It attempts to incorporate phenomena related to workaround design and execution that have been studied by many authors. For example,

- Many workarounds can be viewed as a type of improvisation or bricolage.

- Workarounds are often viewed as exception handling and/or sanctioned or unsanctioned deviations from organizational routines, processes, and methods.
- Workarounds of obstacles, exceptions, and cumbersome processes often occur when performing articulation work, the often invisible background work that enables execution of steps in formal processes and other work.
- Many workarounds occur because technology that is used does not fit realities and contingencies of day-to-day work. Work system participants often see a need for workarounds in order to achieve goals related to efficiency, output, and responsiveness to customer needs.
- Workaround creation is a secondary design process, "where functions and content emerge during interaction, modification, and embodiment of the system in use."
- Agency theory provides many concepts that are relevant to workarounds, e.g., contracts between principals and agents, incentives, alignment or misalignment of goals, moral hazard, adverse selection, and information asymmetry.
- Reward systems that align enterprise and personal interests decrease the likelihood that inappropriate workarounds will be considered. The quality of control systems affects the likelihood that opportunistic workarounds will be noticed.

As represented in Figure 1, the theory of workarounds identifies steps in designing and executing workarounds along with common factors that affect perceived needs for workarounds and decisions about which workarounds will be designed and executed. It encompasses the descriptions of workarounds that were found in 300+ articles in the literature review mentioned earlier, ranging from small, localized workarounds that are forgotten quickly through software add-ons, or shadow systems designed to address work flow or software shortcomings over long time spans. It also covers all of the goals and perspectives on workarounds that were listed in Table 1.

Italicized terms on the left side of Figure 1 identify generic steps in perceiving the need for a workaround and then creating it. The sequence reflects a rationalist view in which work system participants create workarounds by identifying obstacles and deciding what to do about them. The theory combines ideas from the theory of planned behavior [18], improvisation and bricolage [19, 20], and agency theory [21].

The factors included in Figure 1 have significant impact in some situations and minimal impact in others because the theory spans a wide range of situations. For example, monitoring systems and ethical considerations usually are more important for workarounds that affect activities, information, or results elsewhere and usually are less unimportant for workarounds of temporary, local conditions that have no impact elsewhere. The theory addresses a different scope than agency theory even though some agency theory issues are relevant in some cases, such as moral hazard, information asymmetry, and the cost of monitoring. Where agency theory focuses on establishing mutually beneficial contracts between agents and principals, the theory of workarounds focuses on whether a workaround might be appropriate, and if so, which possible workaround to pursue. Further details are covered in [1].

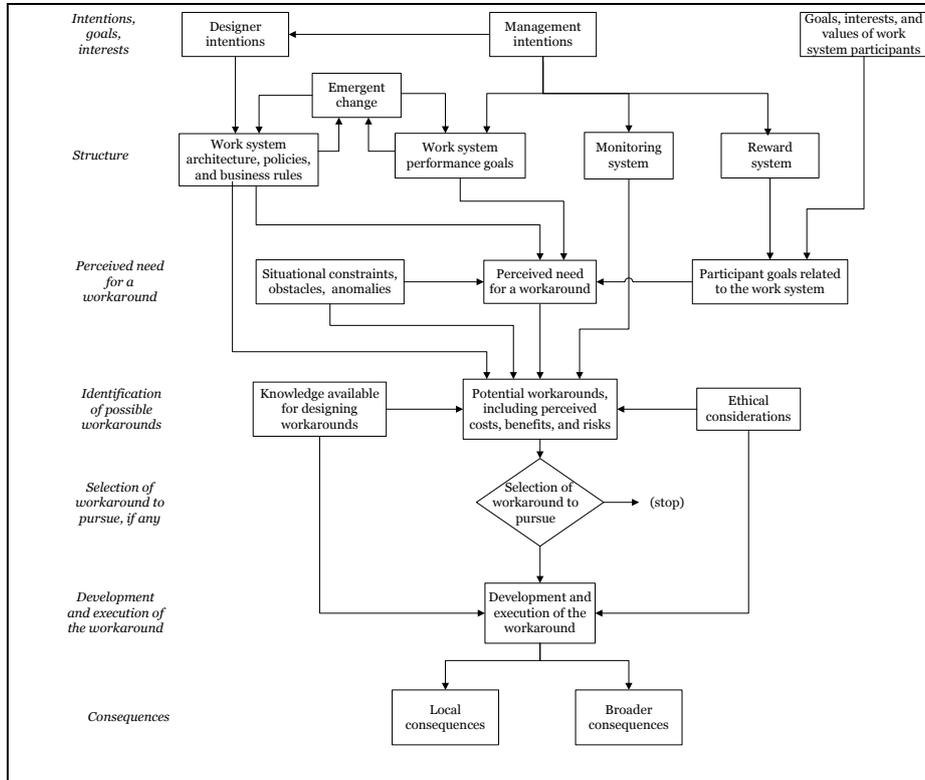


Fig. 1. Theory of workarounds [1]

#### 4 A Workaround Design System

The proposed WDS is a temporary work system whose participants use an interactive workaround design tool (WDT) to design workarounds related to a proposed or existing work system in an organization. Potential applications and benefits of a WDS were mentioned at the outset, e.g., anticipation of foreseeable workarounds, insight for managers and system designers, training and assistance for work system participants, and more effective user participation in analysis and design efforts.

**Design assumptions for a WDS.** Table 2 shows important differences between assumptions underlying the proposed WDS and assumptions for textbook descriptions of systems analysis and design. WDS challenges the assumption that systems in organizations will operate as designed or intended. It challenges the assumption that business processes and information systems represent best practices that remain appropriate even as the surrounding context changes over time and as occasional exceptions and obstacles prove awkward or insurmountable with established practices. It also challenges the assumption that designers are capable of designing processes and

related software that will encompass every possible situation that the work system will encounter. Overall, considering only best case assumptions and most likely cases is myopic and increases the probability of surprise responses to conditions that could have been anticipated.

**Table 2.** Comparing assumptions for a WDS versus Typical Systems Analysis and Design

<i>Topic</i>	<i>Assumptions for typical analysis and design methods</i>	<i>Assumptions for a workaround design system</i>
Unit of analysis	Information system or IT artifact	IT-reliant work system
Usage of hardware and software	Hardware and software will be used as specified in requirements	Hardware and software may be used as specified in requirements if requirements exist (which may not apply for organizational routines that evolved over time)
Responsibility of work system participants	Get work done using the prescribed methods	Get done work using prescribed methods when practical and applying workarounds if appropriate
Nature of official business process	The official business process represents best practices. It is the right way to perform the task.	The official business process was designed or evolved at some time in the past. It may not describe current practices and may or may not represent best practices.
Expectations about compliance	Work system participants will comply with business processes.	Work system participants may or may not comply with official business processes.
Nature of requirements	Requirements are rational and are based on management goals	Requirements may or may not exist because the current system may be the result of emergent change. Any existing requirements may not be appropriate, especially when exceptions occur.
Alignment of goals and incentives	Organization's goals are aligned with participants' incentives	Goals of the organization may or may not be aligned with incentives of work system participants.
Expected mastery and knowledge levels	Work system participants know how to do the job.	Work system participants know how to do the job and also know enough to create appropriate workarounds when they encounter obstacles.
View of workarounds	Workarounds are inappropriate. Work system participants should do their work consistent with the design of the work system.	Workarounds are appropriate when participants encounter exception conditions or obstacles, except when negative consequences would occur or when explicitly prohibited for understandable reasons.

**Structure of a WDS.** The structure of a WDS is based on the rationalist assumptions built into Figure 1. Table 3 uses the format of a work system snapshot [14] to summarize a WDS as a work system that uses a WDT. WDS participants are called designers to avoid confusion with participants in the work system in which the workaround will occur. Notice that the proposed WDT is included as a technology that is used within the WDS. The following discussion focuses on anticipating foreseeable obstacles and workarounds for a proposed work system that contains or uses an information system. Other uses such as producing a workaround to a currently operational work system would call for changes in some of the steps.

**Table 3.** Summary of a WDS in the Format of a Work System Snapshot

Customers		Products/ Services	
<ul style="list-style-type: none"> <li>Managers and designers who attain insights</li> <li>Trainers who use plausible workarounds in training</li> <li>Work system participants who use insights and documented information generated by the WDS</li> </ul>		<ul style="list-style-type: none"> <li>List of anticipated exceptions or obstacles and plausible workarounds, if any</li> <li>Documentation, suggestions, and warnings for each plausible workaround that is identified</li> </ul>	
Major Processes or Activities			
<ul style="list-style-type: none"> <li>Designers identify the work system in which workarounds might occur.</li> <li>Designers summarize the work system using a work system snapshot</li> <li>Designers identify foreseeable exceptions or obstacles that call for a workaround.</li> <li>Designers identify plausible workarounds for specific exceptions or obstacles.</li> <li>Designers evaluate plausible workarounds.</li> <li>Designers decide how to adjust the design, if necessary.</li> </ul>			
Participants	Information	Technologies	
<ul style="list-style-type: none"> <li>Designers of workarounds, who may be managers, work system designers, or participants in the affected work system</li> </ul>	<ul style="list-style-type: none"> <li>Description of work system</li> <li>Description of exceptions, obstacles, other relevant factors</li> <li>Description of workarounds</li> </ul>	<ul style="list-style-type: none"> <li>Workaround design tool (WDT)</li> </ul>	

Each of the major processes or activities in Table 3 calls for a bit of elaboration.

**Identify the work system.** Designers name the work system using a verb phrase (e.g., invoicing for construction work, answering customer queries, producing month-end financial statements, finding and fixing bugs in operational software).

**Summarize the work system using a work system snapshot.** Identifying possible workarounds requires a more detailed description of a work system than just a verb phrase identifying the work system. The format of Table 3 shows that a work system snapshot summarizes a work system by identifying customers, product/services, processes and activities, participants, information, and technologies on no more than one page. Work system snapshots have been used by many hundreds of MBA and Executive MBA students [14, 22]. The proposed WDT would have a method for entering the work system snapshot or would import it from existing documentation of the proposed work system.

**Identify foreseeable exceptions or obstacles that might call for a workaround.** This involves naming possible obstacles or exceptions that a proposed work system might encounter. To facilitate that task, a computerized version of Table 4, a table of common pitfalls and stumbling blocks for work systems, could be presented to suggest common obstacles that might be considered.

**Table 4.** Common Stumbling Blocks and Risk Factors (Alter, 2006, p. 65), abbreviated

Customers		Product/Services	
<ul style="list-style-type: none"> <li>• Unrealistic expectations</li> <li>• Unmet customer needs or concerns</li> <li>• Customer segments with contradictory needs</li> <li>• Unsatisfying customer experience</li> <li>• Lack of customers or customer interest</li> </ul>		<ul style="list-style-type: none"> <li>• Unfamiliar products or service</li> <li>• Products/ services are difficult to use</li> <li>• High cost of ownership</li> <li>• Incompatibility with other aspects of the customer’s work environment</li> </ul>	
Process and Activities			
<ul style="list-style-type: none"> <li>• Inadequate resources or capacity</li> <li>• Inadequate quality controls</li> <li>• Uncertainty about work methods</li> <li>• Excessive variability in work practices</li> <li>• Over-structured work practices</li> <li>• Excessive interruptions</li> <li>• Excessive complexity</li> <li>• Inadequate security</li> </ul>		<ul style="list-style-type: none"> <li>• Omission of important functions</li> <li>• Built-in delays</li> <li>• Unnecessary hand-offs, authorizations</li> <li>• Steps that don’t add value</li> <li>• Unnecessary constraints</li> <li>• Low value variations</li> <li>• Inadequate scheduling of work</li> <li>• Large fluctuations in workload</li> </ul>	
Participants		Information	
• (entries omitted)		• (entries omitted)	
Technologies		• (entries omitted)	
Infrastructure		• (entries omitted)	
Environment		• (entries omitted)	
Strategies		• (entries omitted)	
Work System as a Whole		• (entries omitted)	

Table 4 is organized around the six elements included in a work system snapshot plus the three other elements of the work system framework and “work system as a whole.” Due to page limitations entries for only three cells are shown, but these suffice to illustrate the content of this type of table. A version of Table 4 was proposed as a sort of negative design space consisting of things to be avoided in a new or existing work system [23]. Other versions of this type of table can be developed. For example, many of 228 risk factors in a survey of IS risk [24] might be included in a different version. Versions could be developed for specific types of situations, such as purchasing or manufacturing systems, by interviewing people who perform or manage that type of work in different organizations and asking them about the types of exceptions and obstacles that are encountered frequently.

**Identify plausible workarounds related to specific exceptions or obstacles.** A similarly formatted table [25, p. 8] could display common design “moves” that describe the form of a workaround. For example, a workaround might skip a business process step, might not conform to a business rule, might use different information, or might be performed by a substitute. The proposed WDT could help designers by dis-

playing a checklist-oriented version of that type of table to see whether each type of move had been considered. As with the previous step, different versions of this type of table could be developed for different generic situations, such as transaction processing systems or field service systems.

A second type of support for identifying foreseeable workarounds could take the form of an organized checklist of frequently used workarounds. That could come from empirical studies of workarounds and/or analysis of hundreds of examples in the literature. For example, [26,27] used an earlier version of the theory of workarounds in a multiple case study analysis of workarounds in healthcare, accounting, and automotive. That research identified generic types of workarounds that apply in many situations. Names of the following examples differ from their designations in [26,27], which have more complete explanations.

- download protected data (that should be protected, but is inconvenient to access when it is protected)
- provide a vague reason for accessing data (when a formal reason is required)
- post a password in view (when remembering passwords is too onerous)
- share one person's password (when using individual passwords repeatedly is too cumbersome)
- split transactions to avoid alarms (with a 5000 euro limit, convert 1 6000 euro transaction into 2 x 3000 euro transactions)
- use shell accounts (move money into accounts designed only to bypass audits)
- camouflage innovations as change requests (bypassing procedures related to justifying new functionality)
- convert special requests into standard tools (to minimize future effort in re-applying that functionality)
- exaggerate resource requests (ask for more funding than is needed because budgeting processes often cut initial requests)
- integrate functionality (combine separate requests into one functional deliverable)

A third type of checklist or script for identifying workarounds could be based on design characteristics that might be viewed as design dimensions. [25, p. 9] identifies characteristics that might be included in a script or checklist to help designers consider how a workaround might make the process more structured or less structured, more complex or less complex, more collaborative or less collaborative, and so on. A WDT might present those characteristics using low-to-high sliding scales that would be used by positioning the current system in relation to a characteristic and imagining what changes might occur if the slider moved substantially higher or lower.

**Evaluate plausible workarounds.** Although evaluating alternatives might be treated as a separate step in an idealized decision making process, evaluating plausible workarounds might also be merged, at least partially, with the previous step (identification of possible workarounds) if WDS participants believe that approach would minimize redundant effort. In either case, the designers would consider only workarounds that have a plausible rationale and that work system participants might actually consider. The WDT could facilitate this evaluation by providing a checklist or script based on issues implied by factors in Figure 1, such as the following:

- impacts on successfully performing the task despite the specific exception or obstacle that would prompt consideration of the workaround
- likelihood that the workaround will be noticed by management
- impacts on the work system participants, including benefits and problems for them
- impacts downstream, i.e., positive or negative consequences for the organization

## Discussion and Conclusion

This paper proposes a theory-based workaround design system (WDS) that includes a workaround design tool (WDT). This paper illustrated how existing knowledge related to common problems and issues, design moves, and work system characteristics could be incorporated into a WDT in the form of checklists, scripts, or sliders. Extensions for specific types of situations would display selected, domain-specific examples that occurred in the past, such as common ways to work around log-on procedures, authorizations, and controls in service and order entry systems.

WDS provides a perspective for looking at many topics and issues in new and different ways. It provides a system and tool for making accumulated knowledge about workarounds visible and useful for managers, designers, and trainers. It augments established pedagogy, practice, and research in systems analysis and design by outlining an approach for exploring implications of the common assumption that the system being designed actually will operate as designed. It frames inquiries related to workarounds that occurred or that did not occur, thereby bringing a perspective for exploring topics and issues related to adaptations, workarounds, and emergent change in organizations. A WDS has potential value in many areas such as the following.

**Value to managers.** A WDS could help in anticipating workarounds that might occur, thereby leading to tactics and strategies for facilitating appropriate workarounds and preventing inappropriate workarounds. It might help managers attain more realistic views of capabilities and limitations of computerized systems that try to embody and enforce best practices. It might support better communication about what is expected, both in terms of following rules and in terms of behavioral discretion.

**Value to system designers and developers.** A WDS would augment established systems analysis and design methods by providing a practical way to deal with workarounds, a topic that does not appear in the glossaries of most systems analysis and design textbooks. A WDS would complement established methods that generate UML or BPMN documentation of system structure and operation. The WDS would explore possible uses of workarounds to address practical limits of any particular idealized view of how work should be performed. Especially useful applications of WDS might occur during ERP configuration processes, which often encounter misfits between the situational needs within a department and the options offered by the ERP software. Like many formal methods and tools, WDS and WDT might be especially useful for less sophisticated designers, implementers, and users who had not yet honed their ability to anticipate, design, and evaluate possible workarounds.

**Value to trainers and trainees.** After initial training on a work system's basic structure and operation and the related software, trainees could use WDS and WDT to

suggest possible workarounds and to evaluate possible consequences for themselves, for their work groups, and for others in the organization. That level of engagement might generate deeper learning than current training approaches. It might overcome some of the common inadequacies of training on ERP and other complex software.

**Value to business students and their instructors.** Current introductory courses often contain hands-on exercises using tools such as transaction processing software, databases, spreadsheets, and search engines. While that hands-on experience is useful, most of it is about details and concepts of a specific information system or software tool. It is not about more challenging questions such as how to propose and evaluate workarounds when confronted with anomalies that call for workarounds. Use of a WDS in classroom settings could generate a more creative atmosphere in IS courses and could lead to deeper understandings that help students become more productive employees. Classroom applications could use crowd sourcing and gamification approaches to make the entire experience more engaging.

**Value for future development of specific work systems.** Use of WDS and WDT might help in setting paths for future improvements of IT-enabled work systems. This possibility is consistent with research concluding that workarounds often are the springboard for future improvements (e.g., [28, 29]).

**Supplementing traditional approaches and assumptions.** The idea of WDS reflects a distinctly unconventional stance toward systems analysis, design, and development. The system is an IT-enabled work system, not a technical artifact. Regardless of how well the initial requirements capture management intentions and “best practices,” the system-in-operation probably will deviate, at least occasionally, due to workarounds when unanticipated obstacles and contingencies arise. Thoughtful design of work systems should find ways to include workarounds that might occur, should help work system participants and their managers understand likely rationales for those workarounds, and should help them recognize positive and negative consequences of the types of workarounds that can be foreseen. This paper suggested the idea of WDS, explained its theoretical basis, and explained how concepts can be built into tools that support a WDS. The next step is to produce and test working prototypes to understand how these ideas can be used effectively in practice.

## 5 References

1. Alter, S.: Theory of Workarounds, *Communications of the Association for Information Systems*, 34, 1041-1066 (2014)
2. Feldman, M., Pentland, B.T.: Reconceptualizing Organizational Routines as a Source of Flexibility and Change, *Administrative Science Quarterly*, 48, 94-118 (2003)
3. Orlikowski, W. J.: Improvising organizational transformation over time: A situated change perspective. *Information Systems Research*, 7, 63-92 (1996)
4. Lyytinen, K., Newman, M.: Explaining information systems change: a punctuated socio-technical change model, *European Journal of Information Systems*, 17, 589-613 (2008)
5. vom Brocke, J., Rosemann, M.: *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems* (Vol. 1), Berlin: Springer (2010)

6. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Berlin-Heidelberg: Springer (2012)
7. Dumas, M. van der Aalst, W. M, ter Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software through Process Technology. Wiley (2005)
8. van der Aalst, W. M. P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, Berlin: Springer (2011)
9. Ploesser, K., Recker, J., Rosemann, M.: Challenges in the Context-Aware Management of Business Processes: A Multiple Case Study, Proceedings of ECIS, (2011)
10. Germonprez, M., Hovorka, D., Gal, U.: Secondary design: A case of behavioral design science research, Journal of the Association for Information Systems, 12, 662-683 (2011)
11. LeClair, C., Miers, D.: Dynamic Case Management: Definitely Not Your Dad's Old-School Workflow/Imaging System, Forrester Research, Inc., (2011)
12. Bartels, A., Moore, C.: Smart Process Applications Fill a Big Business Gap, Forrester Research, Inc., (2012)
13. Harel, D., Marron, A., Weiss, G.: Behavioral programming, Communications of the ACM, 55, 90-100 (2012)
14. Alter, S.: Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future, Journal of the Association for Information Systems, 14, 72-121 (2013)
15. Alter, S.: Defining Information Systems as Work Systems: Implications for the IS Field, European Journal of Information Systems, 17, 448-469 (2008)
16. Mohr, L. B.: Explaining organizational behavior San Francisco: Jossey-Bass (1982)
17. Markus, M. L., Robey, D.: Information technology and organizational change: causal structure in theory and research, Management science, 34, 583-598 (1988)
18. Ajzen, I.: The theory of planned behavior, Organizational Behavior and Human Decision Processes, 50, 179-211 (1991)
19. Orlikowski, W. J.: Improvising organizational transformation over time: A situated change perspective. Information Systems Research, 7, 63-92 (1996)
20. Baker, T., Nelson, R. E.: Creating Something from Nothing: Resource Construction through Entrepreneurial Bricolage, Administrative Science Quarterly, 50, 329-366 (2005)
21. Eisenhardt, M, K.: Agency theory: An assessment and review, Academy of Management Review, 14, 57-74 (1989)
22. Truex, D., Alter, S., Long, C. Systems Analysis for Everyone Else: Empowering Business Professionals through a Systems Analysis Method that Fits their Needs, Proceedings of ECIS (2010)
23. Alter, S.: The Work System Method: Connecting People, Processes, and IT for Business Results, Larkspur, CA: Work System Press (2006)
24. Sherer, S., Alter, S.: Information System Risks and Risk Factors: Are They Mostly about Information Systems? Communications of the Association for Information Systems, 14, 29-64 (2004)
25. Alter, S.: Design Spaces for Sociotechnical Systems, Proceedings of ECIS. (2010)
26. Röder, N., Wiesche, M., Schermann, M.: A Situational Perspective on Workarounds in IT-enabled Business Processes: A Multiple Case Study, Proceedings of AMCIS (2014a)
27. Röder, N., Wiesche, M., Schermann, M.: Why Managers Tolerate Workarounds – The Role of Information Systems, Proceedings of ECIS, (2014b)
28. Beckman, S. L., Barry, M.: Innovation as a learning process: Embedding design thinking, California Management Review, 50, 25-56 (2007)
29. Safadi, H., Faraj, S.: The Role of Workarounds during an Open Source Electronic Medical Record System Implementation, Proceedings of ICIS (2010)