Project Collaboration, not just User Participation

Steven Alter
University of San Francisco, alter@usfca.edu

Recommended Citation

Alter, Steven, "Project Collaboration, not just User Participation" (2009). *AMCIS 2009 Proceedings*. 658.
https://aisel.aisnet.org/amcis2009/658

Project Collaboration, not just User Participation

Steven Alter
University of San Francisco
alter@usfca.edu

ABSTRACT

Markus and Mao (2004) provide a thorough and insightful review of the user participation literature. They identify confusions and other issues related to situational factors that affect the importance and feasibility of user participation. They propose a new user participation framework and nine related propositions.

This paper proposes a more radical approach of focusing on business/IT collaboration in projects related to IT-reliant work systems. It argues that "project collaboration" should replace "user participation" as the core of the discourse because it better expresses major challenges and clarifies the nature of project success. The concept of project collaboration applies to the larger project of improving an IT-reliant work system, rather than the more limited project of developing and implementing software/ hardware configurations. In conjunction with other Figures presented here, this paper's project collaboration framework can be used in research, teaching, and guiding projects in practice.

Keywords

User participation, system life cycle, project management, project collaboration, work system, work system framework

INTRODUCTION

Markus and Mao (2004) note that "the participation of users in system development and its role in IS success have been core topics in IS research since the 1960s." They explore the literature related to user participation and identify confusions and other issues. Their Abstract calls for "an updated theory of user participation that is robust enough to accommodate changing IS practice" (p. 514)

Their critique notes that the existing literature leaves "important conceptual issues unresolved, particularly when viewed in the light of developments such as ERP system implementations, outsourcing, and new software development approaches such as contextual design." (p. 514) They cite examples showing "that in many 'IS' projects today, it is difficult to differentiate the system from the other aspects of an IT-based business intervention, such as process redesign, physical layouts of the workplace, changes in job design and compensation, or development of IT infrastructure." They note that many authors have argued that IT investments deliver the greatest business value when they are combined with complementary changes." (p. 526) Despite noting many issues that extend beyond the traditional topic of software development, Markus and Mao seem to follow the IS field's tradition of treating the system under development as a software/hardware configuration. Also, they seem to assume that the central topic is the process and impact of user participation when IT specialists create technical artifacts.

This paper proposes that a fundamentally different frame of reference might be more beneficial in today's environment. It assumes that user participation in the creation or improvement of a technical artifact is not the main headline, and that the idea of "user participation" is fundamentally limited and possibly outdated. Looking at system development and implementation from a business viewpoint, the systems of interest are not software/hardware configurations that users use, but rather, systems of doing work in organizations. Instead of producing technical artifacts, the goal is to improve business performance by creating new work systems or improving the performance of existing work systems.

This paper is based on three basic premises that are expressed in the project collaboration framework in Figure 1.

- The fundamental goal of acquiring and implementing application software is to create or improve the performance of work systems in organizations, regardless of whether application software is purchased, developed from scratch, or developed by improving existing software.

- The success of software development and implementation should be evaluated from a business perspective based on two types of goals: performance goals for development and implementation processes and performance goals for the target work system. Whether or not technical artifacts satisfy various stakeholders is an important but secondary consideration.

- The topic of "project collaboration" is more important than "user participation," which is a misnomer in relation to many of today's system projects. Project collaboration comes closer to expressing major challenges and revealing alternative paths that may or may not lead to successful projects.

Reframing some of Markus and Mao's observations, the project collaboration framework in Figure 1 expresses these premises in the following ways:

- It says that project results depend jointly on project and situational characteristics, the actual form and content of the project, and the extent of discrepancies between the actual and ideal form of project collaboration in this situation. The discrepancies can have positive or negative impacts on costs, risks, schedule performance, and completeness and other results.

- It says that the main results are process results (such as schedule, budget, and completeness of the development and implementation phases) and product results (how well the new or updated work system performs).

- It refers to collaboration but does not refer to users, user participation, or software usage. Those topics are all important, but the first order issues concern work system participants, collaboration between business and IT stakeholders, and improved operation of the work system.
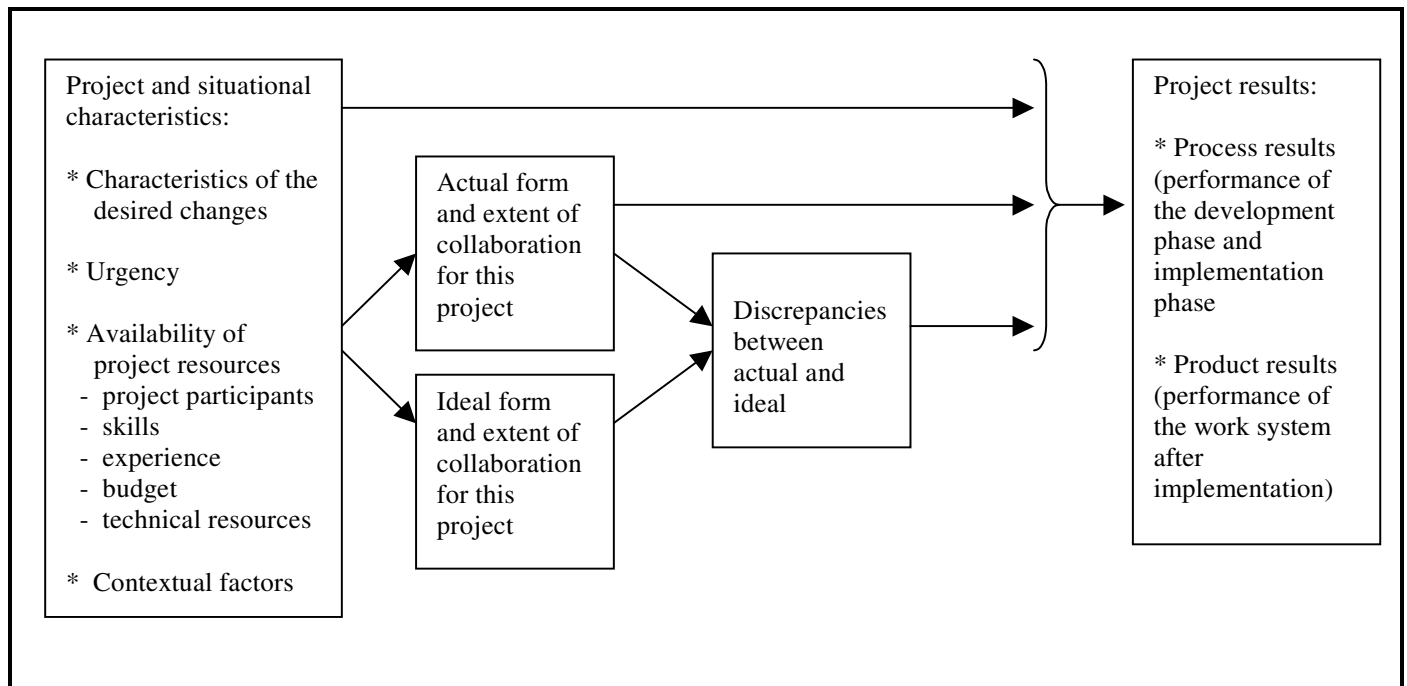
Figure 1: Project Collaboration Framework

This paper proceeds as follows: It defines work system and summarizes the work system framework and work system life cycle model, which have been explained in depth elsewhere (Alter, 2006, 2008). It explains project collaboration by extending the work system life cycle model into more detailed models (Figures 5 and 6) that identify generic steps within phases and summarize a typical division of roles between business and IT professionals. By doing so, it provides a flexible process model that addresses many of Markus and Mao's observations, criticisms, and insights while also clarifying and simplifying the whole discussion of user participation. Next, it identifies a series of basic concepts within the traditional user participation paradigm and shows how switching to a project collaboration paradigm provides greater clarity and realism. It concludes by summarizing how the project collaboration approach promotes a broader, more inclusive view of participation.

Given its length limitations, this paper does not repeat the citations provided by Markus and Mao (2004).

WORK SYSTEMS

A work system is a system in which human participants and/or machines perform work using information, technology, and other resources to produce products and/or services for internal or external customers. Typical business organizations contain work systems that procure materials from suppliers, produce products, deliver products to customers, find customers, create financial reports, hire employees, coordinate work across departments, and perform many other functions.

The work system framework (Figure 2) is the basis for describing and analyzing a system in an organization. This is a static model of a work system's components at a point in time. The work system life cycle (WSLC) model (Figure 3) describes how work systems change over time through a combination of planned change (explicit projects with initiation, development, and implementation phases) and unplanned change (incremental adaptations and experimentation). Figure 3 summarizes the WSLC's phases and iterative form. Figure 5 will identify generic steps within phases; Figure 6 will identify a typical division of roles between business and IT professionals in each phase. All four of these figures are designed to be relevant and useful regardless of whether IT is important in the work system and regardless of whether organization change projects involve IT changes.

With a work system approach, the relevant system is a work system that produces products and services for internal or external customers. The relevant system is not application software or other tools that are "used." Rather, application software is part of the work system. Similarly, the relevant change model is the WSLC, not a project model such as various forms of the system development life cycle (SDLC) model, even though the terms development and implementation appear in both types of models.

Work systems in general vs. specific work systems. When viewed in general, information systems, projects, supply chains, e-commerce web sites, etc., are special cases of work systems. (Alter, 2008) When specific information systems and work systems are examined, one sees that some information systems (e.g., payroll systems) exist to perform complete business processes or functions, whereas others (e.g., those applying CRM and ERP software) exist to support work systems that may involve much more than just processing information.
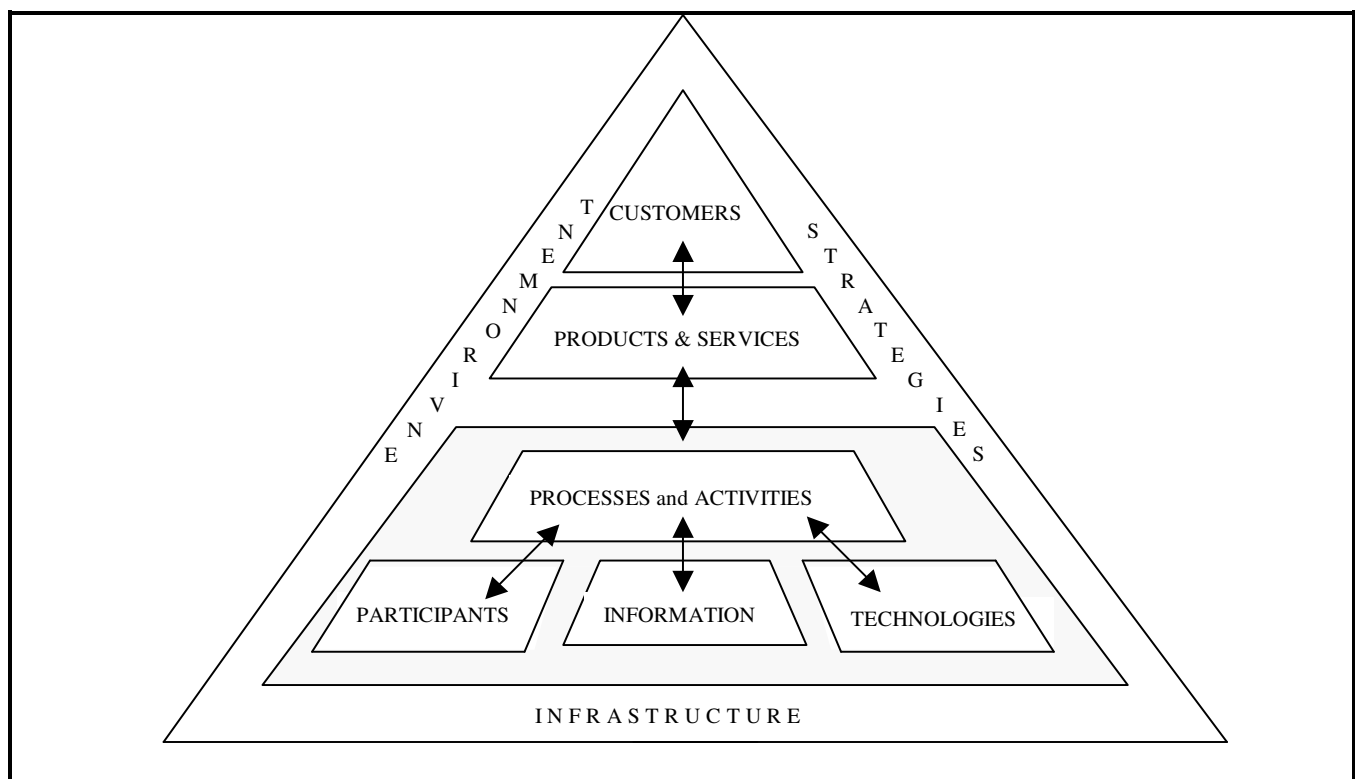

Figure 2: The Work System Framework. Source: Alter (2006), slightly updated
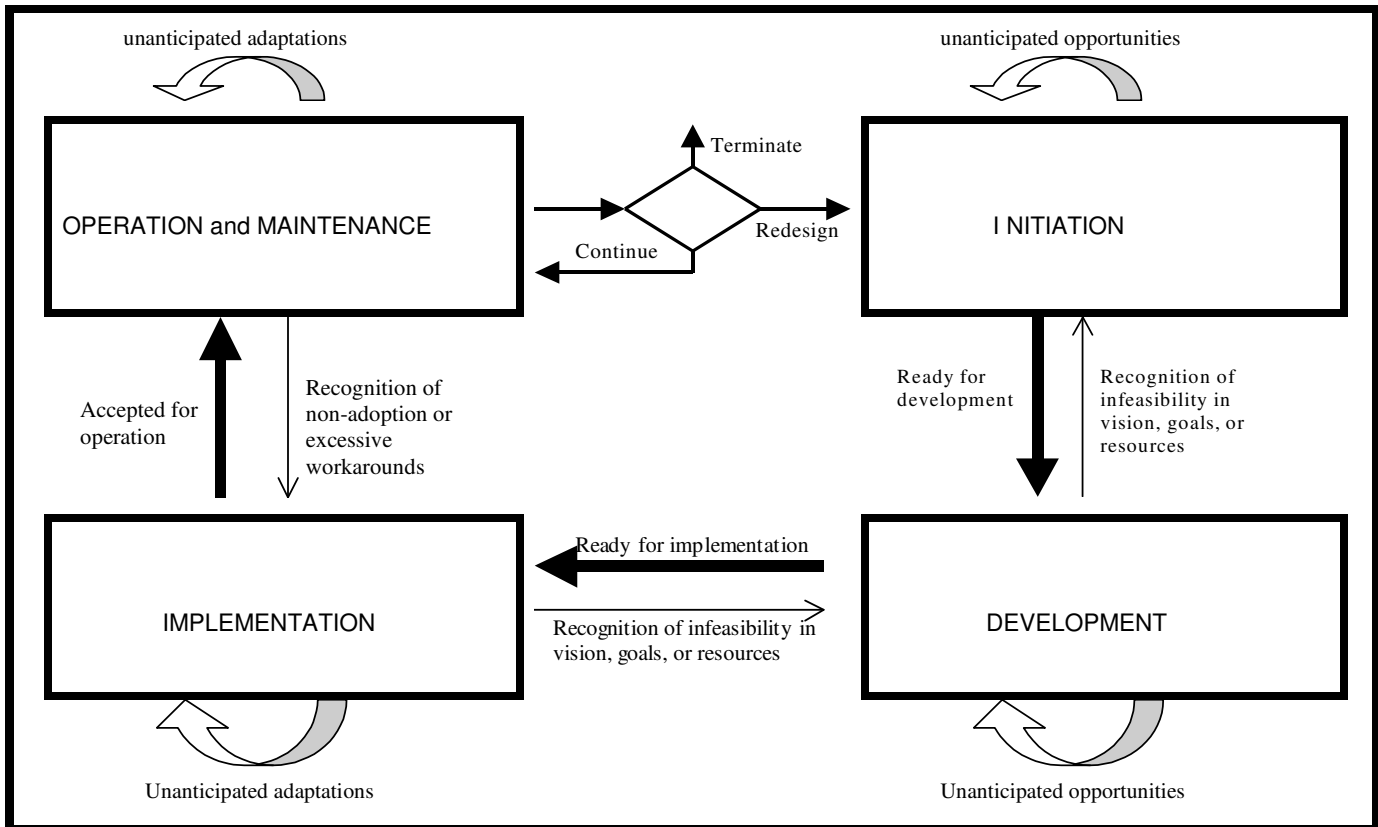
Figure 3: The Work System Life Cycle Model.   Source: Alter (2006)

Information systems and work systems they support. Most information systems overlap with work systems they support. (See Figure 4) The work system cannot operate well without the information system, but the work system may also include decision-making, communication, negotiations, physical activities, and other activities that are untouched by the information system or influenced by it indirectly.   Information systems and the work systems they support are often inextricably connected. Remove the information system and the work system can't operate. Ignore the work system and the information system is meaningless.

Work system success vs. information system success. Figure 4 represents an observer evaluating system effectiveness and success … but of which system? … the information system in isolation or the work system that the information system supports and of which the information system is an integral component? As represented by Figure 4, performance from a business viewpoint is generated by the work system, not by the information system that supports the work system. In contrast to common practice in the academic IS field, Figure 4 treats business performance as the primary outcome of interest and does not mention IT performance. The increasing degree of integration between (specific) information systems and the (specific) work systems they support makes it more difficult to evaluate the information system and its impact on work system success. IS success metrics that ignore the performance of the work system is being supported are increasingly less useful.

Activities and responsibilities in development and implementation. Two additional figures explain the work system life cycle model in more depth. Figure 5 identifies typical activities in all four phases. The development phase includes specification of planned details of the work system, not just the information system. Similarly, implementation includes conversion to the new or revised information system and work system. Thus, implementation of a work system entails much more than just installing a new information system.

Figure 6 identifies responsibilities in each phase that are typically shared between business and IT specialists, and other responsibilities that typically belong to one or the other. It shows that business responsibilities exist in each phase, even in development. By referring to shared responsibilities and business responsibilities, rather than user participation, the WSLC

establishes the expectation that something is wrong if participation responsibilities are not met. This is a starting point for identifying the discrepancies between ideal and actual that appear in the project collaboration framework in Figure 1.

Beyond the scope of this paper, it is possible to provide specialized versions of Figures 5 and 6 that focus on work system life cycle steps when specific IS development approaches are used, such as waterfall development, agile development, prototype-based development, acquisition and configuration of commercial application software, and end-user computing. Going that direction potentially provides normative models detailed enough to guide both practice and research about direct and indirect impacts of various types, amounts, and qualitative aspects of collaboration between business and IT professionals in a variety of fundamentally different development and implementation situations. For both practice and research, the resulting insights could be far more valuable than trying to pursue broad-brush questions about the impact of user participation in general.
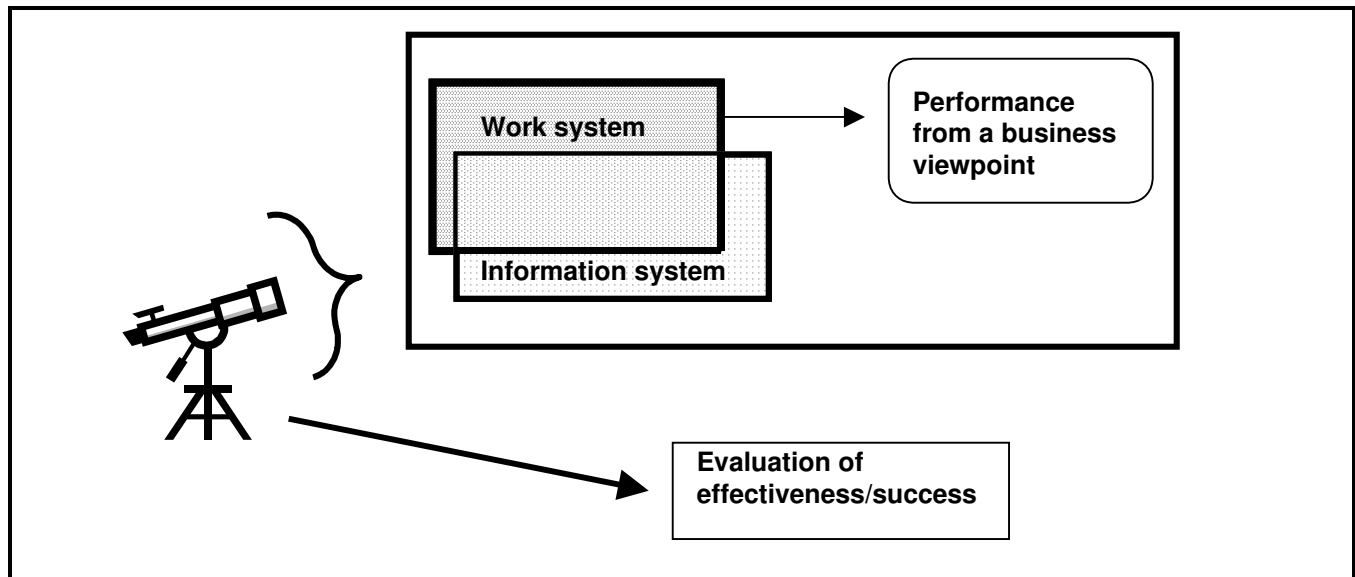

Figure 4: Cause of Increasing Difficulty in Evaluating Information System Success

THE PROJECT COLLABORATION PARADIGM VS. THE USER PARTICIPATION PARADIGM

Many of Markus and Mao's observations and examples fit with concepts in the project collaboration framework (Figure 1) and with other concepts expressed or implied by Figures 3, 5, and 6. Those ideas and concepts include the following:

- The dynamics and impacts of participation call for attention to specific situational variables, such as the type, size, and scope of the project.
- The impact of user participation may occur through different paths, including a higher level of buy-in from stakeholders, better information about requirements, and better relationships between IS developers and stakeholders.
- The form of user participation may be different in different types of situations.
- It is important to "to bridge three conceptual gaps" – "the gap between participants and affected parties who did not participate, the gap between developers knowledge of requirements and the quality of the solutions they produce, and the gap between functional outcomes and outcomes related to relationships and acceptance." (p. 524)

Ultimately, Markus and Mao try to shift the central issue of user participation studies from "how should system developers involve users?" to "how can change agents employ participation practices to increase the chances of success in varied IS development contexts?" (p. 524) This paper proposes a different reframing based on an explicit work system focus.
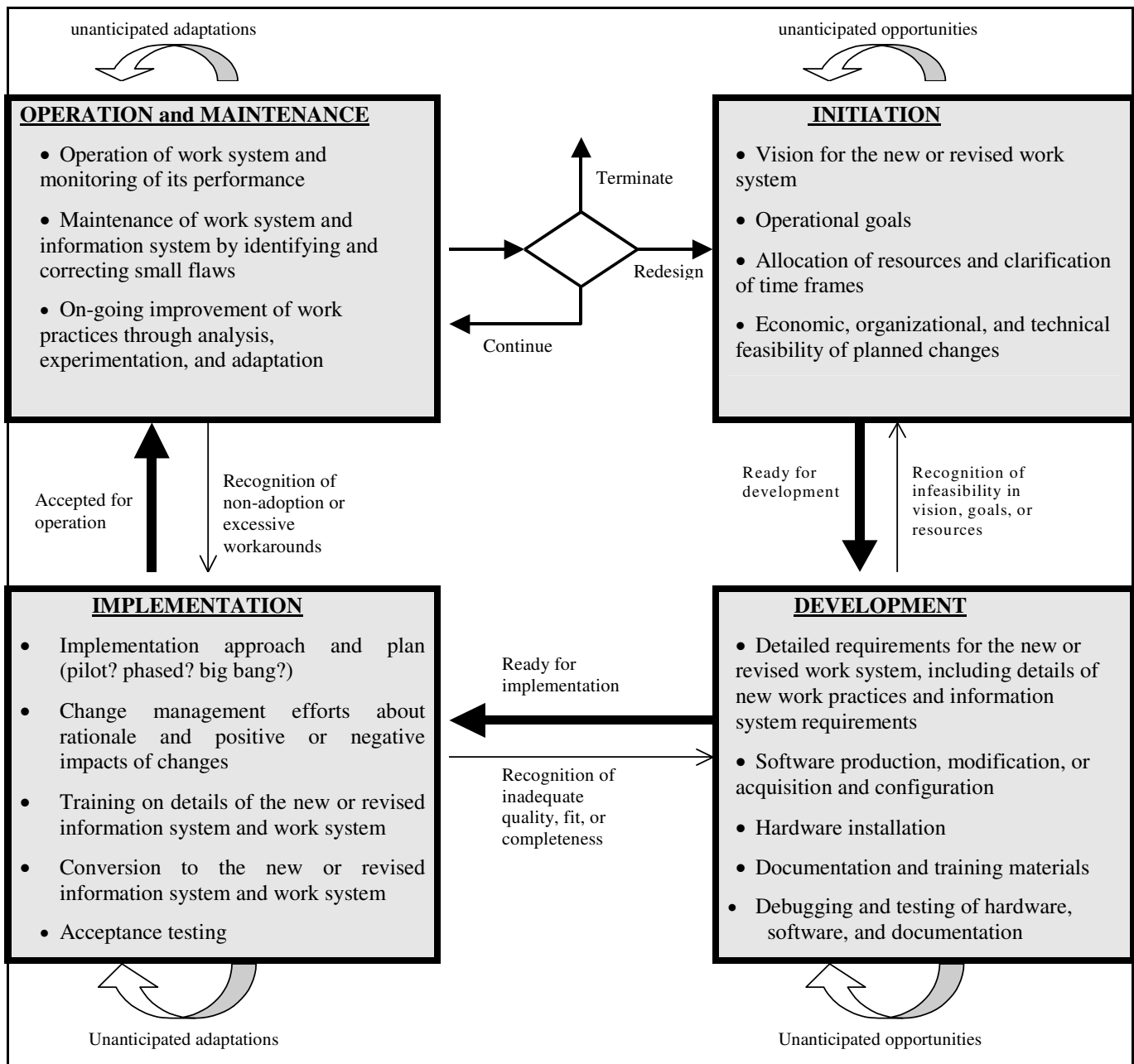
Figure 5: Typical activities in each phase of the work system life cycle. Source: Alter (2006)
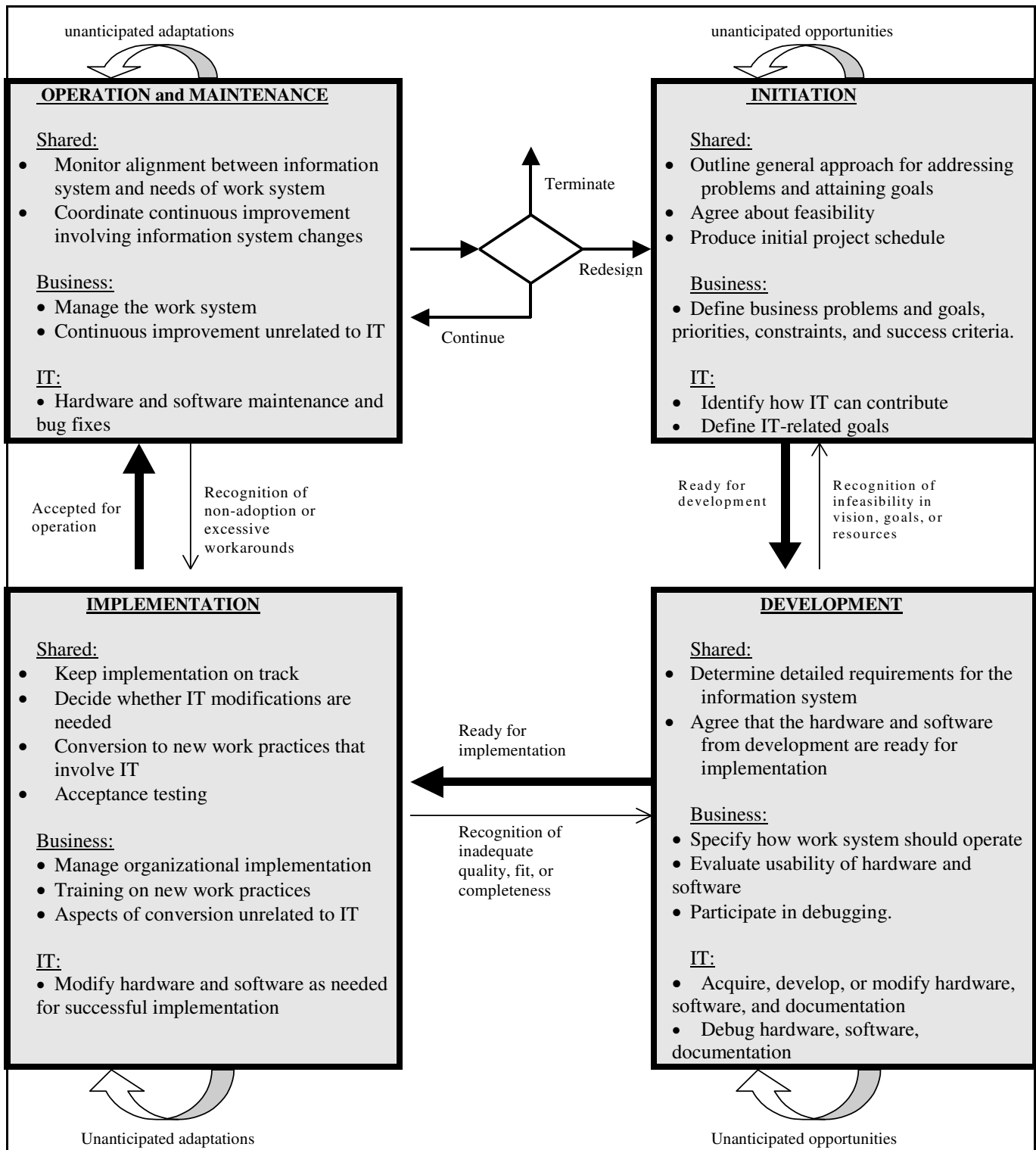
OPERATION and MAINTENANCE	INITIATION
OPERATION and MAINTENANCE Shared: • Monitor alignment between information system and needs of work system • Coordinate continuous improvement involving information system changes Business: • Manage the work system • Continuous improvement unrelated to IT IT: • Hardware and software maintenance and bug fixes	**INITIATION** Shared: • Outline general approach for addressing problems and attaining goals • Agree about feasibility • Produce initial project schedule Business: • Define business problems and goals, priorities, constraints, and success criteria. IT: • Identify how IT can contribute • Define IT-related goals

unanticipated adaptations

unanticipated opportunities

Terminate

Redesign

Continue

Accepted for operation

Recognition of non-adoption or excessive workarounds

Ready for development

Recognition of infeasibility in vision, goals, or resources

IMPLEMENTATION	DEVELOPMENT
IMPLEMENTATION Shared: • Keep implementation on track • Decide whether IT modifications are needed • Conversion to new work practices that involve IT • Acceptance testing Business: • Manage organizational implementation • Training on new work practices • Aspects of conversion unrelated to IT IT: • Modify hardware and software as needed for successful implementation	**DEVELOPMENT** Shared: • Determine detailed requirements for the information system • Agree that the hardware and software from development are ready for implementation Business: • Specify how work system should operate • Evaluate usability of hardware and software • Participate in debugging. IT: • Acquire, develop, or modify hardware, software, and documentation • Debug hardware, software, documentation

Ready for implementation

Recognition of inadequate quality, fit, or completeness

Unanticipated adaptations

Unanticipated opportunities

Figure 6: Typical Responsibilities across the Work System Life Cycle. Source: Alter (2006)

Cutting the Anchor and Moving to a Different Frame of Reference

Although many of Markus and Mao's ideas reflect some affinity with a work system approach, they still use the established terminology of user, user participation, systems as software/ hardware configurations and system development and implementation as development and implementation of software/ hardware configurations. This paper's reframing cuts the anchor of traditional terminology and attempts for greater clarity, generality, and usefulness in research and practice.

The reframing can be summarized as follows:
- The system being created or improved is an IT-reliant work system, not an IT system.
- The relevant life cycle models are various versions of the WSLC, not IT-focused SDLC models.
- For both research and practice, project collaboration is a more fruitful focus than user participation.
- The most meaningful success metrics are business metrics (not IT metrics) related to the product and process of development and implementation.

To explain immediate implications of this reframing, we distinguish between a traditional "user participation paradigm" and a new "project collaboration paradigm." Markus and Mao already provide important steps beyond the traditional user participation paradigm. Our reframing tries to extend their insights by switching the frame of reference from software/ hardware projects to work system projects.

This paper's length limitations call for explaining the reframing by presenting Figures 1 through 6 and by showing how the project collaboration paradigm (PCP) facilitates questioning or reinterpreting terms from the user participation paradigm (UPP). A longer paper would include a more extensive discussion of each distinction. The difference between PCP and UPP can be seen in how they treat terms such as system, complementary assets, user, development, implementation, and success.

The system in UPP is a software/ hardware configuration that is used by users. The surrounding context and resources are important, but the system being developed consists of software and hardware. The users are not part of the system; rather, they are users of the system.

- In PCP, the system is a work system (typically, but not necessarily IT-reliant) whose participants perform work to produce products and services for internal and/or external customers. Some important work system participants may not be technology users in their work system roles. A work system may be totally automated in some cases, such as an automated production line or automated software delivery system.

- As in much of the IS literature, Markus and Mao do not define the term *system* explicitly and seem to treat *system* as a rough synonym of "application software used through hardware devices." If *system* is meant to be synonymous with *application software*, it would be clearer to use the latter term rather than the more general term system that appears in many other contexts (e.g., systems of doing work in which IS/IT may or may not be a central focus, such as employee evaluation systems, customer service systems, manufacturing systems, delivery systems, and so on).

Complementary assets in UPP are organizational assets that are relevant to successful implementation of the software/ hardware configuration but are external to it. Such assets include business processes, job designs, and infrastructure.

- In PCP, business processes, incentive structures, interpersonal communication patterns, and workplace layouts are considered part of the work system. Aspects of the surrounding environment such as the organization's culture and policies are considered part of the environment in the work system framework. In other words, things might be considered complementary assets in UPP are treated as work system components or as factors that are directly relevant to the work system's operation and success.

- Although they treat complementary assets as externalities, Markus and Mao fully appreciate the importance of such factors. Specifically, they cite examples showing "that in many 'IS' projects today, it is difficult to differentiate the system from the other aspects of an IT-based business intervention, such as process redesign, physical layouts of the workplace, changes in job design and compensation, or development of IT infrastructure. Indeed, many authors have argued that IT investments deliver the greatest business value when they are combined with complementary changes." Markus and Mao suggest replacing "the concepts of system development success and system implementation success with the concepts of *solution* development success and *solution* implementation success, respectively, where solution

refs to a package of IT plus complementary assets." (p. 526) (The concepts solution development success and solution implementation success will be discussed in the concluding section.)

System usage in UPP is described in the same terms (usage, usage patterns, etc.) as usage of a tool.

- In PCP, system usage is a non-topic because the system is not a tool. Instead, the system is a system of performing work.

Relevant actors (people) in UPP start with users of software/hardware and include other stakeholders.

- In PCP, relevant actors include a range of stakeholders, starting with work system participants, some of whom are users of software/hardware.

- "User" is often a misnomer when applied to information systems. The term user applies to the usage of a thing, not inclusion within a thing. Thus, a user of information or of an automobile is not part of either. In contrast, accountants who produce financial statements are participants in information systems that produce financial statements; they are not users of those systems. Hence, IS "users" often are better described as work system participants who happen to use IT. Markus and Mao recognize this issue, saying, "traditional participation literature often appears to assume that the participants are the intended hands-on users. … Despite the commitment of participants who are hands-on users, the system may fail because of a lack of buy-in from the users' managers who did not participate." (p. 517) Their framework for participation theory does not mention users per se.

System quality in UPP is the quality of a software/hardware configuration, evaluated based on technical quality, usage of the software/ hardware, and fit with business requirements.

- In PCP, system quality is work system quality, evaluated based on whether the work system is capable of meeting performance goals while operating consistent with organizational guidelines and objectives. The multi-factorial dimensions of system quality include characteristics of all nine work system elements (Figure 2).

- The concept of system quality presents a double bind for user participation research. If the system is software/hardware, then system quality might seem independent of aspects of an IT-based intervention noted by Marcus and Mao, such as "process redesign, physical layouts of the workplace, changes in job design and compensation, or development of IT infrastructure." (p. 526) It seems questionable from a business viewpoint that a technically sound software/hardware configuration satisfying stated requirements is a high-quality system, regardless of whether the requirements were misguided and whether the system contributes to achieving desired or expected business results. If the system is basically a work system (process, people, information, technology, etc.) then system quality is only partially determined by software/hardware that is developed and implemented. Work system quality is often more directly related to other things, such as processes, work system participants, and whether its products and services truly satisfy customer needs and desires.

System success in UPP is evaluated based on usage of the software/ hardware and user acceptance.

- In PCP, system success is evaluated in terms of whether the work system is able to meet performance goals consistent with organizational guidelines and objectives. The multi-factorial dimensions of system success include characteristics of all nine work system elements.

The project in UPP is an IT project that creates or improves a software/ hardware configuration to satisfy business requirements.

- In PCP, the project is a business project that creates or improves a work system through activities involving changes in any work system element mentioned in the work system framework (Figure 2).

The analysts in a project in UPP are IT professionals who understand software and hardware and are trained to document processes and information requirements.

- In PCP, the analysts are business and/or IT professionals who can understand and analyze business situations. The relevant tools range from IT documentation methods through Six Sigma and other methods not usually associated with IT.

The project leader in UPP is typically an IT professional, because this is an IT project.

- In PCP, the project leader should be the business owner of the work system, or that individual's knowledgeable representative, possibly working as co-lead with an IT professional.

What is developed in UPP is the new or improved software/ hardware configuration, plus related documentation and training material.

- In PCP, development produces whatever resources are needed to implement the new or improved work system. These resources may include new or improved software, hardware, business process specifications, training material, documentation, and other computerized or non-computerized resources.

Major project challenges in UPP include getting the IT requirements right and testing and debugging the software/ hardware configuration. Getting the IT requirements right is a key goal of user participation.

- In PCP, major challenges include identifying work system changes that are economically, technically, and organizationally feasible, and converting from the previous work system to the new work system. Getting the IT requirements right is a subset of the major challenges.

The WSLC operation and maintenance phase through a UPP lens involves operating and maintaining software and hardware.

- In PCP, this phase involves operating the work system so that it meets or exceeds performance goals. It also involves incremental changes that allow the work system's processes and activities to be performed efficiently and effectively. Thus, this phase includes much more than just operating and maintaining software and hardware.

The WSLC initiation phase through a UPP lens involves defining business goals; outlining the approach for building, acquiring, or modifying hardware and software; and producing a preliminary project plan including timing and resource requirements.

- In PCP, the initiation phase is similar except that it focuses on the entire work system.

The WSLC development phase through a UPP lens involves specifying detailed requirements for the software and hardware; building, acquiring, or modifying the software and hardware; and testing and debugging the software and hardware.

- In PCP, the development phase is similar except that it focuses on the entire work system.

- In UPP *system development* is ambiguous because it is unclear whether the system is software or a new work system. Markus and Mao recognize this issue when discussing activities that are not specifically software-focused. They note that "user participation in development projects today can easily extend into business process redesign and IT infrastructure development." An interviewee says, "it's hard to know what part of these things are under this bucket you would call 'an IT initiative' and what is sort of 'a business process initiative.'" (p. 522) A PCP perspective would assume that a work system (business processes plus software plus other things) was being developed, not just software.

The WSLC implementation phase through a UPP lens involves communicating about the need for the new software/hardware; training users of the software/hardware; converting from the previous software/hardware to the new software/hardware; and verifying that the new software/hardware are acceptable.

- In PCP, the implementation phase is similar except that it focuses on the entire work system.

Development success in UPP involves product and process components. The product of development is evaluated based on whether features of the software/ hardware configuration satisfy business requirements. The process of development is evaluated based on whether development meets budget, schedule, and completeness goals.

- In PCP, development success is similar except that the product and process cover on the entire work system.

- Markus and Mao say that "system *development* success [is] a high quality process of system development (methodologies used, interactions and conflicts, progress against schedules and budgets) and/or a high quality outcome of system development, namely a project, a system, or an IT artifact." (p. 525)

- From a business viewpoint, success is about meeting or exceeding work system performance goals, regardless of whether the software development process meets budget, schedule, and completeness objectives. System success should be measured in terms of business success rather than IT project success.

Implementation success in UPP involves product and process components. The product of implementation is evaluated based on whether the conversion to the new software/ hardware resulted in a solid, usable software/ hardware environment and whether users use the software successfully. The process of implementation is evaluated based on whether the implementation process meets budget, schedule, and completeness goals.

- In PCP, implementation success is similar except that the product and process cover on the entire work system.

- Markus and Mao say that "system *implementation* success [is] a high quality process of preparing the target user community for use of the system (often called 'change management') and/or a high quality 'change' outcome, namely that the intended users (regardless of whether they participated in development) adopt the system, use it as expected, and/or use it with the desired effects." (p. 525)

- From a business viewpoint, success is about meeting or exceeding the performance goals of the work system(s) that are being improved, regardless of whether the implementation process seemed successful in terms of budget, schedule, and completeness objectives.

CONCLUSION: A BROADER, MORE INCLUSIVE VIEW OF PARTICIPATION

Markus and Mao provide a thorough and thoughtful review of the user participation literature, identify shortcomings, and argue that the today's IS context requires a more nuanced view of the topic. Instead updating and extending a paradigm focusing on users, usage, and user participation, we argue for a more radical approach of switching to a more general and useful paradigm focusing on business/IT collaboration in projects related to IT-reliant work systems.

The Figures summarized the essence of this approach, which starts with the work system framework (Figure 2), a highly summarized work system life cycle model (Figure 3), and the assumption that a business-oriented view of success focuses on work system performance (Figure 4). A more detailed representation of this starting point identifies typical steps within each phase (Figure 5) and characterizes typical division of roles between business and IT in each phase (Figure 6). The project collaboration framework (Figure 1) says that project results can be subdivided into process results and product results. Those results are affected directly by project and situational characteristics plus discrepancies between the ideal and actual form of collaboration for a project with those specific characteristics. Consistent with Markus and Mao's call for a more nuanced view of participation, this approach calls for creating and testing a series of specialized normative models for project collaboration in projects involving different development approaches, such as waterfall, agile, commercial software, and end user computing.

The previous section of this paper summarized how the terminology of a project collaboration paradigm would differ from the terminology of a traditional user participation paradigm. That is only a first step toward understanding the advantages of a project collaboration paradigm.

An additional step, beyond this paper's scope, would compare aspects of the new approach with each component of Markus and Mao's "new theoretical foundations." There would be strong agreement in areas such as their observations about the importance of considering development and implementation separately since the WSLC also treats development and

implementation as separate phases. Divergence would occur in other areas, such as their discussion of success, their terminology related to the term *solution*, and several of their nine propositions.

In particular, their use of the term *solution* in their concepts *solution development success* and *solution implementation success* would seem unnecessary and conceptually questionable within a WSLC-based viewpoint. They introduce the concept of solution ("a package of IT plus complementary assets" (p. 526) because successful application projects involve much more than IT. With a work system approach, most complementary assets are already included and must be dealt with as part of a work system project. Also, by the nature of Figures 5 and 6, collaboration about the new work system would start in the initiation phase and would continue in the development and implementation phases. In contrast to viewing a specific software deliverable or commercial software product as a solution, a completely definable "solution" would never occur until the new work system entered its operation and maintenance phase. Even then, it wouldn't be a solution because unplanned changes would continue. In general, emergent changes (unanticipated opportunities and adaptations) represented in Figures 3, 5, and 6 would probably occur throughout all four phases unless management somehow blocked emergent changes.

In summary, this paper has attempted to build on Markus and Mao's efforts by extending the discourse to a different level that explicitly focuses on work system projects rather than projects whose main goal is the development and implementation of software/hardware configurations. This paper provided a starting point that leads in a number of directions:

- Create normative models (specialized versions of Figures 5 and 6) that summarize work system development and implementation phases when various specific software development approaches are used.

- Validate those models empirically and through comparison with the literature.

- Identify implications of the validated models in relation to the existing user participation and project success literature.

- Develop a series of propositions, perhaps similar to the nine proposed by Markus and Mao, that apply to each type of situation.

- Test those propositions empirically to extend current knowledge about user participation and project collaboration.

REFERENCES

Alter, S. (2006) The Work System Method: Connecting People, Processes, and IT for Business Results, Larkspur, Ca: Work System Press.

Alter, S. (2008) "Defining Information Systems as Work Systems: Implications for the IS Field," European Journal of Information Systems, 17(5), pp. 448-469.

Markus, M.L. and Mao, J-Y. (2004) "Participation in Development and Implementation – Updating An Old, tired Concept for Today's IS Contexts," Journal of the Association for Information Systems, 5(11), pp. 514-544. http://aisel.aisnet.org/jais/vol5/iss11/14/