Resolving Belonging Tensions in IT Transformation Programs for Restoring Sustainable Equilibria

Jessica Sanmartin  
*University of Bern*, jessica.sanmartin@iwi.unibe.ch

Thomas Hurni  
*University of Bern*, thomas.hurni@iwi.unibe.ch

Jens Dibbern  
*University of Bern*, jens.dibbern@iwi.unibe.ch

Follow this and additional works at: [https://aisel.aisnet.org/ecis2021_rp](https://aisel.aisnet.org/ecis2021_rp)

**Recommended Citation**  
[https://aisel.aisnet.org/ecis2021_rp/108](https://aisel.aisnet.org/ecis2021_rp/108)
RESOLVING BELONGING TENSIONS IN IT TRANSFORMATION PROGRAMS FOR RESTORING SUSTAINABLE EQUILIBRIA

Research Paper

Jessica Sanmartin, University of Bern, Bern, Switzerland, jessica.sanmartin@iwi.unibe.ch
Thomas Hurni, University of Bern, Bern, Switzerland, thomas.hurni@iwi.unibe.ch
Jens Dibbern, University of Bern, Bern, Switzerland, jens.dibbern@iwi.unibe.ch

Abstract

Successful IT business transformations require a departure from silo thinking in individual projects to a broader perspective in holistic, process-oriented programs. Such programs face latent paradoxical tensions that can become salient throughout their (re-)design and execution. Prior research suggests ambidextrous leadership by the program management team to resolve such salient paradoxical tensions. However, little is known about whether such unilateral and top-down approaches can restore sustainable equilibria or even trigger follow-up tensions between program-level objectives and project-level needs. Therefore, this study explores whether and how ambidextrous leadership to resolve strategic tensions can trigger follow-up tensions and how such follow-up tensions can be addressed to restore sustainable equilibria. To this end, we conducted an ethnographic study in an IT transformation program at a large Central European telecommunications company. We find that ambidextrous leadership is likely to create ripple effects by increasing belonging tensions between projects and programs, thus preventing sustainable equilibria. Moreover, we identify three ways programs and projects can address such follow-up tension. Our findings have significant implications for program management theory, paradox theory, and practice.

Keywords: Program Management, Project Management, IT Transformation, Business Transformation, Paradox, Belonging Tension
1 Introduction

Successful IT-enabled business transformations call for a departure from silo thinking in projects to a broader perspective in holistic, process-oriented programs (Davenport & Redman, 2020). Such IT transformation programs describe collections “of projects and activities structured to achieve an expected benefit (e.g., improved financial performance or value creation)” (Jiang, Klein, & Fernandez, 2018, p. 41). However, in achieving these benefits, IT transformation programs face latent, paradoxical tensions that can become salient throughout their (re-)design and execution. In (re-)designing programs, the challenge lies in addressing the paradox of opposing strategic orientations, such as differentiation vs. standardization. In executing programs, global program-level objectives may clash with local project-level objectives. Tensions that become salient can cause organizational drift, for example, deviating from the transformation program’s original objectives (Gregory, Keil, Muntermann, & Mähring, 2015).

Prior research on IT program management has contributed to our understanding of how paradoxical tensions can be resolved through ambidextrous leadership. Specifically, research has found that blending and balancing paradoxical demands can resolve such tensions. Blending is defined as “finding blended solutions that convince the organization that a given set of two demands is only seemingly paradoxical and is in fact harmoniously combinable”—balancing “as recurrently devising compromises among contrasting demands through dedicated coordination mechanisms and roles” (Gregory et al., 2015, p.76). Despite these insights, there are two gaps in understanding how to address tensions that become salient. First, we know little about ambidextrous leadership’s temporality or under what conditions sustainable equilibria can be restored to prevent recurrent tensions (Smith & Lewis, 2011). Second, there is a lack of understanding of how projects respond to ambidextrous leadership. Importantly, adjustments at the program level to address tensions between conflicting strategic demands could render latent tensions between program- and project-level objectives salient. While prior research on platform ecosystems has already observed such ripple effects in the form of global-local tensions following strategic decisions (Hurni, Huber, Dibbern, & Krancher, 2020), research on program management has not yet considered this critical issue. Therefore, we question (1) whether and how does resolving strategic tensions by ambidextrous leadership create new tensions and (2) how can such follow-up tensions be addressed to restore sustainable equilibria?

As part of our research, we decided to conduct an in-depth ethnographic study of an IT transformation program at SwissTelco, a large telecommunications company in Central Europe (Myers, 1999). Our analysis is inspired by the dynamic equilibrium model of organizing by Smith and Lewis (2011). Using this model, we identify ripple effects of ambidextrous leadership in IT transformation programs and understand how sustainable equilibria can be restored. We find that ambidextrous leadership can, indeed, render latent tensions in IT programs salient. Our findings suggest that applying ambidextrous leadership to resolve strategic tensions can make project members experience a lack of belonging to the program (patterns 1a and b). Beyond that, we identify three patterns to address such follow-up belonging tensions. First, ambidextrous leadership can resolve tensions in the short- and medium-term but cannot create sustainable equilibria because it does not address belonging tensions (pattern 2a). Second, belonging tensions resolved unilaterally by projects can create sustainable equilibria, although it is only considered a workaround (pattern 2b). Third, belonging tensions resolved mutually through programs and projects can resolve tensions and establish sustainable equilibria (pattern 2c). These findings contribute to program management and paradox theory and have important implications for practice. Next, we elaborate on program management and paradox theory. Following this, we present our method, findings and discuss the contributions, implications, and limitations of our research.
2 Background

2.1 IT Program Management and Ambidextrous Leadership

IT transformation programs strive for organizational benefits (e.g., improved value creation) by coordinating projects and resources during organizational changes or transformation situations (Jiang et al., 2018). In doing so, programs likely encounter paradoxical tensions with regard to their strategic direction (e.g., between efficiency and innovation, standardization and differentiation, and integration and replacement) or in terms of their operational alignment (e.g., between agility and stability, control and autonomy, and coordination and isolation). Although these tensions usually remain latent (Smith & Lewis, 2011), environmental turbulence and business dynamics could render them salient and cause organizational drift (Gregory et al., 2015).

Programs rely on two distinct ambidextrous leadership strategies to address salient paradoxical tensions: Blending and balancing (Gregory et al., 2015). Blending refers to finding harmonious combinations that persuade the organizations that a given set of two demands only seem paradoxical. This form of ambidextrous leadership is particularly well suited to address tensions within a program’s strategic orientation. Balancing refers to recurring compromises between opposing demands through dedicated coordination mechanisms and roles. This ambidextrous leadership strategy is particularly well suited to address tensions between the global objectives of the program and the local objectives of the projects. Both blending and balancing strategies are employed top-down by the program, with only indirect consideration given to the projects. This unilateral focus on the ambidextrous leadership through the program is likely to be problematic for restoring sustainable equilibria. It constantly relies on the projects to accept such top-down attempts to resolve tensions.

2.2 Equilibria Model of Paradox

To put ambidextrous leadership into perspective, it appears instructive to revisit the more general approach of addressing paradoxical tensions as described in the dynamic equilibrium model of organizing by Smith and Lewis (2011). According to them, paradoxes describe a tension “between interdependent elements” (Schad, Lewis, Raisch, & Smith, 2016, p. 6) that seem logical in isolation but inconsistent and even absurd when juxtaposed (Lewis, 2000). For example, exploitation and exploration seem meaningful in isolation as they compete for scarce resources (He & Wong, 2004). However, they reinforce each other, as companies need to acquire new knowledge through exploration to exploit it. On the other hand, exploration requires basic knowledge of processes, technologies, and products, which in turn requires exploitation (Andriopoulos & Lewis, 2009; Smith & Tushman, 2005). There are four types of paradoxical tensions in organizations (Smith & Lewis, 2011): Performing tensions arise when defining what to do and what not to do, organizing tensions when defining how to do and how not to do, belonging tensions when defining who does what and who does not, and learning tensions when defining the time horizon. However, these tensions only become salient when environmental factors or cognitive efforts accentuate the dualities’ oppositional and relational nature (Smith & Lewis, 2011).

Paradoxical tensions are either latent or salient. Latent tensions are an integral part of organizations and are likely to remain “dormant, unperceived, or ignored.” (Clegg, da Cunha, & da Cunha, 2002; Smith & Lewis, 2011). However, if paradoxical tensions become salient, they trigger vicious or virtuous cycles (Lewis, 2000). Vicious cycles lead to defensive mechanisms, such as denial, repression (Vince & Broussine, 1996), and humor (Hatch & Erhlich, 1993). In contrast, virtuous cycles lead to creativity and create opportunities (Beech, Burns, de Caestecker, MacIntosh, & MacLean, 2004). In these cases, paradoxical tensions are resolved by iterating between splitting to choose either pole and integrating to find synergies among them. Ideally, such a resolution restores sustainable equilibria (Smith & Lewis, 2011).
2.3 Ripple Effects of Ambidextrous Leadership

Prior research on paradoxes in IT transformation programs has contributed to our understanding of how programs can resolve paradoxical tensions through ambidextrous leadership (e.g., Gregory et al., 2015). However, said research has mainly focused on the program level and has only, indirectly, considered the role of projects and their members. This focus has understudied the potential interactions between program-level (re-)design (i.e., blending) and its implementation at the project level (i.e., balancing). Moreover, addressing strategic tensions through program changes may render latent tensions between the program and the project level salient. The evidence of ripple effects shows that ambidextrous leadership may not solely restore sustainable equilibria but also call for additional means.

Potential triggers for ripple effects are belonging tensions between the individual and the collective, as well as tensions between their competing values, roles, and memberships (Smith & Lewis, 2011). These belonging tensions may arise between projects and programs as projects are part of a program but act as separate entities (Jiang et al., 2018). Prior research on program and project management does not address whether changes at the program level can make project members experience a lack of belonging to the program. This may result in the creation of ripple effects.

3 Method

Our team conducted a longitudinal ethnographic study of the Omni Channel Experience (OCE) IT transformation program at SwissTelco. OCE was initiated in November 2017 to provide SwissTelco’s customers with the best online experience in a highly competitive and rapidly changing business environment. SwissTelco faced increasing aggressive competition in the saturated market and therefore had to cut costs to remain competitive. To this end, OCE aimed to integrate the different online customer touchpoints into one while reducing overall costs. This integration became necessary as the boundaries between products and processes became increasingly blurred. The need for a more consistent and holistic online experience arose.

OCE was suited for our study because its complexity as a major IT transformation program involving five projects (i.e., Alpha, Beta, Gamma, Delta, and Epsilon) promised various paradoxical tensions. To uncover these tensions, together with their triggers and consequences, we chose to closely observe the program and its projects in an ethnographic study. The first author conducted in-depth observations of OCE from June 2018 to June 2019 to understand the program and its projects, the stakeholders involved, and the business environment in detail (Myers, 1999). Only by closely observing the individual project and program members it was possible to uncover paradoxical tensions between the program and the projects over time and develop an understanding of how to address them. This close involvement also allowed us to sample interviewees within the program and projects to gain additional insights.

3.1 Data Collection

We collected three types of qualitative data: In-depth observations, interviews, and archival data. The in-depth observations from June 2018 to June 2019 were under the first author. She was in the OCE program organization two days a week, resulting in 750h of observation. The close relationship between the first author and the OCE stakeholders involved, which already existed before data collection, facilitated observations of formal meetings, informal interactions, and routine work activities at both the program and project levels. These extensive observations formed the basis for identifying tensions along with their triggers and consequences. We conducted 22 in-depth, semi-structured interviews (Table 1) with purposefully sampled stakeholders at both the program and project levels (Yin, 2018). These interviews aimed to explore the observed behaviors further and thus complement the insights from the observation. For this purpose, we interviewed stakeholders only when we needed additional information. Particularly informative were the interviews with the People Managers, responsible for staffing the projects and thus involved in OCE from the beginning, and the Project and Program Engineers, in charge of the organization and cooperation. The interviews lasted from 30 minutes to 1 hour, with an average of 45 minutes. Following Myers and Newman’s (2007) recommendations, all interviews were conducted
in the interviewees’ native language (i.e., German) using a semi-structured interview guide and took place on-site or using videoconferencing applications. Every interviewee was questioned about their professional background, their role within OCE, and the events and changes they observed before and during the observation. These questions allowed us to uncover the triggers for tensions and patterns of managing those tensions. We continuously adapted our interview guidelines to reflect the insights from the observations and preceding interviews. Table 1 provides an overview of the interviewees. To further triangulate, we gathered archival data, including internal documentation (i.e., visualizations of strategic initiatives, business cases, reports, key performance indices, and internal wiki pages), and communication (i.e., chat protocols, blog posts, and presentations of strategic initiatives to employees). This archival data helped us understand events and changes even before the first author entered the program. Table 2 provides an overview of the gathered data.

<table>
<thead>
<tr>
<th>Level</th>
<th>Job Title</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project Engineer Alpha</td>
<td></td>
<td></td>
<td>55 minutes</td>
</tr>
<tr>
<td></td>
<td>Project Engineer Beta</td>
<td>50 minutes</td>
<td>30 minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Engineer Delta</td>
<td>30 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Architect Beta</td>
<td>45 minutes</td>
<td>60 minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>People Manager Alpha</td>
<td>30 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>People Manager Beta</td>
<td>30 minutes</td>
<td>60 minutes</td>
<td>30 minutes</td>
</tr>
<tr>
<td></td>
<td>People Manager Gamma</td>
<td>30 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>People Manager Delta</td>
<td>55 minutes</td>
<td>30 minutes</td>
<td>45 minutes</td>
</tr>
<tr>
<td></td>
<td>People Manager Epsilon</td>
<td>30 minutes</td>
<td>45 minutes</td>
<td>60 minutes</td>
</tr>
<tr>
<td>Program</td>
<td>Program Engineer</td>
<td></td>
<td></td>
<td>55 minutes</td>
</tr>
<tr>
<td></td>
<td>Program Architect</td>
<td></td>
<td></td>
<td>60 minutes</td>
</tr>
<tr>
<td>Portfolio</td>
<td>Transformation &amp; Change Agent</td>
<td>45 minutes</td>
<td></td>
<td>55 minutes</td>
</tr>
</tbody>
</table>

* Rounds of interviews (we conducted a total of three interview rounds)

Table 1. Gathered interviews

<table>
<thead>
<tr>
<th>Empirical Data</th>
<th>Units</th>
<th>Main Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation</td>
<td>~ 750 hours</td>
<td>Identify significant changes and tensions</td>
</tr>
<tr>
<td>Interviews</td>
<td>22 interviews with an average duration of 45 minutes</td>
<td>Understand tensions and patterns of managing those tensions</td>
</tr>
<tr>
<td>Other material</td>
<td>Visualizations and presentations of strategic initiatives, business cases, reports and measurement boards, internal wiki pages, chat protocols, blog posts</td>
<td>Understand triggers of changes and tensions that emerged before the period of observation</td>
</tr>
</tbody>
</table>

Table 2. Gathered data

### 3.2 Data Analysis

Our data collection and analysis approach was iterative, allowing us to respond to novel insights by continually refining our data collection instruments. To analyze our data, we relied on NVivo 11. Throughout the analysis process, the collected data from multiple sources, levels (i.e., program- and project-level), and points in time offered various triangulation opportunities (Eisenhardt, 1989). We followed an iterative bipartite approach for coding our data, as suggested by Charmaz (2006). First, we coded each piece of data line-by-line and in extensive detail. This line-by-line analysis allowed us to curb the human tendency of making conceptual leaps and orienting the coding along with extant theory rather than developing entirely new categories from the ground up (Charmaz, 2006). Second, we explored relationships between our initial codes to identify general patterns across the organizational...
entities and what changes arose over time. For this purpose, we engaged in axial coding (Charmaz, 2006) and systematically compared the dynamics within and across cases using replication logic, memo writing, and tables (Miles, Huberman, & Saldana, 2013). This systematic comparison led to an understanding of the meaning of identity within programs and projects and the identification of five patterns and their underlying theoretical mechanisms (Charmaz, 2006). Figure 1 illustrates our data structure as proposed by Gioia, Corley, and Hamilton (2013).

<table>
<thead>
<tr>
<th>1st Order Concepts</th>
<th>2nd Order Themes</th>
<th>Aggregate Dimensions</th>
</tr>
</thead>
</table>
| • Standardized user interfaces vs. self-contained systems  
  • Customers’ product vs. process experiences  
  (Products considered as parts of processes) | Strategic Tensions | Latent Tensions Turned Salient |
| • Project members resist program objectives  
  • Project members identify with self-contained systems as opposed to standardized user interface | Belonging Tensions | Resolving Approaches for Salient Tensions |
| • The project introduces a decoupling layer  
  • The project aligns with the program objectives | Unilateral Resolution | |
| • The project leaders discuss how their topics and areas of responsibility could converge  
  • The project members acknowledge their common objective and no longer pursue five separate objectives | Mutual Resolution | |

Figure 1. Data structure

4 Results

In November 2017, SwissTelco established the OCE program to provide the best user experience at all online touchpoints. OCE was assigned four projects, each responsible for one system: Alpha was assigned to the customer care tool, Beta to the customer relationship management system, Gamma to the ecommerce platform, and Delta to SwissTelco’s homepage. Every project was composed of six to fifteen teams and approximately 120 members.

The biggest challenge for OCE was to standardize the four IT systems. To this end, OCE imposed a specific vision on each project that described their contribution to the overall objective of the program. Furthermore, OCE introduced uniform roles, structures, and processes to ensure alignment and synchronization. The uniform roles included a Program Manager, a Program Engineer, and a Program Architect. The Program Manager was responsible for prioritizing the work assigned to the program. The Program Engineer was responsible for improving the program’s efficiency by making adjustments in the projects’ organization and setting standards for cooperation. The Program Architect was responsible for the program’s IT architecture. This role aims to increase the quality of all IT systems involved by setting technical standards and improving interaction. OCE identified similar roles at the project level (i.e., Project Managers, Project Engineers, and Project Architects) and identified People Managers responsible for skill development and other human resource concerns.

4.1 Introduction of a New IT Architecture at the Program Level

In early 2018, OCE realized that it could not meet its objective of providing the best user experience to its customers. Although the customers expected the same user experience for all IT systems, they faced different ones. In other words, there was no unified online user interface to access SwissTelco’s products, services, and support. This variety of user experiences was because the projects were still
focused on their self-contained IT systems. The individual project members had difficulty realizing that the best user experience hinges on the program as a whole, and thus, on the user experience on all IT systems in interaction. OCE was facing a strategic tension: the program’s objective was to achieve standardized user interfaces, but the projects’ objectives were still pursuing differentiated user experiences on their self-contained IT systems. To resolve this tension, the Program Architect unilaterally decided to introduce a new, program-wide IT architecture for all IT systems. This new IT architecture aimed to standardize the interfaces between the IT systems and the databases by implementing a decoupling layer, allowing OCE to introduce a standard user interface and monitor user experience on all IT systems.

The projects and their members resisted the unilaterally imposed IT architecture. This unilateral decision made the projects aware, for the first time, that an entity above them had the authority to govern them and influence their agendas. However, this new IT architecture was problematic in that it undermined the modular IT systems’ advantages. The projects considered it their core competence to develop their IT systems while managing the complex processes and interfaces to the others: “Complex ordering processes are our core competence.” (Project Architect Beta)

Despite project resistance, the competitive environment demanded rapid implementation of the new IT architecture—the user experience had to be improved as quickly as possible. Project resistance and the fact that current IT systems had to be retained to ensure operations jeopardized a speedy implementation. For these reasons, the Program Engineer decided to launch a fifth project and commission it with implementing the new IT architecture. This fifth project, Epsilon, was set up for a limited time. For the Program Engineer, this was the only way to maintain the current IT systems with their architectures and simultaneously introduce the new program-level IT architecture.

Regarding the implementation time, the Program Engineer proved to be correct—within 13 months, Epsilon implemented the decoupling layer and the standard interface of the new IT architecture:

“To build the decoupling layer, we need project Epsilon. Project Epsilon was the only way to separate the new business from the old [...] Within just 13 months, we built the new IT architecture and kept our legacy IT systems up and running. That went very fast, in my opinion.” (Program Engineer)

However, with Epsilon, the knowledge about the new IT architecture became encapsulated, while the knowledge about the existing IT systems remained limited to the other projects. To address this encapsulation and its potentially detrimental effects, the Program Engineer aimed to promote a knowledge transfer between the four main projects and Epsilon. The Program Engineer appointed teams and members from the four other projects to Epsilon to blur the boundaries between them. It became an arduous task to encourage the teams and the members of the projects Alpha, Gamma, and Delta to transfer knowledge with project Epsilon:

“The lack of acceptance [of Alpha, Gamma, and Delta] meant that too few resources were put into this new approach.” (Program Engineer)

The intervention by the Program Engineer contradicted the projects’ understanding of autonomy. Another obstacle was that this transfer of knowledge was hardly compatible with the projects’ goals of focusing on specific IT systems.

The autonomy of the members of project Beta brought different outcomes. Unlike the other projects, Beta recognized the need for a knowledge transfer with Epsilon. Early on, Beta’s Project Engineer turned to Epsilon’s Project Manager to align their goals. This cooperation resulted in a joint roadmap, which came into force in January 2019. This roadmap allowed both projects to synchronize their releases and tests without competing with each other:

“We are in the same boat, and nobody got the feeling that the other one wants to take over [...] In the end, it was not their desire [of the program] as we did it of our own accord.” (Project Engineer Beta)
Over time, this cooperation enabled Beta to adapt its IT system to the new program-level IT architecture and the standard user interface. In contrast, the knowledge transfer of the three other projects with Epsilon did not proceed as the program had initially anticipated. By March 2019, knowledge about the new IT architecture and the standard user interface was mainly with Epsilon. Contrary to what the program envisioned, Alpha, Gamma, and Delta continued to focus on their IT systems’ development and operation. This situation caused challenges, especially for the Program Engineer:

“The know-how [about the new IT architecture] is now practically exclusively with project Epsilon. That was the price we had to pay for the chosen implementation speed. But now it is time to reduce the efforts in the as-is world.” (Program Engineer)

Although Epsilon’s isolation initially favored a fast implementation, the negative consequences for the intended IT transformation became apparent. The implementation of the new IT architecture through a dedicated project now made it difficult to reintegrate the four main projects:

“The challenge is to invest more in the new and less in the legacy IT systems.” (Program Engineer)

To mitigate the risk, the Program Engineer decided to end project Epsilon in April 2020. Alpha was only marginally affected by this decision—there was little need for adjustments within their IT system. Unlike the other three initial projects, Beta had a joint roadmap with Epsilon for 15 months, in which they achieved mutual goals. For Gamma and Delta, the end of Epsilon had much more severe consequences. They needed to replace large parts of their IT systems, but due to the lack of knowledge transfer with Epsilon, they lacked the necessary know-how. To address this issue, the Program Engineer approached Gamma and Delta to find solutions. It quickly became apparent that Epsilon members would need to be assigned to either Gamma or Delta to bridge their lack of expertise. Besides, the Program Engineer gave Gamma and Delta responsibility for the final implementation of the new IT architecture and IT system:

“Over the last weeks, the management teams of projects [Gamma and Delta] met twice with Product Owners and some Scrum Masters of Gamma, Delta, and Epsilon. They discussed how their topics and areas of responsibility might converge.” (Project Engineer Gamma)

4.2 Blending Programs to Meet Customer Expectations

Throughout OCE, the expectations of SwissTelco’s customers constantly changed with increasing frequency. These changes were particularly apparent when it came to the online experience, the main objective of OCE. The more the customers made less use of physical touchpoints (i.e., stores or call centers), the more they focused on the online world. In consequence, the boundaries between products, services, and process design became more blurred. The online experience became an integral part of product and service design from the customers’ perspective as it increasingly influenced products and services. Due to these observations, OCE realized that a user-friendly ordering process could have positive effects on the purchase decision for products or services:

“The customer experiences the product within a process.” (People Manager Delta)

The program was facing a strategic tension between its program objectives and customer expectations because OCE was unable to meet customers’ ever-changing expectations. In building the best online customer experience, OCE focused almost exclusively on processes that ranged from the discovery of a product or service by customers to resolving problems with those products and services. In other words, OCE’s innovation efforts focused almost exclusively on process design, while another program was in charge of product and service design. Over time, this separation led to more challenges concerning the ever-changing customer expectations and behaviors. In particular, the coordination and knowledge transfer between the programs severely delayed time-to-market. The repercussion was more work for OCE and made product and service design more expensive. In addition to these problems, the separation
also limited cross-pollination, which led to conflicting developments. For example, while OCE’s objective was to make ordering processes as user-friendly and straightforward as possible, the product and service design program aimed to offer customers many configuration and combination options. The programs often fought each other to achieve their objectives, which meant they were often unable to meet customer expectations.

These shortcomings were perceived at both the program and the project levels, especially by Gamma and Delta. For Gamma and Delta, the separation of process and product design in two programs meant that they depended on inefficient coordination processes between these programs. These projects relied on the Program Manager or Program Engineer to transfer knowledge, ideas, or impediments. The projects were relieved when the Program Engineers and managers agreed to work with them to address these deficiencies because the projects had raised the awareness of these shortcomings for a while back.

In June 2019, the Program Manager and the Project Managers of the most affected projects, Gamma and Delta, agreed to integrate parts of the product and service design into OCE. This strategic realignment not only reduced interfaces but also made knowledge transfer and coordination much easier and faster. The project stakeholders could now rely on the standardized planning processes and prioritization rules within the program. This integration increased efficiency while making innovation more accessible. The projects now combined product and service design objectives to define processes for building the best online customer experience, which allowed for mutual goal setting. In other words, process managers could now contribute to easy-to-use and straightforward products and services, while product managers could now contribute to user-friendly processes:

“Gamma and Delta are very product and service-oriented. With the extension of both projects, it was now possible to bridge the gap between market demand and the building of the appropriate online customer experience.” (Program Engineer)

The projects and their members reacted with cautious enthusiasm to this change. In the end, the extended projects helped them achieve their objectives while solving the prior problem of having to rely on the programs to coordinate with the product- and service development projects:

“We are geared towards a common objective. Therefore, we no longer pursue five goals simultaneously.” (People Manager Delta)

5 Analysis

OCE faced two strategic tensions that were or became salient during the observation period. The main trigger for these tensions was the need to improve cost efficiency while increasing customer satisfaction. For addressing these tensions, program management relied on ambidextrous leadership. However, although ambidextrous leadership proved successful in resolving these tensions, it also triggered previously latent tensions between the global program and local project objectives to become salient.

5.1 Ambidextrous Leadership as a Trigger for Ripple Effects on Tensions Between Global Program and Local Project Objectives

The first strategic tension became salient in 2018, just after the establishment of OCE. By that time, OCE observed that the projects’ IT architectures prevented standardization at the program level to a growing extent. In response to this tension in 2018, the program management team decided to adapt its IT architecture for standardizing the interfaces. The new IT architecture introduced a standard user interface and better monitoring of the four projects. The Program Architect of OCE unilaterally blended the demands for differentiation on the project level and standardization on the program level by introducing a decoupling layer. This solution combined the program’s need to standardize interfaces among IT systems, user interfaces, and monitoring while allowing the projects to maintain a certain degree of freedom within their IT systems. While this unilateral blending resolved tensions at the
program level, its benefits tended to remain hidden at the project level. Consequently, latent tensions between the global program and local project objectives became salient. In particular, the projects and their members felt patronized by OCE and felt unable to identify with the new strategy. They were also excluded from the decisions to integrate the new IT architecture. This paternalism gave the project members the impression that they primarily belonged to the individual projects and were not necessarily part of the overall program. The projects and their members realized that OCE could unilaterally undermine the projects’ autonomy for the sake of program control. This tension caused the individual projects to offer more resistance to the program, resulting in an organizational drift that again called for a resolution. Based on this insight, we found:

**Pattern 1a:** Ambidextrous leadership to resolve salient strategic tensions triggers previously latent tensions between the global program and local project objectives to become salient, provided that the project members do not feel belonging to the program.

The second strategic tension became salient in June 2019. By that time, OCE realized that its focus on innovative processes to provide the best user experience throughout every SwissTelco online customer access point was overly detached from SwissTelco’s product and service development. Since OCE placed the most significant emphasis on providing the most innovative processes, it was mostly detached from the product and services offered by SwissTelco. Therefore, OCE had to put an increasing effort into coordinating with the program responsible for product and service development. In response to this tension, the program management decided, together with the Project Managers and Engineers, to merge parts of the program responsible for product and service development with Gamma and Delta. Unlike OCE’s decision to unilaterally adapt its IT architecture, this time, the program management team decided together with the affected projects. Together, they determined the scope of the reorganized projects in a participatory approach that allowed project members to contribute their ideas and experiences to the blended solution design. The program and the projects solved a perceived problem, allowing the projects and their members to identify with the new strategy. In doing this, the projects and their members felt a sense of belonging to the program. Based on our analysis, we identified the following pattern:

**Pattern 1b:** Ambidextrous leadership to resolve salient strategic tensions does not trigger previously latent tensions between the global program and local project objectives to become salient, provided that the project members feel belonging to the program.

### 5.2 Establishing Sustainable Equilibria Between Global Program and Local Project Objectives

Given the salient tensions between the global program and local projects’ objectives that emerged through ambidextrous leadership, we identified three different patterns. The first pattern describes a unilateral resolution by OCE’s management. In detail, OCE unilaterally resolved the emerging strategic tension between its global objectives without involving the projects with their local objectives. This conflict of interest was evident in 2018 when OCE unilaterally decided on the new IT architecture. At that time, the OCE Program Engineer chose to commission project Epsilon to implement the new IT architecture. This decision made it possible to accelerate the overall implementation without investing too much time into integrating all projects beforehand. However, this decision also meant that the four initial projects Alpha, Beta, Delta, and Gamma, continued to experience patronizing attitudes from the program. They were neither part of project Epsilon nor involved in its formation. This patronizing environment led to organizational drift away from the global program objectives towards realizing their local projects’ objectives. In other words, the ambidextrous leadership of OCE rendered new tensions between the global program- and the local project-level objectives salient.

The Program Engineer of OCE decided to involve selected teams and project members in Epsilon to end the growing tensions between its global program’s interests and the projects’ local interests. This
unilateral decision aimed to enable a transfer of knowledge between Epsilon and the four initial projects. These projects developed and operated the IT systems that were to be integrated to or replaced by the new IT architecture. In other words, the program management team reacted by balancing its global interest in implementing the new IT architecture while simultaneously maintaining the initial projects’ local interests in achieving their local IT system-oriented objectives. However, as the initial projects were once again excluded from this decision and were still focused on meeting their local objectives, they hardly listened to their teams and project members that had become part of Epsilon.

To make matters worse, the insights these teams and members brought in frequently collided with their project’s objectives. Fortunately, for project Epsilon, this integration of teams and project members had a positive effect, as it allowed vital insights into the four projects. In conclusion, although the four initial projects and their members experienced a lack of belonging to the program, project Epsilon was able to advance the new IT architecture for the program. Based on our analysis, we thus found the following pattern:

**Pattern 2a:** Unilateral resolutions by the program can temporarily balance paradoxical tensions on the global program level, but they cannot sustainably restore the projects’ sense of belonging.

The second pattern was the unilateral resolution of emerging belonging tensions within the projects. In this case, the projects adapted their local objectives to the anticipated program’s global objectives. This reaction was evident in the case of Beta. Unlike the other projects, Beta had recognized the potential benefits of adapting its objectives to project Epsilon’s objectives. By January 2019, Beta adapted its project objectives to project Epsilon. Beta and Epsilon’s Project Managers and engineers defined a roadmap together to synchronize releasing and testing. This roadmap allowed Beta to shift its efforts to the new IT architecture and enabled its members to identify with the program strategy. As a result, Beta resolved the belonging tension sustainably. This was accomplished by balancing its local interest to develop and operate its standalone IT system and the program’s global interests to implement the new IT architecture. Based on our analysis, we thus found the following pattern:

**Pattern 2b:** Unilateral resolution by the projects can resolve paradoxical tensions between their local and the program’s global objectives and sustainably restore their sense of belonging.

The third pattern observed was a mutual resolution of emerging belonging tensions by the program and the projects. In this case, the program, together with the projects, adapted its objectives. We observed this reaction in two cases. The first instance was that the Program Engineer and the Project Managers and Project Engineers of Gamma and Delta mutually sought solutions to replace large parts of the existing IT systems. Nonetheless, knowledge transfer from project Epsilon was far from reaching completion. Part of the solution was to assign members of the previously completed project Epsilon to either Gamma or Delta to assure knowledge transfer. Another part of the solution was that Gamma and Delta took joint responsibility for implementing the new IT architecture and IT systems. As a result, OCE resolved belonging tensions together with Gamma and Delta to restore sustainable equilibria. The second case showed that belonging tensions between OCE and projects Gamma and Delta were resolved mutually, even before OCE employed ambidextrous leadership to achieve its global objective. OCE agreed with Gamma and Delta to integrate parts of the product and service design into OCE by blending the two seemingly paradoxical demands of the process and product design. In doing this, OCE, Gamma, and Delta proactively prevented latent belonging tensions from becoming salient. Based on our analysis, we found the following pattern:

**Pattern 2c:** Mutual resolution by the program and its projects can resolve paradoxical tensions between the program’s global objectives and the projects’ local objectives as the projects’ sense of belonging is restored sustainably.
6 Discussion

In our paper, we aim to answer the following questions: (1) whether and how does resolving strategic tensions by ambidextrous leadership create new tensions, and (2) how can such follow-up tensions be addressed to restore sustainable equilibria? Regarding the first question, we identified the decisive aspect of belonging. More specifically, individual project members identify themselves as part of larger entities such as projects or programs. Ideally, these members would sense that they belong to both the project and the program. In this case, tensions between the global program-level and the local project-level objectives remain latent. If, however, project members feel that they belong to the project but not to the program, these tensions are likely to become salient. Thus, if ambidextrous leadership reduces the project members’ sense of belonging to the program, it is likely that new tensions will become salient. Our findings suggest that project members will react with resistance to the program and are subject to organizational drift towards standalone project objectives.

Regarding the second question, we identified three different approaches with varying degrees of success in restoring sustainable equilibria. The first approach mirrors the unilateral ambidextrous leadership approach proposed in prior literature (Gregory et al., 2015). This approach only focuses on the program’s actions that only indirectly consider or estimate the projects’ objectives. Although this unilateral approach can establish short-term equilibria, it fails in restoring them sustainably. This approach fails because it cannot dictate either the projects’ or their members identity and cannot resolve the follow-up belonging tensions. The second approach mirrors the opposite of ambidextrous leadership in that it solely focuses on the unilateral resolution of follow-up belonging tensions by the projects and their members. Although this approach can restore sustainable equilibria, it is considered an interim solution by the concerned projects. Although this second approach fosters the concerned projects’ and their members’ sense of belonging to the program, it will not restore the sense of belonging to the program of other projects and other members that did not apply a unilateral resolution mechanism. The third approach implies a mutual resolution of emerging belonging tensions by the program and projects. In doing so, the program and the projects balance the program’s global and the projects’ local objectives to restore the sense of belonging sustainably. As a result, projects and their members adapt their identities in a compatible way within the program. This approach restores sustainable equilibria without collateral effects.

6.1 Theoretical Contributions

Our findings make new contributions to research on program management and organizational paradoxes. In terms of program management, our results add to prior research in two ways. First, our paper shows that ambidextrous leadership is not necessarily capable of restoring sustainable equilibria. In this way, our analysis expands on prior research that has provided rich insights into how paradoxical tensions could be resolved by ambidextrous leadership (e.g., Gregory et al., 2015). We show how resolving strategic tensions by ambidextrous leadership creates new tensions between global program-level and local project-level objectives. However, such tensions only become salient when the projects and their members no longer perceive a sense of belonging to the program due to unilateral ambidextrous leadership. In this case, projects are likely to focus on their objectives, increasingly drifting away from the overall program-level objective. This drifting phenomenon is highly problematic, as project and program objectives must be aligned for successful IT transformation. Interestingly, programs can proactively prevent such drifting events by including projects’ perspectives in ambidextrous leadership. In doing this successfully, projects and their members are likely to continue to experience a sense of belonging to the program. Thus, we show that ambidextrous leadership can have ripple effects on projects and their members in terms of belonging tensions that become salient.

Second, we identify three approaches for dealing with follow-up tensions between global program-level and local project-level objectives caused by ambidextrous leadership. In identifying those approaches, we extend on prior research focusing mainly on unilateral approaches (e.g., Gregory et al., 2015). Our results show that it is unlikely that ambidextrous leadership alone can restore sustainable equilibria.
Instead, we find that sustainable equilibria require the resolution of belonging tensions, which is hardly possible with unilateral ambidextrous leadership because projects and their members are unlikely to have their identity imprinted on them. Instead, programs can rely on the projects to adapt themselves and resolve salient belonging tensions or work with the projects to find solutions. At the same time, the former is seen by projects merely as an interim solution. While both approaches are in principle capable of restoring sustainable equilibria, only the latter leads to satisfied programs and projects. In conclusion, we show that ambidextrous leadership cannot restore sustainable equilibria as it cannot resolve salient belonging tensions.

In terms of organizational paradoxes, we build on prior research in other contexts with one-to-many relationships that emphasized the subordinate organizations’ importance for maximizing value co-creation (e.g., Hurni et al., 2020). The unique context of IT transformation programs allowed us to identify the importance of belonging. This topic has received little attention in prior IS literature. Our findings show that the key to successful IT transformation programs is integrative decision-making in strategic and execution efforts. Otherwise, belonging tensions are likely to become salient, preventing programs from achieving their objectives. In conclusion, we emphasize the importance of including projects in strategic and execution decisions to prevent belonging tensions from becoming salient.

### 6.2 Practical Implications

This paper has two important implications for practitioners. First, program management teams are often confronted with paradoxical tensions. One approach for resolving such tensions is ambidextrous leadership, either in its unilateral or mutual form. It is important to note that ambidextrous leadership has the potential of causing ripple effects as projects and their members could experience a lack of belonging to the program, resulting in follow-up tensions. To prevent follow-up tensions from becoming salient, programs must adopt an integrative approach by including projects and their members in their strategic and execution decisions. Only when programs have to adapt quickly to a changing environment tensions should be consciously accepted. This action enables fast adoption of the IT transformation program without spending time negotiating with the projects. A downside of accepting a follow-up tension is that it may endanger the program’s ambidexterity by causing organizational drift within the program and resistance among the program management team and project members.

Second, ambidextrous leadership can trigger follow-up tensions. In these cases, programs can rely on the projects to adapt themselves and resolve salient belonging tensions or work with the projects to find appropriate solutions. Those responsible within the programs must be aware that unilateral decisions only provide a short-term remedy. Much more sustainable is when tensions are resolved jointly by the programs and projects. Only this mutual approach can restore sustainable equilibria and prevent further follow-up tensions from emerging.

### 6.3 Limitations and Future Research

Our paper faces three main limitations. First, our study builds on an ethnography of one single IT transformation program, using in-depth data from observations, interviews, and secondary data. Future research could potentially investigate whether our findings apply to other programs. Second, although the first author observed the program over one year, the long-term consequences may remain undiscovered. Future research could therefore investigate programs over an even more extended period. Third, although this study identified critical patterns for whether and how ambidextrous leadership can have ripple effects and how the resulting follow-up tensions could be resolved, it did not put these patterns into context to explain the underlying process. Future research could attempt to derive a process theory for explaining how to restore sustainable equilibria.
References


Resolving Belonging Tensions in Programs


