



# Bringing templates to life: overcoming obstacles to the organizational implementation of Agile methods

**Michael Durbin**

Saint Louis University  
143 Fairington Dr., Troy, Illinois, USA 62294  
USA  
[durbinmp@gmail.com](mailto:durbinmp@gmail.com)

*Viewpoint*

**Fred Niederman**

Management Department, Saint Louis University  
Chaifetz School of Business, 3764 Lindell Blvd., St. Louis, Missouri, USA 63108  
USA  
[fred.niederman@slu.edu](mailto:fred.niederman@slu.edu)

## **Abstract:**

Agile software development methods have become accepted as a template for organizations to create new products. Though generally viewed as an aid to productivity, there are a number of barriers to experiencing their full benefit. One such barrier pertains to the implementation of agile methods across the range of organizational levels from the use of tools to culture, norms, and policies creating the context within which projects are performed. This essay examines in detail the experiences of one expert at integrating agile technique, approach, and philosophy into the broader organizational setting. Numerous particular lessons and prescriptions result from this discussion. Turning around the grounded theory approach where numerous individuals are interrogated mildly in regard to a particular phenomenon, the discussion surfaced in this paper results from repeated interviews with one domain expert. Lessons and comments are organized into four sections: individual team member, organization, transitioning, and tools and techniques.

## **Keywords:**

agile methods; software development; project management; portfolio management; agile mindset; SAFe.

**DOI:** 10.12821/ijispm090301

**Manuscript received:** 19 August 2021

**Manuscript accepted:** 7 September 2021

## 1. Introduction

Many organizations have moved from solely using a traditional plan-based to either a mix of plan-based and agile approaches or solely to agile. As Birkinshaw [1] states: “Most Information Technology (IT) departments in large companies today are adopting agile techniques to some extent, although with varying degrees of success (p. 39)”. At the level of individual projects, it is widely held that agile methods provide net benefits relative to traditional Systems Development Life Cycle (SDLC) plan-based methods, and some evidence points to this (e.g., Lee and Xia [2]). However, converting the possibility of benefit from management of individual projects to fully realizing them may require more than shifting the practices of individual projects. However, for firms moving from a traditional to an agile organizational approach to IT projects, the transition is often difficult. As Hobbs and Petit [3] state:

*“During preliminary interviews prior to undertaking this research, the authors discovered several large organizations that had been experimenting with agile methods over a period of five years or more and that these organizations are struggling to scale from a few agile teams to an organization-wide implementation of agile methods (p. 3).”*

Where numerous templates exist, such as SAFe, for describing how agile methods at the project level may be integrated within organizational systems (technical and non-technical), the actual implementation and institutionalized is often not fully realized. The purpose of this discussion is to provide insights into the problems and potential solutions to shifting toward greater use of agile and acquisition of its benefits.

The writing of this viewpoint paper has been an educational process for both authors. During our investigation, an emerging theme running through the bulk of the transition-related problems and issues pertains to “agile mindset”. We will refer to this in regard to many of the specific observations and recommendations below. What we mean by this is an understanding and commitment at a deep level to the practices of agile methods, even when they seem unusual or non-intuitive, not only in the broad stroke but in the everyday details. This is a mindset that may come naturally to some people who see agile as an exciting alternative to traditional approaches from their first introduction, but, as it happens, such an attitudinal shift is certainly not universal and can impinge on many ways in the shift from a traditional to agile project management organization. We use the term “mindset” following Imran and Gregor [4] as they conceptualize “IT mindset” as referring to a set of cognitive filters, the product of an individual’s history developed through an iterative process that guides the interpretation of new information<sup>1</sup>. In this sense, the cognitive filters of one imbued in the traditional methodologies would vary from those of one committed to and comfortable with the agile approach. It is our contention that while difficult to measure, even to observe, the shift from the traditional to an agile mindset from top to bottom among those involved in implementing agile projects is critical. In some ways, our findings are the inverse of those of Conboy [5], who found that some highly touted agile practices like customer inclusion on the project team and standup meetings were not effective. On the other hand, this might be interpreted to reinforce how important it is to invoke sincere application of agile “mindset” even in agile techniques.

Every organization moving from an SDLC traditional approach to an agile one confronts challenges effectuating a successful conversion or transition Conboy [5]. Particularly for organizations strongly steeped in traditional development approaches, initiating, managing, and solidifying such transition is critical to extracting value<sup>2</sup> from the application of agile methods. Dingsøyr [6], for example, presents a set of 10 lessons in five categories that roughly align with the lessons we present in this paper. Where our lessons overlap, we attempt to present further detail and a broader range of managerial responses. Our discussion is organized in four categories – individual adoption, organizational adoption, transition, and tools and techniques, which enhances the set of questions raised for future research in Niederman et al. [7, p. 7]. Note we do not claim these observed principles to be the only ones that may be effective, nor

---

<sup>1</sup> This is one of a number of references to “mindset” as defined by Imran and Gregor [4], but we feel this one comes closest to describing how we use the term in this paper.

<sup>2</sup> We use the term “value” in the sense of enabling new user capabilities or reducing process complexity and the like, rather than a dollar value of finished components, as one would with earned value management programs.

that they will work in all circumstances. Rather we present them as a sort of “proof of concept” that they have worked (or have not worked) in at least one set of experiences and should be worthy of consideration for those confronted with similar issues and circumstances.

This essay is based on a practitioner-academic dialogue. It highlights the experiences and lessons learned by a particular highly successful project and portfolio manager who has assisted with the transition from traditional to agile development methods settling on a particular sort of hybrid approach<sup>3</sup>. This approach addresses project management concerns that are not addressed directly by approaches to agile (or SDLC) per se. It also acknowledges the likely benefits in some situations of shifting SDLC-oriented teams gradually toward agile methods by adding agile tactics such as standup meetings and Kanban activity lists as the team members become open to assimilating them.

The practitioner (first author) has 30 years of experience managing projects, portfolios and now introducing agile techniques to organizational project management orientation. The academic (second author) has studied the literature on traditional and agile methods, co-edited a special issue of Project Management Journal on moving agile technique from software development to other task areas, and taught systems analysis and design as well as project management to graduate and undergraduate students for two decades. As a result, the observations are based on the practitioner’s experiences filtered and extended by acquaintance with the academic literature. The goal is to bring this heuristic knowledge into a more formal setting for generalization and gathering support (or refutation) for these practices and observations.

On the surface, our research method can be viewed as reversing some of the tendencies of grounded theory. Where grounded theory compresses detailed observations into more general themes using successive rounds of coding to create broad theory by proposing how such themes are related, the approach of this study focuses on the detailed observations themselves, probing more deeply into their specifics and what they might mean for others. Where grounded theory, typically, interviews a range of individuals one time across a range of subjects to get overall views on topics of interest, this approach used multiple iterations of interviews to look in increasing detail at the issues raised. This approach mimics the agile method of iterations for developing artifacts (in this case lessons), observing and filling gaps between them, and examining not only the “what” leads to more conversion success but also the “how” and “why” such lessons have worked and are advocated for consideration.

One recent trend in IT project management is toward developing “hybrids” that combine elements of traditional and agile methods. This may be a temporary phenomenon as organizations add agile methods and remove traditional ones in the process of evolving toward agile, or it may be a more static approach to creating a new form that combines the best of both or appeases staff members with multiple philosophies. We do not focus on these hybrids per se but rather are proposing a sort of hybrid not between traditional and agile methods but by surrounding relatively pure agile methods with explicit project management techniques. Difficulties adopting pure agile “straight out of the gate” are documented by Baskerville et al. [8], which traces changes from traditional to agile approaches in particular organizational settings. That organizations adjust the “pure” practice of agile is well known, but the shape of such hybrids after adjustment and what is needed to substitute them for well-established practice is not as well examined.

It is worth noting that many of the issues and concerns of this viewpoint are less pressing and even non-existent in “greenfield” start-ups, whether new organizations or new IT departments, where the traditional practice is not ingrained and, therefore, need not be removed. On the other hand, the impression that agile practices in the full or hybrid form are universally implemented is belied by the observation of organizations currently beginning transformation and others remaining in a limbo of claiming to use agile but still clinging to traditional practices.

---

<sup>3</sup> Tripp et al. [18] describes a variety of components from both traditional and agile approaches considering, among other things, how these are combined in various configurations.

Some of the models referenced in this paper are fairly standard “off the shelf” frameworks available on the web or in agile process books. We will generally not reiterate what these helpful models have provided, but rather it is our goal to point out ones that have been particularly useful and elaborate on nuance regarding how to put them into practice. It should also be noted that much of our findings are consistent with the long research experience described in Conboy and Carroll [9]. For example, among their major challenges are issues of ensuring a common vocabulary (#1 in their scheme), issues of organization structure (#4), of top-down versus bottom-up implementation (#5), and misalignment of customer processes and large scale frameworks. All of these challenges are observed in our discussions. We aim to provide examples of these and some, hopefully, useful approaches to addressing them in this study.

Below we discuss fundamental detail level issues grouped by individual adoption and adaptation, organizational leadership and structure, transitioning tactics, and tools and techniques. No claim is made that these are either the only or the best way to group these lessons and observations. Nor that there is no overlap and commonality among these lessons, particularly as regards the pervasive yet moderately differentiated need for a changed “mindset” to make agile work well. Following agile approaches requires the spirit of agile as well as the mechanics of following its “rules”.

## 2. Individual adoption and adaptation

Individuals work as part of teams with varying levels of collaboration and interdependence, but at the same time teams do not come to life without individual members. The individual is enabled and constrained by the organization, while the organization can accomplish nothing unless individuals complete their assigned and ad hoc tasks. The following specific issues and lessons can be more easily identifiable in terms of individuals as they work in teams. These can be divided into adoption issues regarding characteristics of individuals engaging in agile development and adaptation issues referring to procedures, interactions, and effects encompassing tendencies of whole projects.

### 2.1 Adoption

**Self Discipline and Collaboration.** Typically, agile approaches require shifting to a different type of work style by team members. Some who are accustomed to traditional approaches have not the experience, imagination, or habits to begin asserting initiative and taking responsibility for completion, that a fully realized agile approach requires of all team members. It is also important to look for situations where team members feel “forced” to shift to agile approaches. In some cases, shifting these folks back to teams using traditional methods works best; in cases where there are no traditional method teams to be shifted to, team members may resign or look for work elsewhere.

*Managerial approaches* include: careful selection of team members with a preference for responsibility-taking and incentives/reinforcement for taking the initiative – note that this might require allowing more range for diverse approaches and room for some failure (with opportunities for correction and improvement). Acceptance, even eagerness to move to agile approaches tends to correlate with the traits of a higher level of comfort with communication. Managers are advised to look for and cultivate this characteristic in structuring a high communication culture.

**Agile leadership experience.** Successfully leading projects using agile technique is aided by experience. This shows up in terms of confidence in making key choices, determining which issues are normal and which need special handling, and the like. Experienced leaders may not be available for all (or any) projects, particularly at the start of a new program or shifting from traditional to agile.

*Managerial approaches* include: bringing in outside coaches, providing some slack on schedule and cost targets when supervising less experienced leaders, and training programs for new project managers. Certifications for scrum masters and the like can help impart skills and create a common language and approach.

**Developing judgment and self-governance.** In traditional project teams, much responsibility is embedded in individual roles. The database specialist makes sure tables are set up properly, and the network specialist makes sure that communications among components are reliable. However, in agile teams, responsibility devolves to the entire

team. Individuals may have a specialty but are expected to function in multiple ways exhibiting what is often referred to as “T-shaped skills”. In a well-functioning team, people expressing conflicting views and take action to resolve their conflicts. In ill-functioning groups, people tend to walk away from these responsibilities. When the committed work is not finished at the end of a sprint due to team members not accepting this responsibility, there is a temptation for the scrum master or project manager to step in and resume assigning responsibility. However, this can become a habit, and the team may learn that it can avoid having to accept such responsibility.

Many leaders and team members embrace the freedom and responsibilities that come with the transition to agile and its tendency toward self-governing teams. However, not all key personnel are likely to embrace this to the same degree, particularly when they come to see the drudgery as well as influential aspects of administration. Those who do embrace the idea of self-governance may still need to develop abilities to exercise freedom wisely and to add clarity to communications to collaborate with other team members. Those who do not embrace such freedom and responsibility may also need to learn how to exercise higher levels of personal responsibility, whether this is their preference or not.

*Managerial approaches* include: assessing formally or informally the capabilities of each team member; modeling excellence in their own freedom tempered by responsibility and task completion; and generously working with team members when their skills need improvement or attitudes need reevaluation.

**Semantics.** It is important to establish common meaning to common terms. Team members and other stakeholders can get really confused when using terminology differently. Rather than adopt common meanings, individuals may use alternative meanings to fit narratives that take away from rather than contribute to the overall mission. For example, it is important when using Jira [20] to understand the names of levels like epic, user story, and the rest.

*Managerial approaches* include: Listen carefully to what is meant as well as what is said when interacting with team members and other stakeholders. Be quick to react when terminology begins to slip from the defined usages. Note that observing such a divergence of meaning can be a useful opportunity for insight into deeper problems that the team needs to have addressed.

## 2.2 Adaptation

**Breaking down the work into manageable user stories.** Use cases are more associated with traditional than agile methods. A use case describes how a system will act. In agile a user story is about the result and benefit of the story being described. User stories leave out details for the agile team to explore. When developers are solely using agile methods, typically they will only focus on user stories. Hybrid methods will sometimes extend the range of methods to identify end results to include information requirements, and use cases along with user stories. User stories differ from information requirements in their central concern for creating ultimate value for users (including sponsors and those affected by rather than directly using systems). As a result, the emphasis shifts from detailed enumeration of all features in the new system to creating particular results with mechanisms only generally specified. It is a shift of attention from means to ends. Team members immersed in traditional information requirements development tend to experience discomfort with the “loose ends” of not detailing all specifications before starting the building process. Note that this is an area where team member concerns may be inflamed by management demands, as will be discussed below. This is one area where “greenfield” teams are more likely to move directly into a focus on creating value as quickly and efficiently as possible. More specific issues for agile teams relative to user stories include; integrating stories of multiple users and tasks that are likely addressed by single or related applications; ensuring that all steps are included; disaggregating continuous user actions into discrete value-adding tasks; and focusing on translating physical processes into digitally managed or enacted ones.

*Managerial approaches* include: establishing a continued and consistent focus on value creation throughout the development process supporting the agile mindset.

**Responding to setbacks.** If a project “goes south”, perhaps it only makes sense to return to familiar procedures before more or irreversible damage is done. On the other hand, there are likely times when it is only by pressing ahead that temporary setbacks are overcome and where momentum is gained for more permanent change. Even at some cost,

moving through setbacks and staying the course can provide important lessons that inform long-term future actions. For example, shifting to written approaches to tracking issues and exceptions may temporarily slow down product development progress, but as it becomes routine may accelerate task completion and smooth task assignments. Knowing when to move forward anyway and when to draw back is pivotal.

*Managerial approaches* require developing a sensitivity to the effects going forward or backing off will have on the team. Will going forward ultimately pay off, even if not obviously so in the short run? When is it better to capitulate in the short term in order to return to the issues and, having gained insight into the resistance, approaching it in, hopefully, more effective ways?

**Lines of authority.** As in all projects, it is difficult to motivate team members to perform all tasks at a high level of quality when their reporting is to their own functional area manager rather than the project manager. At one client site, it was almost impossible to hold members of the agile team accountable, including their boss who was the product owner. The project team resisted accountability unless it came directly from their functional manager. One of them actually told the project manager that it was not his responsibility, even if it was his responsibility. Clearly, this is a problem not unique to agile project work, but the agile approach is not immune from it. Managers, to our knowledge, have no automatic or foolproof ways to address this sort of problem.

*Managerial approaches* include: working to fundamentally alter how organizations structure their lines of reporting; use informal levers such as culture and peer pressure when available; though perhaps unfair at times, shift work to more responsible employees; attempt to align incentives with productive engagement where possible; and elevate the problems to unitary managers where they may be addressed. It can also be helpful for managers to establish the “rules of the road” early, which involves standards for social interaction and behavior such as when and how to escalate issues and standards for naming items, storing in common directories, and version controls from initiation to having fully tested components. All of these approaches have the potential to backfire and make matters worse but are worth considering and customizing for individual circumstances.

### 2.3 Summary

Table 1 summarizes the individual and team adoption and adaptation factors.

Table 1. Individual and team adoption and adaptation factors.

Factor	Description
<b>Adoption</b>	
Self-discipline and collaboration	Team member selection, “weaning” from traditional motivators.
Agile leadership experience	Acquiring and using experience and expertise. This can create a paradox starting out with agile because inexperienced leaders may run into serious difficulties blamed on the method rather than viewed as just a natural part of the learning curve.
Developing judgment and self-governance	Need to appreciate that agile requires more “art” skills than traditional development beyond the “science” of technical skills.
Semantics	Determine a meaning for each key term, use these consistently, and insist on the same from all team members.
<b>Adaptation</b>	
Breaking work down into manageable user stories	Build competencies in the art of disaggregating epics into user stories based on units of value delivery and developer activities.
Responding to setbacks	Both the SDLC and agile are complex systems where various activities are intertwined. One task may be more complex in agile, but doing it in an agile way may make downstream tasks more effective. A quick reaction to the initial difficulties of an upstream task being more cumbersome may restrict the downstream ability to capitalize on new procedures.
Lines of authority	Establish, where possible, informal levers for asserting authority and ensuring completion of activities where formal ones are missing or not organizationally supported. A good project manager can surface and manage conflict among group members, whether in terms of details of a task, assignments, or understanding of various instructions. Without a designated project manager, alternative tactics for such conflict resolution must be established, or such conflict can fester and slow project completion.

### 3. Organizational leadership and structural factors

As individuals and teams work within the organizational umbrella, organizations are composed of individuals and groups. We can look at organizational activity through either lens, but at the end of the day they are intertwined. We have divided consideration of the organizational factors into leadership and structure. Leadership pertains mainly to attitudes and skills of senior leaders generally outside, but responsible for, the performance of the development teams. Structure refers to how the operations of projects and teams are organized.

#### 3.1 Leadership

**Top Management Support.** The ability of agile to take root and flourish is inhibited if senior leaders set policy and act upon the mindset of task performance rather than value delivery. Senior leaders tend to have risen through the ranks understanding traditional methods and frequently are uncomfortable without the reporting and signs of progress that attend traditional projects. It can be disturbing to agile team members to be asked to shift to value delivery when senior managers continue demanding task performance reports. Much of the value of the agile approach derives from focused concern on delivering value in the customers' workspace and treating tasks fluidly as mechanisms to create such value, adding and discarding them as new ideas emerge. This adds a measure of unpredictability that is difficult to capture in traditional reports and depends on the understanding and initiative of team members to effectuate.

*Managerial approaches* include: It is not location, location, location, like with real estate rather, it is value, value, value. It is not just training but establishing a commitment to agile and an agile mindset that pushes the method from abstraction to realization.

**Cadence.** Project cadence refers to the pacing and flow of the project as it is carried out. Agile methods, of course, differ from SDLC by emphasizing iteration rather than progression sequentially through stages. There is a temptation to think of each iteration as a "mini-SDLC" where a module in the larger application can be worked through from requirements, through coding and testing, to implementation. Such a way of thinking may be helpful in breaking up a large project into smaller well-known pieces, but misses to a large extent the exploratory value-seeking aspect of the agile process. The leadership of the larger organization through that of each project needs to focus on the alternative rhythm of embedding the creativity of the project into the understandings being derived from the partially completed new states and what they suggest for the ultimate product. This is not to say that there should not be steady progress toward overall goals, but it is to say that such progress may not be accrued in one steady predictable line.

*Managerial approaches* include: it is important to realize that Scrum is both iterative and incremental. Each iteration is focused on delivering incremental value; in some cases, items embodying new value are moved to users in batches as completed at the end of iterations, even though the value is planned to be created throughout the process. The critical awareness is that agile processes deliver value regularly, but the exact specification of that value is not precisely definable at the beginning of the project.

**Insulating.** A major purpose of hybrid approaches derives from the different mindsets of the agile team (once fully established) and more traditional senior managers. The move to agile shifts a team from structured sequences in development, to accumulation of knowledge and partial product with the intention of creating maximum value (rather than completing the plan). However, traditional managers are not necessarily comfortable allocating money and effort with only a vague idea of the value to be created rather than seeing the detailed elements of the product. It is worth pointing out that many of the commonly practiced project management tasks such as risk analysis, change management, or issue tracking are explicitly part of the SLDC, which is focused on the steps in producing new artifacts rather than managing the process per se.

With the switch from SDLC to agile, however, the unspecified nature of such project management activities becomes apparent, and project, program, and portfolio managers have the opportunity to translate broad agile work team goals into the sort of reporting that will allow senior managers to have insights into the current state of projects as they mature. This buffering activity is more of an art than a science and requires ongoing adjustment so that formal plans and actions on the ground can be steadily realigned as they co-evolve.

*Managerial approaches* include: insulating the team so it can do its scrum activities, but PM translates the JIRA metrics into something that senior managers are accustomed to seeing.

### 3.2 Structure

**Bringing Work to the Team.** Organizations moving to agile require a shift from a mindset of moving teams to the work to work to the team. In general, agile teams should remain intact from project to project, receiving new assignments rather than having teams built afresh for each new task as it is moved to being a project. When teams stay together this way, they can build knowledge as well as work procedures instead of reinventing it whenever the team is abandoned. DevOps, in principle, incorporates this even further by leaving the team building a new application intact to maintain what it builds. This has the added advantage of teams being incentivized to build concerns for ease of maintenance into the design and building of new systems – a practice that may add some time and cost to projects but the payoff in their lifetime with faster and more reliable updating.

*Managerial approaches* include: forming teams that stay together or bringing the work to the team so that the team can specialize in particular categories of projects rather than forming new teams for each project. The purpose, of course, is both to develop strong team dynamics and expertise in more focused business areas. Constant flow of projects, customers get to know the team members, more domain knowledge; like a baseball team, same team together through the whole season.

**Cross-functional integration.** The product manager plays a key role in the agile development team. Sometimes known as the “customer representative”, the role may demand multiple, even contradictory, actions and positions by those fulfilling it [10]. However, some projects call for products and services that cross boundaries within the organization (or across multiple customer segments). When there is a product manager, but the product is aimed at supporting multiple processes, departments, or divisions, it is important to account for the effects of the product across user groups. Obvious solutions, but not necessarily ideal long-term solutions, are to add all features requested by any group. This may provide a product that serves many user stories but may generate bloated, inefficient products based on weak architecture. The product manager may not be able to represent all the various outside stakeholders and, even if cognizant of all the diverse requirements, may not be able to blend them into a single lean set. As a result, skills needed for this position include the ability to elicit needed and wanted features; to differentiate among these and prioritize them, to find solutions that may differentiate features for users while providing integrated and streamlined “plumbing”, and, perhaps most importantly, the ability to explain and persuade clients when requests cannot be met while retaining the integrity of the system (or at least not at the prices they are willing to pay).

*Managerial approaches* include: maintaining awareness of the varied uses that different groups in the organization will make of the final products, begin early to develop tactics for meeting as many needs as possible, providing, where possible, common architecture for meeting multiple goals and communicating about tradeoffs that seem to be inevitable.

**Scaling.** Techniques such as Kanban [21] and Scrum [15] can work very effectively in small groups but become difficult to manage as the size of the project and team grows. Kanban lists can grow and become overwhelming to navigate. Scope changes with the addition of project tasks can cripple the ability to come to completion unless user story prioritization and screening are accomplished by the product owner. Scrum meetings with more than 10 participants can start to leave out important messages, take a burdensome long duration, or proceed so quickly that not all are adequately informed about all critical items.



It is critical to use the support of software tools such as Jira to administer a more complex and higher quantity of tasks. It may be important to divide teams up into separate scrums and/or to perform them through collaboration software like Zoom [22] to provide multiple channels such as dialogue and chat as well as to automate the recording of proceedings and ease the capture of minutes for the permanent record.

*Managerial approaches* include: Conducting a “scrum of scrums”. By engaging the scrum masters in debriefing together, brings disparate pieces together toward integration. Release plans and calendars can be distributed so all scrum masters and team members know what deadlines are being worked toward as well as integrating standards so that naming conventions, interfaces, and the like, which may tend to stray, are brought back into conformity.

**Product delivery.** It generally does not help to speed up an assembly line if there are no trucks available to deliver the products. Agile methods ideally produce products more quickly but organizational institutions are needed to move them to actual use. One model for this is the SAFe model [19] that displays the relationship of agile project methods to an array of allied organizational programs for smoothly moving new applications into production. Another model is DevOps [23], where the transition from development to operations becomes a focus of process actions. Hemon et al. [11] describe a case showing the transition from agile development to a DevOps delivery program.

*Managerial approaches* include: realize that models like SAFe are relatively easy to conceptualize but more difficult to actualize. That said, these sorts of models provide a useful template or outline of needed structures into which individual projects, clustered programs, or portfolios can be embedded. Build capabilities for the transition of products from development into operations.

**Method selection.** There is a benefit on a project-by-project basis for optimizing choice between SDLC and agile approaches for each project. The Stacey Matrix (e.g., Agrawalla [12], see also Stacey [13, 14]) for a broader discussion of the underlying conceptualization) provides one approach to assessing which to use based on the clarity/obscurity of the requirements and the familiarity with the technology. In essence, projects where dimensions are well known, can be handled routinely with SDLC, but as they grow in complexity, the value of agile emerges.

*Managerial approaches* include: Set up a process that assigns projects to traditional or agile approach teams. Maintain a backlog of projects for both and refrain from shifting projects to less optimal selections just because this set of teams has available capacity. Over time, calculate the ratio of traditional and agile projects and align the size of team members to more or less the same ratio.

**Administering Multiple Approaches.** In order to accommodate both traditional and agile approaches fully in the organization, either all staff need to be familiar and comfortable with both, or staff members need to be grouped and assigned to those projects adopting the techniques for which they are skilled. Either approach necessitates some management overhead and effort. Costs, therefore, of managing multiple development styles should be balanced against the rewards of more optimal project matching.

*Managerial approaches* include: The ratio of traditional to agile should be a function of the number of complex and innovative projects versus those that are routine and more predictable. Assuming discrete structures for traditional and agile portfolio management, the organization will want to install an efficient method for sorting the right project to the right team, staff the pool of team members for each group approximately according to the expected ratio of each, provide cross-training for all team members so that, when necessary, for example, if the ratio changes, staff members can shift between the two project structuring approaches.

### 3.3 Summary

Table 2 summarizes the organizational factors leadership and structure.

Table 2. Organizational factors leadership and structure.

Factor	Description
<b>Leadership</b>	
Top Management Support	Senior managers must transition from a task orientation to a value creation orientation and develop new ways to understand interim steps toward creating value rather than task accomplishment.
Cadence	The project manager buffers the agile team from the paradox of applying creativity while being asked for detailed plans while also providing ongoing project reporting for senior management sponsors.
Insulating	Project, program, and portfolio managers have to assume the role of translator or buffer between what actions teams are taking to create value and the information senior managers need to understand current states during project execution.
<b>Structure</b>	
Bring work to the team	Leaving teams intact builds human capital that can be applied to enhance downstream projects.
Cross-functional integration	The product manager needs to manage demands and requests across the range of user stakeholders.
Scaling	For large projects, scaling requires more adjustment than simply breaking larger units into smaller ones or just adding more tasks and people than a process is envisioned to accommodate
Product delivery	Agile methods can ideally produce products more quickly, but organizational institutions like continual delivery programs are needed to move them from the development arena to actual use.
Method selection	Use the STACEY model or an equivalent to move lower uncertainty projects toward traditional methods and higher uncertainty projects toward agile methods
Administering multiple approaches	In order to realize the benefits of using the more appropriate method for each project, organizations may need to maintain capabilities for both traditional and agile approach teams. Achieving such benefits requires more than a skillful assignment of projects to approaches but also skillful recruiting, retention, and training for team members, skillful adjustment of incentives for team members using each approach, and effective communication among divisions to optimize the use of resources, human and digital.

## 4. Managing the transition

Some issues relate more clearly to transitioning to agile methods without being specific to individuals, teams, or the whole organization. During the transition from SDLC to agile, there is the need to simultaneously work at the most elemental levels of work implementation and the highest levels of policy and senior management strategy. The theme of the transition is moving policies and practices across the range of participants in a manner that keeps them in alignment and/or returns them to alignment when gaps appear.

**Suppressing the old ways.** New ideas need to be introduced and demonstrated repeatedly. It should not be misinterpreted that this can be a “one and done”. New practices need to be reinforced, and successes need to be noted. Persuasive communication with team members needs to be consistent in the rewarding movement to the new agile terminology and procedures and moving away from the traditional. By not translating or dropping the vocabulary of SDLC, the benefits of shifting to a completely different system are not fully realized. As a result, some of the old procedures are unnecessarily retained, or new procedures are not fully enacted. While practicing the new techniques, teams may not be generating information that would have been normally produced and passed along in traditional development projects. Senior leaders in the organization can express support for moving to agile methods but retain an expectation of receiving status reports and deliverables in the same manner as when using traditional methods. This can happen even when equivalent information is reported but in an unusual format or with broader ranges of precision.

*Managerial approaches* may include: serving as a buffer to produce data from agile projects that approximate the sort of data traditional projects automatically produce; interacting with senior leaders both to forward what information exists and to assure progress, even if produced through processes less focused on technical milestones. The project manager may need to work with both external stakeholders and internal project team members to implement the transition to some degree in parallel so that neither group gets too far outside the expectations of the other.

**Managing the baseline.** In traditional approaches, the baseline generally represents an absolute that should not change other than through approved change management processes. Agile team members need to view this as a starting point that will continue to evolve. The agile team may retain the original baseline to see from where they have come, but it is not used as a target to achieve. Change shifts from something to be avoided to something accepted and expected. Using the baseline in the agile method often begins with a slender set of tasks and fuzzy ideas of the finished product. As the tasks become known and the finished product takes form, the evolving baseline reflects this growth. Management of what might be added and taken away from the baseline becomes a new process management task that may require new approaches relative to traditional change management.

*Managerial approaches* include: establishing a continued and consistent focus on presenting products that provide customer value throughout the development process even if the exact nature of the products is not fully described in the original baseline.

**Shift to new tools.** There is a tendency, particularly where projects have relied on standard Office tools (e.g., Word, PowerPoint, and/or Excel) to have the reluctance to use more specialized and powerful tools like Jira. Rather than implement Kanban, for example, in Excel, a transition and descent down the learning curve for more powerful tools is advised. These new tools are designed for supporting agile processes and offer additional features that, once adopted, help shape agile development work practices. Adoption of new work practices can stimulate and sometimes require advanced tool use. Movement to full use of these tools can be gradual, starting with activities that mirror those of traditional processes to the addition of new work practices along with support from new tools.

*Managerial approaches* include: investing in powerful new tools, ensuring that team members learn effective tool use skills, and monitor that tool use occurs (correcting if and when bad habits form).

**Agile coaching.** The pure agile method offers a powerful template for development projects, but in organizations, it needs to be customized to account for team member preferences and personalities; senior management requirements; and the particulars of the business task environment. Agile coaches can be helpful in translating the general ideas of agile development process for the needs of the individual firm. Where qualified and motivated consulting is available from outside the organization, there is an opportunity to profit from experience across a range of industries, work products, team structures and personalities, and senior management goals and preferences. Sometimes an outside voice can bring a different voice of authority to both team members and external stakeholders that helps facilitate a willingness to experiment with new approaches.

*Managerial approaches* include: squeezing as much value as possible from general agile techniques, looking for coaches with the sort of experience that will help customize general agile templates for local conditions, and define the role that consultants are expected to play and monitor that their advice and support is effective.

**Top-down roll out.** Only very small organizations can implement Agile piecemeal depending on the needs of individual projects. Very large organizations need to start with Enterprise Agile and then work it down to the project level. This is because organizations are generally structured in silos, and agile is essentially outside silos. This tends to upset organizational structures and traditions, particularly regarding incentives. For example, the number of people reporting or how their people perform on projects. Organizational structure works against you setting up agile unless these are also changed.

*Managerial approaches* include: 1) understanding limits on implementation of agile at the work level if not supported through structure at the senior level; 2) manage up in terms of providing what's necessary for them while performing the work differently (acting as a buffer); and 3) advocating as possible for shifting senior management mindset and practices in a way that supports transformation. As people go agile, they are moving from a project approach to a product approach; instead of funding the work, they are funding the final product. In SAFe, there are three levels of Kanban, and at each level, there are limits to the number of projects handled at one time. Managing the backlog at each of these levels is critical, particularly prioritizing the sequencing of activities so that in each period, the team is working on the user stories promising the most value. In principle, this is fairly standard work, but in practice, the key is the rigor with which it is implemented.

### 4.3 Summary

Table 3 summarizes the transition requirements.

Table 3. Transition requirements.

Factor	Description
Suppress old ways	Managers need to bring external stakeholders and team members long at roughly the same pace to maintain mutual understanding and alignment.
Managing the “baseline”	The use of the baseline needs to be shifted from a target that team members put effort into meeting to a documentation of the evolution of a project as the team focuses on creating end-user value.
Shift to new tools	It is important to align the shift to new tools reflecting changing work practices in order to facilitate and reinforce both.
Agile Coaching	When carefully selected and skillfully monitored, outside consultants can bring new ideas and help reinforce essential new practices.
Top-down roll out	A significant amount of top management support and understanding is necessary for the roll out of agile project approaches in large organizations.

## 5. Tools and techniques

Of course, tools and techniques are used by individuals or teams and endorsed and supported by organizations. Nevertheless, it is worth focusing attention on these particular tools and techniques as the choice to acquire, use, master, and extend them provides a level of structure that may persist as individuals come and go over the years.

### 5.1 Tools

**Chargeback versus overhead.** Paying project by project will inhibit adoption rather than accelerate it. Some companies push back on licensing specialized agile software per user (even when software may be as low cost as \$10-20 per user). Assessing the value of using tools in increasing productivity across projects is important relative to how and from what budget they are paid for.

*Managerial approaches* include: using the best tools available should be viewed as an investment that can be spread over many projects. Just as a farmer would prefer not to use a manual plow or an obsolete tractor and expect maximum crop yield, senior managers should not expect optimal development team performance with staff constrained to use awkward tools prone to errors or difficult functioning.

**Full Use of Tools.** Clients of the first author have tried to use JIRA and Scrum without the full understanding of backlogs and running the sprints. The mechanics of adding data through JIRA to Kanban lists is straightforward but can be ineffective if there are not procedures to evaluate and prioritize these additions. Individual team members are not the ideal ones to be scanning long lists of needed actions and deciding which are of the highest priority. It is not sufficient to just walk activities through Kanban, even though that technique can work, for example, where an IT dept handles problem tickets. However, for this to work in agile teams, there needs to be more information. There is no inherent accountability in Kanban in and of itself. It requires the explicit accountability of Scrum to make it work. Kanban by itself only gives the appearance of agile but does not adhere into getting the work done well.

*Managerial approaches* include: establishing new tools to replace familiar ones to support routine tasks, introducing new features and procedures to extend the range of tasks supported by new tools, and integrate new tool use with an agile mindset regarding overall development approaches.

### 5.2 Techniques

**Confusing Minimal Viable Product (MVP) with Minimum Marketable Product (MMP).** Where MVP focuses on validating assumptions and learning about users' preferences – this level of work should remain internal to the team. In contrast, MMP incorporates a core set of functionalities that addresses customer/user needs, creates the desired user experience, and can start creating quantifiable value for the business. It can be released to the customer. This distinction

is important to keep in mind because there are necessary elements to products that are being released that go beyond those that affirm that the central functionality is useful to the customer. For example, the user may not directly interact with automated backup, internal documentation notes, peak load capacity, or potential for errors with useable but not fine-tuned interfaces. Managerial actions include establishing a checklist of “behind the scenes” standards checked and refined, when needed, before releasing even approved products to the customer.

*Managerial approaches* include: establishing a continued and consistent focus on value creation throughout the development process consistent with an agile mindset.

**Reframe Stages.** Highsmith [15] presents and details an alternative stage model for agile projects. The early stages focus on envisioning the goals of the project and building the initial backlog and planning releases. Exploration involves work iterations to build both the detailed elements that create value and the technical elegance to realize these. These are followed by the mechanics of putting the creations into use, including launching and refining particularly, if necessary, capacity and security issues need to be addressed relative to differences in development and production environments. Development should be behind firewalls, but when the new system moves to the production environment, it needs an additional review to ensure security.

*Managerial approaches* include: reinforcing the flexibility within the overall stages and effectuate useful fit in the production environment.

**INVEST.** Developing user stories may seem straightforward, but there are six attributes of effective user stories [16]: independent; negotiable; valuable; estimate-able; small, and testable. These provide tools for evaluating the quality of user stories. It is very practical to assess each user story on the basis of these criteria as well as for writers of such stories to embed these characteristics more and more automatically into new ones as they create them. The application of these criteria is fairly routinely made when working with business analysts who have received certification but much less common with business analysts or project managers that are not certified and not inculcated with this perspective. This is important to emphasize because without attention being called to these attributes of good user stories, many assigned this task just do not do it well. Those trained in information requirements are often not good at disaggregating user stories. When the project team is given epics, they become the ones to break them down to the level where they can be handled in an ad hoc manner. Undertaken by someone adept at this, the team can minimize “fumbling around” and reinventing the wheel. It is really an art form writing really good ones.

*Managerial approaches* include: working with team members and clients to define user stories at the right level following the hierarchy of Epic > Feature > User story > sub-Task, encouraging team members to pursue the Professional Business Analyst program from PMI (or equivalent courses from other institutions).

**Art of the backlog.** Managing the backlog is an art, at least as much as it is a science. Disaggregation needs to break epics into user stories, so that team members can have the appropriate amount to chew; getting the right person to do the disaggregation. This contrasts with decomposition, which breaks modules into submodules where they reside without considering the relative value of how they are subdivided.

*Managerial approaches* include: recruiting a product owner who takes ownership (who represents the “single ring-able neck”), understands and can break epics into user stories, and interpret the business value in terms of technical actions.

**Story point estimation.** Story points are difficult to comprehend and do not have a single standard, but rather can vary in how they are assigned. Some team members go straight to hours for estimation of effort, but this can be problematic when shifting to agile. Even normalized story points where one equals about four hours of work, three a full day, and so on can absorb unproductive time in the estimation process. Another approach to story points is to estimate them relatively, rather than in absolute terms. Setting a standard point value on the easiest story with a score of, say 3 on a scale of 10, allows all the other stories to be estimated relative to that first one. An easier than standard story maybe a 1 or 2, where a more difficult one can be rated up to 10. Thus, like with Analytical Hierarchical Process [17] cognitive load is eased by using pair-wise comparisons. The risk with time estimates is of taking a great deal of effort just in the estimation process, with the estimates turning out to be notoriously inaccurate anyway. With the user story, work is

defined in terms of benefit to the user in contrast to the traditional project management concept of a work package viewed in terms of internal activities. Assessing accomplishments of an agile iteration are the product of the story point estimation and the amount of value delivered by each user story so that the accumulation of large numbers of low-value tasks are not necessarily appreciated more than a moderate number of higher-value tasks.

*Managerial approaches* include: establishing a continued and consistent shift from the estimation of time by task to comparative estimation.

**Representing the architecture.** Particularly as the size of the project scales, creating independent features which move forward value for the client, it is necessary for all team members to be extending the same base product. When it is necessary for one team member to adjust the base product in order to facilitate the smooth operation of a new feature, it is important that this change becomes part of the representation of the base product that all team members are continuing to work from in the case where such a change does not affect other components but most importantly when it does (or where its effects may not be immediately apparent).

*Managerial approaches* include: assigning responsibility for selecting, using, and updating architectural model(s) at the level of detail necessary for streamlining processes for adding new features.

**Issue lists.** Distributed teams may each be committed to different tools, even if the final product needs to be written in one language. Such an issue needs to be recognized, listed, and “time-boxed” for the project manager to resolve, even if a unilateral and perhaps arbitrary decision is required. Similarly, issues of risks including potential adverse events, exceptions, proposed changes, and the like, need to be created. To complete their usefulness they must be augmented with monitoring and action taking where needed.

*Managerial approaches* include: establishing a list of issues to be recorded as they arise and resolved at the first opportunity.

### 5.3 Summary

Table 4 summarizes the tools and techniques factors.

Table 4. Tools and techniques factors.

Factor	Description
<b>Tools</b>	
Chargeback versus overhead	Shifting to tools designed for agile is a critical part of its adoption. Charging individual projects for tools is a sure way to stifle their adoption.
Full use of tools	Software tools for agile are extremely effective when, and only when, associated with the additional elements of the method that provide the governance for project progress.
<b>Techniques</b>	
Confusing MVP and MMP	The MVP is used for understanding and communication but rarely is ready for release into production. Much of the MVP work can be conducted manually behind the scenes in ways that are not visible for users in practice. The MMP cannot exist without the existence of at least one MVP.
Reframe stages	Early stages need to focus on visions of creating value rather than technical or even business requirements.
INVEST	High-quality user stories are more efficient to translate into useful systems and components.
Art of the backlog	Managing the backlog is an art, at least as much as it is a science.
Story points outperform time estimation	Shifting the emphasis from exact time estimates to approximate ones while also accounting for value created as well as work performed.
Representing the architecture	Modeling the base product into which the individual components fit and enable the architecture to evolve
Issue lists	Traditional project management (often outside either pure SDLC or agile guidelines) recognizes, lists, and addresses issues ranging from technical disputes to change requests in a standardized and comprehensive manner.

## 6. Conclusion

As with any study, the reader must keep in mind its limitations when drawing conclusions. In this case, some constraints that the reader should take into account are: (1) even though our domain knowledge expert has worked on a wide range of projects, portfolios, and programs across a good deal of industries, similar deep investigations with other experts may reinforce some lessons, refute others, and provide a range of additional ideas that can be applied to managing the transition to agile forms; (2) our domain knowledge expert has practiced exclusively in the United States and, more specifically, in the Midwest of the US, although members of his teams have been spread around the world, so encountered practices may differ markedly with other cultures and other regulatory regimes; and (3) our domain expert has worked exclusively with organizations that are primarily users of information technology, though some create many applications directly sold to customers as product or service. However, these are not software companies in the sense of broadly selling software tools for users to develop in their own way, as would be exemplified by Microsoft, SAP, or Adobe.

Although a good deal of knowledge was surfaced through our iterative interview process, as is typical, much more remains to be investigated. For example, a number of trends may affect the technical infrastructure and motivation supporting agile approaches. Cloud computing can enable agile teams as it supports team self-organization. The developers can decide how to build their cloud environment in a more decentralized (modular) way and often without long approval processes for new hardware and software. More general management shifts from traditional control and command mode to transformational and collaborative forms may also affect the headwinds as well as the demand for agile development methods.

Naturally replicating this study using similar or varied methods would be important to verify the efficacy of the lessons, surface new ideas, and qualify when, where, and how these lessons can be effectively applied. On the other hand, the possibility is raised of using these ideas to pose probing questions that extend this work. Open remaining questions include:

1. From an individual and team perspective, are there ways to ease the initial learning process for mastering the agile approach? Are there ways to continue evolving the agile approach to make it even more effective? Are there ways to ensure that all team members, agile coaches, and senior leaders are immersed in the “agile mindset”?
2. From an organizational perspective, assuming that a good working alignment is achieved between project teams and senior managers, how are these maintained, and how are the successes accrued using agile reinvested into continual process improvement? For example, are there particularly effective ways to calculate the net financial outcomes balancing project and allied costs versus the actualized benefits from the new products?
3. From a tool and procedure perspective, are there straightforward, perhaps even automated, ways to follow emerging best practices and incorporate new tool features as they are invented, for example, extensions of metrics and analytics to accelerate best practice? Are there better ways to estimate use story points?
4. From an academic perspective, are there heuristics for the use of multiple approaches (assigning projects to traditional or agile rather than using either exclusively) – if so, are there ways to minimize costs associated with such multiple approaches?
5. From an academic perspective, is the use of project management technique to serve as a “buffer” between management stakeholders and team members relatively effective compared to other sorts of hybrid approaches that actually combine traditional and agile method components?

This viewpoint is written in the spirit of extending our understanding of the implementation of agile and hybrid methodologies at a detailed level based on actual work experience. There is a growing belief that agile practices are “better” than traditional ones. Evidence and anecdotes suggest this is true in many cases. They are clearly being adopted widely by organizations. However, the successful movement from traditional to agile often ends up with a sort of hybrid which may take uncountable forms. Moreover, the transition to using agile among teams that have deeply embedded traditional practices takes ingenuity, patience, persistence, and some strategy. This viewpoint discussion is intended to highlight the lessons of an expert project/portfolio manager who has effectuated such transition both as an organizational employee and as an external consultant. The overall theme is less about integrating traditional and agile components and more about using traditional project management techniques customized to the agile environment in order to provide an alignment between the working culture of the team members and the monitoring culture of senior leaders. Additional themes running through these various more detailed lessons and observations are:

1. The project manager can intervene with management allowing team members to use an agile technique to create positive organizational value in a more spontaneous discovery-oriented mode while providing data regarding the progress that senior managers are used to and need for their own strategic decision making;
2. The movement from a traditional to agile-oriented approach requires much learning on the part of typical team members, but single doses of training sessions should only be viewed as the beginning to be reinforced over time in a persistent manner until the mindset and practices become habitual;
3. Tools and procedures can be used together to create new work practices and support them using tools designed for those purposes.

## References

- [1] J. Birkinshaw, “What to Expect From Agile,” MIT Sloan Management Review, vol. 59, no. 2, pp. 39-42, 2018.
- [2] G. Lee and W. Xia, “Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility,” MIS Quarterly, vol. 34, no. 1, pp. 87-114, 2010.
- [3] B. Hobbs and Y. Petit, “Agile Methods on Large Projects in Large Organizations,” Project Management Journal, vol. 48, no. 3, pp. 3-19, 2017.
- [4] A. Imran and S. Gregor, “Conceptualising an IT mindset and its relationship to IT knowledge and intention to explore IT in the workplace,” Information Technology & People, vol. 32, no. 6, pp. 1536-1563, 2019.
- [5] K. Conboy, “Agile from First Principles: Reconstructing the Concept of Agility in Information System Development,” Information Systems Research, vol. 20, no. 3, pp. 329-354, 2009.
- [6] T. Dingsøy, T. Dybå, M. Gjertsen, A. O. Jacobsen, T.-E. Mathisen, J. O. Nordfjord, K. Røe and K. Strand, “Key Lessons From Tailoring Agile Methods for Large-Scale Software Development,” IT Professional, vol. 21, no. 1, pp. 34-41, 2019.
- [7] F. Niederman, T. Lechler and Y. Petit, “A Research Agenda for Extending Agile Practices in Software Development and Additional Task Domains,” Project Management Journal, vol. 49, no. 6, pp. 3-17, 2018.
- [8] R. Baskerville, J. Pries-Heje and S. Madsen, “Post-agility: What follows a decade of agility?,” Information and Software Technology, vol. 53, no. 5, pp. 543-555, 2011.
- [9] K. Conboy and N. Carroll, “Implementing Large-Scale Agile Frameworks: Challenges and Recommendations,” IEEE Software, vol. 36, no. 2, pp. 44-50, 2019.
- [10] L. Maruping and S. Mahook, “The Multiplex Nature of The Customer Representative Role in Agile Information Systems Development,” MIS Quarterly, vol. 44, no. 3, pp. 1411-1437, 2020.
- [11] A. Hemon, B. Lyonnet, F. Rowe, B. Fitzgerald, B., “From agile to DevOps: Smart skills and collaborations,” Information Systems Frontiers, vol. 22, no. 4, pp. 927-945, 2020.



- [12] R. K. Agrawalla, "Towards a theoretical foundation for agile development," in 2015 IEEE International Symposium on Systems Engineering (ISSE), 2015, pp. 461-468.
- [13] R. D. Stacey, "The science of complexity: An alternative perspective for strategic change processes," *Strategic Management Journal*, vol. 6, no. 6, pp. 477-495, 1995.
- [14] R. D. Stacey, "Complexity and the Group Matrix," *Group Analysis*, vol. 34, no. 2, pp. 221-239, 2001.
- [15] J. Highsmith, *Agile Project Management*, Second Edition, Pearson Education Inc.: Boston, pp. 197-198, 2010.
- [16] B. Wake, B. (2021, July 17). INVEST in Good Stories, and SMART Tasks, web-posting, XP123, August 17, 2003 [Online]. Available: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- [17] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw Hill International, 1980.
- [18] J. F. Tripp, C. Riemenschneider and J. B. Thatcher, "Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign," *Journal of the Association for Information Systems*, vol. 17, no. 4, pp. 267-307, 2016.
- [19] SAFe. (2021, July 17). Scaled agile "about us" page [Online]. Available: <https://www.scaledagile.com/about/about-us/>. Accessed July 17, 2021.
- [20] Atlassian. (2021, September 7). Atlassian Agile Coach [Online]. Available: <https://www.atlassian.com/agile/project-management/epics>
- [21] J. K. Pinto, *Project Management: Achieving Competitive Advantage*, Fourth Edition, Pearson: New York, cp. 372, 2016.
- [22] Zoom. (2021, September 7). Zoom home page [Online]. Available: <https://zoom.us>
- [23] L. Leite, C. Rocha, F. Kon, D. Milojicic and P. Meirelles, "A Survey of DevOps Concepts and Challenges," *ACM Computing Surveys*, vol. 52, no. 6, article 127, 2019.

**Biographical notes****Michael Durbin**

Michael Durbin is a principal consultant at Daugherty Business Systems with over 25 years of experience managing IT projects, programs and portfolios using traditional, agile, and hybrid methods. He serves as an adjunct assistant professor at Saint Louis University's School for Professional Studies where he teaches a variety of project management classes. In 2013 Michael received a Ph.D. in Higher Education Administration from St. Louis University, where he explored the perceived use and value of project management in institutions of higher education.

**Fred Niederman**

Fred Niederman serves as Shaughnessy Endowed Professor at Saint Louis University. He was selected as a Fellow of the Association for Information Systems in 2020. He serves as editor in chief for Communications of AIS. His work has been recognized as a "publication of the year" by AIS in 2015 and inaugural Communications of AIS, Paul Gray thought-provoking paper of the year in 2014. His areas of research interest include: IS personnel, IS project management, and philosophy of science applied to IS. He has served as senior editor for Journal of AIS and Project Management Journal. He has served as program chair for ICIS (2010) and as a member of the doctoral consortium faculty members (2018). He recently finished serving as chair for the special interest group on MIS in ACM (2017-2021). He is recognized as a member of the "circle of compadres" for the KMPG PhD Project.