Pure Coding Pleasure: How BMW Involves App Developers in the Design of Automotive Onboard APIs

Completed Research

Niklas Weiss  
Technical University of Munich  
Boltzmannstraße 3, 85748 Garching  
niklas.weiss@tum.de

Manuel Wiesche  
Technical University of Munich  
Boltzmannstraße 3, 85748 Garching  
wiesche@in.tum.de

Helmut Krcmar  
Technical University of Munich  
Boltzmannstraße 3, 85748 Garching  
krcmar@in.tum.de

Abstract

Digital transformation requires incumbent companies to accelerate the development of their digital products. The automotive industry is a prominent example of this change. Even though, research frequently considers digital transformation of organizations, there are rare perceptions on the change of the actual technology. This action research study provides deep insights in the endeavors of the global operating car manufacturer BMW towards a generic design of onboard application programming interfaces (APIs) which should enhance accelerated development of digital products. To help the firm embrace generativity, we therefore infused lead user involvement theory and API evaluation criteria into the API design team. As a result, we present 5 majorly refined APIs, which are implemented in the BMW App SDK. Further, we identified critical challenges and benefits for the involvement of lead users in the design of enabling technology within an incumbent company.

Keywords

API design, lead user involvement, action research, incumbent firms, digital platforms.

Introduction

Motivation

Digital transformation affects incumbent enterprises. It challenges firms from multiple domains (Schreieck and Wiesche 2017; Schreieck et al. 2018). The automotive industry is one of the most prominent examples of this development. The acceleration of digital product development is one major change that has to be faced by the car manufacturers (Svahn et al. 2017). Several car manufacturers have introduced systems of deploying new versions of the vehicle’s software over the internet (AutomotiveWorld 2019). The system enables regular update of the existing software and the deployment of new digital products to a car even after the car has been sold to a customer. Furthermore, the manufacturers integrate service-oriented software systems in the form of digital platforms into their vehicles to enable flexible deployment of digital products offered in the form of apps (BMW 2018; Digitaltrends 2017). In this context, we understand digital platforms as “extensible codebase that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate” (Tiwana 2014). Thereby, after the initial hand-over to the customer, the codebase in the car can be extended. In this way, the modular extensibility of the system decouples the short-term app development process from the long-term vehicle development process. Already, in a traditional automotive development approach, different software modules interoperate via
application programming interfaces (APIs). These APIs expose functionality of one software module to another. However, the utilization of an API is defined in advance. Additionally, the number of an API’s consumers is limited to a small number of experts and the primary criterion for assessing an interface is its functionality. The introduction of a modular extensible software architecture that furtherly enables the deployment of unknown and new utilizations of an API in the future, entails different requirements. The capabilities and limitations of an API need to be transparent to all stakeholders including potential app developers without any domain specific expert knowledge (Pühler 2011; Schlachtbauer et al. 2012). Hence, incumbents need to refine their existing API design. This practical challenge raises this question; how can an incumbent organization evolve its existing system design in the context of digital transformation?

Previous research reveals that multiple traditional companies has failed to transform their technology adequately (Henderson and Clark 1990; Tripsas and Gavetti 2000). Furthermore, literature considers the involvement of users as essential success factor for designing new information systems (IS) (Bano and Zowghi 2014; von Hippel 2005). Therefore, we propose the approach of lead user involvement for the refinement of an existing technology, in this case incumbent APIs for an automotive onboard software system. The evolvement of an existing system entails that this system is already in use in practice. A certain amount of lead users may have identified potential improvements for existing solutions (Lüthje and Herstatt 2004; von Hippel 1986). In the case of automotive onboard APIs, app developers that already implemented such APIs embody that kind of lead users. This study strives to identify the challenges and benefits that emerge in the involvement of lead users during the analysis and refinement of automotive onboard APIs. For doing so, we applied an action research approach (Baskerville 1999; Frank et al. 1998; Keng and Rossi 2011) within the software development department at the global car manufacturer BMW.

In the remainder of the paper, first, we described the theoretical concepts of APIs and lead user involvement. Then, we described BMW’s infotainment system, followed by a detailed description of the research project setup. Subsequently, we described the detailed API analysis and design process. Finally, we presented our results, findings, and discussions.

## Background

### Relevance of API Design

Application Programming Interfaces (APIs) exposes a system’s core resources as a service to stimulate generativity (Henfridsson and Ghazawneh 2013). The usage of an APIs does not have to be determined by design but can be utilized in multiple ways. In this way, APIs are able foster innovation by enabling the development of complementary features (Um et al. 2013). Similarly, the utilization by app developers affect the design of the API (Eaton et al. 2015). APIs potentially generic character enables scalability of operations as well as flexibility in acquiring new strategic partners and realizing new business goals (Iyer and Subramanian 2015). However, the actual design of APIs is critical to maximizing its potentials. Poor API design results in increased development costs during its implementation by apps (Henning 2009). If these costs exceed potential benefits of a complementary feature, it will not be created. In this way innovation is blocked and the attractiveness for end users remains constant or even decreases (Tiwana 2014).

### Lead User Involvement

Lead user involvement is a principle often used in system design research. To reduce the risk of failure, the alignment of product development activities with the needs of actual and potential users is crucial (Jaworski and Ajay 1993). A user-centric focus fosters quality, reliability and uniqueness of a product (Li and Calantone 1998). The involvement of users already in early phases of an innovation project enhances these potentials (Herstatt and von Hippel 1992; von Hippel et al. 1999). In their comprehensive literature review, Bano and Zowghi (2014) identify five relevant perspective of user involvement. The psychological perspective considers aspects as the users’ motivation or interests to participate. Second, the involvement of users requires appropriate management. Moreover, the political perspective considers the degree of power that is given to the involved users. The purpose of user involvement can differ for various groups of users. Therefore, cultural aspects need to be considered. Finally, different intensities of user involvement require specific methodological approaches. The concept of lead user involvement originally is rooted in marketing research, and it considers the involvement of users whose present strong needs will become general in the future (von Hippel 1986). Lead users are well-qualified and motivated to contribute to an
improvement of the status quo (von Hippel 1986). Their prevalent own need enables them to innovate (von Hippel 2005). Since lead users embody the leading edge of a market regarding important market trends; their participation in product development activities facilitates innovation, attractive for future users. In our study, we consider lead users as app developers who implemented the exposed interfaces at a large scale or multiple times. Lead user involvement is mainly considered for enhancing innovation. However, von Hippel (2005) emphasizes that many of the concepts regarding innovations communities “apply to information communities as well.” Considering the app developer community as an information community which utilizes APIs, we strive to involve their expertise in the refinement of already existing APIs.

The BMW Case

Initial Situation

This study considers APIs of a digital platform for onboard automotive apps of a global car manufacturing company, BMW. The platform is part of the BMW OS 7.0, the company’s latest infotainment system, released in July 2018 with the release of the latest BMW X5 series. Upcoming models from the manufacturer will run the system. The car’s central electronic control unit powers the digital platform, and it also enables modular wireless deployment of apps to the digital platform. By exposing multiple functionalities of the car via APIs, the platform provides the base for a broad spectrum of use cases. During the release of the platform in summer 2018, more than 20 apps were available, providing services such as a parking lot finder1, music streaming2, Microsoft Office 3653 and apps for different BMW service calls4. The number of available apps is steadily increasing since the initial release. Although the platform was not opened towards third-party developers, multiple stakeholders around the globe are involved in creating apps for the platform. The app developer community is made up of over 120 active members.

The first author of this study is actively involved in the platform’s development team as a Ph.D. researcher. In the course of this study, we interviewed eight expert app developers from October until December 2017, and the goal of the interview was to identify challenges at the emergence of the platform. The result of the investigation showed that the platform’s APIs has room for significant improvement. The developers pointed out two fundamental issues with the platform’s APIs. First, the design of the API was tailored for a single or small amount of use cases.

“Sometimes it is hard to understand how and why an interface is designed in the way it is. For example, one specific value – which is actual available somewhere in the car - is just missing in the interface while in other cases one value is available in three different ways. However, most probably the missing value was simply not required by the initial use case while in the other case there were three different use cases, which required the value in three different ways.”

Second, the design of the interface was decided by the API provider and the party requesting for the use case. The interest of requesting party is speedy delivery with a minimal budget. The evaluation criteria for the API is the feasibility of the use case while usability is subsidiary. However, abstracting usability was not considered due to the small number of consumers.

“The implementation of some APIs requires massive efforts. You need to write tons of boilerplate code which don’t reveal any functionality towards the application. For example, the implementation of a notification banner in the UI requires more than 350 lines of code.”

Project Setup

The platform team reviewed the result of the interview conducted and decided to start a project to create new and high-quality APIs in the platform SDK. The project was conducted in research cooperation,

4 https://www.bmw-connecteddrive.de/app/index.html#/portal/store/Base_AssistNoTPEGOffer from 19.02.2019
applied action research methodology. This approach allowed the active involvement of the researcher directly in the project and enables deep insights into the actual design process. Hence, the first author of this study took the role of an active architect who was responsible for designing and refining APIs in an iterative approach while an internal BMW engineer was the project lead. This setup embodied the client-system infrastructure, which is required by any action research approach (Baskerville & Wood-Harper, 1998). The project started on March 15, 2018 and ended December 21, 2018.

Besides the establishment of a robust client-system infrastructure, the principles of action research require a sound theoretical foundation of the applied approach (Davison, et al. 2004). Even though, the conducted interviews revealed the demand for action, a more profound approach was required for the actual evaluation of the APIs. For this reason, we derived general evaluation criteria for good API design from prevalent literature. Therefore, we conducted a comprehensive review of literature on API design and coded all identified papers. The analysis revealed a large number of different characteristics of good API design. However, four criteria stand out as being named in most of the considered studies: Simplicity, Documentation, Usability, and Tutorial/Sample Code (see Table 1). Hence, our study does not claim to consider all relevant API characteristics but the most relevant. The identified criteria are not exclusive to each other, nevertheless each aspect has unique characteristics which are explained in the Description column in Table 1.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Source</th>
<th>Evaluation Grades</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplicity</td>
<td>Bhaskar et al. (2016), (Bloch 2006), (Myers et al. 2016)</td>
<td>Low</td>
<td>It is hard to create basic objects and even to trigger basic operations. There is much unclear overhead that makes no sense from a functional perspective.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td>The usage is generally possible. However, there is still some overhead and tacit knowledge needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>The API can be used as it is. With some basic knowledge in software development, it is easy to trigger basic functionalities, and the API behaves as “expected.”</td>
</tr>
<tr>
<td>Documentation</td>
<td>(Bhaskar et al. 2016), (Burns et al. 2012), (Lee et al. 2014)</td>
<td>Low</td>
<td>There is no documentation at all, or there is only some documentation that generates no valuable insights for the developer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td>Some or a big part of the functions are documented. However, use-case oriented usage is unclear. For example, a developer does not know the order of functions calls.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>The API is thoroughly documented (each function and property) and contains necessary additional information (e.g., flow charts or sample usages).</td>
</tr>
<tr>
<td>Usability</td>
<td>(Bhaskar et al. 2016), (Bloch 2006), (Zghidi et al. 2017)</td>
<td>Low</td>
<td>The naming and usage of types are inconsistent. The API does not conform to the coding guidelines of the respective programming language or environment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td>There are some inconsistency issues, but generally, the API is consistent and fulfills the usability compliance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>There are no inconsistencies, and the API conform to usability compliance.</td>
</tr>
<tr>
<td>Tutorial/ Sample Code</td>
<td>(Bhaskar et al. 2016), (Burns et al. 2012), (Zghidi et al. 2017)</td>
<td>Low</td>
<td>There is no tutorial / sample code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td>There is some sample code or tutorials for selected parts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>There are sample code and tutorials available for the API. That means that every complex usage of the API is explained with an example for a better understanding.</td>
</tr>
</tbody>
</table>

Table 1: API Evaluation Criteria

The API Design Process

The following section describes the applied API design process which is based on an iterative action research approach (Baskerville & Wood-Harper, 1998). It contains four steps: API Diagnosing, API Action
Lead User Involvement for Automotive API Design

Planning, API Action Taking and API Evaluation. For comprehensibility reasons, we illustrated each step by the representative example of the startGuidance() method inside the Navigation API, whose design was refined in the API design process. The method enables an application to change the currently set destination in the vehicles navigation system to given GPS coordinates and activate the guidance.

API Diagnosis

During the diagnosis stage, we identified the primary problems that are anchored in the organization’s major drive for introducing change. It is required that the researcher identify these problems in a complex organizational structure (Baskerville 1999). In the context of this research, this step required the analysis of the prevalent APIs. To gain deep insights about a specific API, we contacted all the relevant stakeholders. Being part of the project team granted the researchers’ access to different experts inside the organization. Overall, 21 app developers were involved in the diagnosis of the APIs. For the navigation API, we enlisted experts from the platform architect team, a software engineer from the navigation module and two app developers frequently using the navigation API. Using a code that implements the navigation functionality in an app, we assessed the API based on API evaluation criteria described in the previous section, and the assessment was via open interviews. The result of the evaluation showed that the navigation API’s simplicity was low. The experts reported a high internal complexity in the process of creating basic objects and triggering basic features like starting a guidance. In the case of the startGuidance() method the implementation of the interface required more than fifty lines of code which was commonly considered as inappropriate by the experts. The documentation of the API was rated medium to low. To start the guidance, the instantiation of multiple objects as well as methods calls were necessary. Neither a list of all the required objects and method calls nor the precise sequence of these actions was listed in the official documentation. The API’s usability was also considered as low. The experts reported inconsistent naming and typing patterns. An inspection of the raw code by an app developer confirmed this assessment. Finally,

![Image 1: Original Implementation of startGuidance() Method](image-url)
because there were no tutorial or sample code of the API implementation, on tutorial or sample code the API was rated low.

API Action Planning

Action planning is the collaborative step of the researcher and the practitioners to define next steps to relieve organizational pain or improve the existing situation. In the context of this project, the actual scope of the refined APIs needs to be clarified. Even though large parts of the available APIs were analyzed in the diagnosis phase, the scope of the actual refinements needed to be limited due to the finite capacity of the team. Action research implies the principle of change through action (Baskerville 1999). Following the paradigm, the team decided that the most change for app developers inside BMW could be achieved by covering the most commonly implemented interfaces. Therefore, 26 app development teams were asked for their prioritization. Based on these considerations, the team decided to work on the following APIs: Navigation, User Interface, Vehicle Data, Phone and Speech. Furthermore, a rough estimation of the required time for each APIs’ redesign was made. Finally, the order in which the refined APIs should be drafted, programmed and released was defined. The release of the startGuidance() method inside the Navigation API was determined for mid of October 2018. We announced the resulting timeline for releases of the new APIs to the app developer community.

API Action Taking

Based on the gathered insights in the API diagnosis, a draft for new APIs was created in the action taking phase. Besides the results of the evaluation criteria, we also took into consideration the underlying architecture. The design creation followed an iterative process in which the researchers exposed their design to the experts, collected feedback and refined the API until the design was sufficient for all involved parties. The API action taking involved the same 21 app developers as the initial API diagnosis. For instance inside the startGuidance() method, the destination could not exclusively be provided as GPS coordinates but also as address or as free text. One discussion was raised if the type of the destination should be contained in the method’s parameters startGuidance (destinationType: “GeoCoords”; destination: [48.178325, 11.556802]) or its name startGuidanceToGeoCoords (destination: [48.178325, 11.556802]). The initial proposal was a design that contained this information as property of the method. However, the app developer prompted a shift of the information into the method’s name. The experts argued that the text-completion feature of common code editors will propose all available options (startGuidanceToGeoCoords, startGuidanceToAdress, startGuidanceToFreeText) when the developer starts typing the methods name, while it will not propose all options for the method’s properties at this point. The availability of this text completion feature increases the usability of the API. Hence, the API design was refined. The creation of the complete navigation API design went through three iterations until the experts made no annotations. Furthermore, the implementation of the original interface required multiple lines of so-called boilerplate code (code that does not contain any relevant functionality for app developers but is required in each implementation of the interface). By abstracting this code into the SDK, the required lines of code for an implementation of the startGuidance() method could be reduced from more than fifty lines of code (Image 2) down to three (Image 4). The increased simplicity of the method reduces the efforts for app developers to understand the respective interface and decreases the likelihood of mistakes in the implementation. The design of the actual API in the code affected the evaluation criteria of simplicity and usability. However, also the lack of documentation needed to be fixed. For this reason, as soon as the actual design of the code
was finalized a comprehensive documentation was added inline. Further, the team created tutorials for each API which were provided in a web portal for app developers inside BMW. These instructions were based on an implementation of the refined API in a reference app which was also provided to the app developers. In this way the functional principle of the API could be comprehended by the app developers. Moreover, the functionality of the new APIs was validated from an app developer perspective inside the API team, even before it was officially released.

![Image 3: Refined Implementation of startGuidance() Method](image)

### API Evaluation

About three weeks before the release of the API; the team released a beta version of a new SDK, including the refined APIs. Additionally, a thread in an internal app developer forum was started, asking for feedback from the app developer community. Since the release was tagged as beta, the team was allowed to identify the further need for refinements and implement them in the API. Overall 12 app developers provided feedback via this channel. Based on the response of the app developer community, the team releases three beta versions until a stable version of the SDK was released on the announced date. For the introduction of the navigation API, three beta versions of the SDK were provided to the app developer community until its final version was released. The evaluation of the community increased the usability of the API on the level of methods and properties. App developers pointed out inconsistency in naming patterns and missing properties in specific methods.

### Results

The application of action research strives for the achievement of two goals. First, the researcher tries to introduce real change within the organization (Babioğlu and Ravn 1992). The first part of this section provides an overview of all refined APIs that were created through which change was established in the context of this project. Second, to expand the scientific body of knowledge (Baskerville 1999). The remaining part of this section describes the observed forms of lead user involvement in the API design process. These results serve as the base for the subsequent discussions on lead user involvement in the design of enabling technology.

### Created Artefacts

The initial goal of this research project was the analysis and refinement of poorly designed APIs that were exposed to developers of onboard apps inside the car’s head-unit. A diagnosis of all prevalent API modules embodied the base for our further proceeding. Considering the relevance of APIs to app developers, we decided to focus on the refinement of the five most used APIs. First, the navigation module exposes the functionality of the onboard navigation system to the apps. Further more, the vehicle data API provides information on the current car status as the current vehicle speed or fuel status. The UI and speech APIs provide interfaces that allow the customer to interact with the app via a graphical user interface or voice respectively. Finally, the Phone API provides access to the functionality of a smartphone that is connected to the car’s head-unit via Bluetooth. In all, the refined APIs contain 59 methods and 31 properties that were implemented by the app developers. The evaluation of the original (Orig.) as well as the refined (Ref.) API design reveals a clear improvement regarding all defined evaluation criteria through the API design process (Table 2).

<table>
<thead>
<tr>
<th>API</th>
<th>#Involved App Developers</th>
<th>#Refined Methods</th>
<th>#Refined Properties</th>
<th>Simplicity (Orig./Ref.)</th>
<th>Documentation (Orig./Ref.)</th>
<th>Usability (Orig./Ref.)</th>
<th>Sample Code (Orig./Ref.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation</td>
<td>4</td>
<td>21</td>
<td>6</td>
<td>Low/High</td>
<td>Medium/High</td>
<td>Low/High</td>
<td>Low/High</td>
</tr>
<tr>
<td>Vehicle Data</td>
<td>5</td>
<td>2</td>
<td>21</td>
<td>Medium/High</td>
<td>Medium/High</td>
<td>High/High</td>
<td>Low/High</td>
</tr>
</tbody>
</table>
The likelihood that the final solution would be accepted by a majority of the app developer community would increase if the quality of the final API design (Babüroglu and Ravn 1992) is improved. The implementation of the APIs in the BMW onboard platform SDK should be prioritized. The APIs were released in the platform SDK according to the timeline (Image 3). All currently developed apps implemented the refined APIs until February 2019. These apps will be to available to BMW customers shortly.

Theoretical Learnings

The deep insights we gained through the application of action research allowed us to identify the challenges and benefits of lead user involvement in each step of our applied API design process. Thereby, we classify our findings into psychological, managerial, methodological, cultural and political perspectives (Bano and Zowghi 2014).

By analyzing the implemented APIs per app, we identified the heaviest users of an API within the app developer community. However, even though all contacted app developers replied positively on our request, just a fraction of them was able to participate in our project. Especially employers of external suppliers were impeded by limited time capacity, since their assignment didn’t include such additional activities (managerial challenge). Further, the app project leaders were not motivated to shift capacity from their project towards the API design project. High pressure on the app development teams forced them to focus all their efforts on the app development itself (Psychological challenge). Finally, the evaluation of the original APIs itself, revealed a high complexity within the system (managerial challenge). However, the iterative approach as well as the involvement of the API provider module engineers enabled a comprehensive understanding of all involved parties. The conflation of the API consumer and API provider in common meetings simplified the communication (managerial benefit), facilitated knowledge sharing (cultural benefit) and created mutual comprehension of the challenges and difficulties of the respective other side (psychological benefit). This again motivated all participants to improve the status quo.

The API planning step comprised the prioritization of the refined APIs. Even though the most implemented APIs were refined first, this approach endangers the loss of engaged lead users, since their more specific API wasn’t part of the project any longer (managerial challenge). Further, the announcement of the API refinements raised expectations from management as well as the app developer community, which needed to be handled by the team (psychological challenge). On the other hand, the involvement of the lead users simplified and enhanced the prioritization process (methodological benefit). Thus, there were APIs that were implemented by a large number of apps. However, their status quo was more sufficient than the design of other APIs that were implemented by a slightly smaller number of apps. In this way, the involvement of lead users enabled a better understanding of user requirements. Further, the involvement of app developers enabled a better understanding of the required refinements measures and the support the establishment of realistic expectations towards the project within and outside the team (managerial benefit).

Next, the API action taking step considered the actual creation of refined API designs. The involvement of multiple lead users enabled the definition of a generic API design that satisfies not only specific but a broad spectrum of use-cases. However, the number of iterations needed for a commitment of all involved parties regarding the created design couldn’t be estimated in forward. Hence, the limited amount of time available embody a challenge for the team (managerial challenge). Further, it turned out that different app developers had conflicting requirements on specific API methods. Even though, these conflicts could be immediately resolved by the API design team, they embody a challenge (political challenge). On the other hand, these conflict resolutions increased the commitment on the refined solution and increased the motivation of all involved parties. All involved parties agreed that these kinds of discussions increase the quality of the final API design (methodological benefit). Further, the commitment of all parties increased the likelihood that the final solution would be accepted by a majority of the app developer community (psychological benefit).

### Table 2: Overview on Refined APIs

<table>
<thead>
<tr>
<th>UI</th>
<th>5</th>
<th>19</th>
<th>2</th>
<th>Low/High</th>
<th>Medium/High</th>
<th>Low/High</th>
<th>Low/High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>Medium/High</td>
<td>Medium/High</td>
<td>Low/High</td>
<td>Low/High</td>
</tr>
<tr>
<td>Speech</td>
<td>3</td>
<td>14</td>
<td>1</td>
<td>Low/High</td>
<td>Medium/High</td>
<td>High/High</td>
<td>Low/High</td>
</tr>
</tbody>
</table>
In the final API evaluation step, the beta versions of the APIs were published to the community. The involvement of further users of the API should increase the maturity of the final API release. However, this required a material amount of app developers to implement and evaluate the APIs. This increased the amount of required time for the overall process (managerial challenge). Though, the involvement of even more app developers further increased the likelihood for the identification of insufficiencies or flaws in the final design (methodological benefit). Further, the app developer community felt involved and fetched up for the upcoming changes. In this way the overall acceptance for the project could be increased (psychological benefit).

<table>
<thead>
<tr>
<th>Process Step</th>
<th>Challenges</th>
<th>Benefits</th>
</tr>
</thead>
</table>
| **API Diagnosis** | • Time constraints (managerial)  
• Lack of motivation (psychological)  
• System complexity (managerial) | • Simplified communication (managerial)  
• Facilitated knowledge sharing (cultural)  
• Increased motivation (psychological) |
| **API Action Planning** | • Time constraints (managerial)  
• Users and Managers expectations (psychological) | • Better understanding of requirements (methodological)  
• Development of realistic expectations (managerial) |
| **API Action Taking** | • Time constraints (managerial)  
• Conflicts (political) | • Increased quality of final design (methodological)  
• User acceptance (psychological) |
| **API Evaluation** | • Time constraints (managerial) | • Increased quality of final design (methodological)  
• User acceptance (psychological) |

**Table 3: Challenges and Benefits of Lead User Involvement in the API Design Process**

Our analysis reveal that especially managerial challenges affected the API design process. This approves Svahn et al. (2017) observations from another automotive case at Volvo: The management needs to conceive the need for a shift towards developer-centric software design. Otherwise, real change is hard to achieve in an incumbent context. However, while the Volvo case remains in the observation of these managerial phenomena, our study proposes user involvement as potential approach to address this challenge bottom-up. The interplay of multiple, heterogenous developers is able to achieve real change. Our results prove that this is not just true for matured app platforms (Eaton et al. 2015) but also in the context of a just emerging digital platform in an incumbent context.

**Summary and Outlook**

In this study we achieved a valid involvement of lead users in the evolvement of an incumbent service system embodied by automotive onboard APIs. The participatory approach for the API design enabled the creation of APIs that does not require domain-specific knowledge to implement (Pühler 2011; Schlachtbauer et al. 2012). The identified benefits prove the relevance of user involvement for refining existing technology in an incumbent context. However, BMW as organization is optimized towards the development of traditional products. In this setting, the customer is usually considered as the only user of the developed product. The usability of a feature that is not visible for him is not considered as dispensable. However, APIs are not directly used by the driver of the car, but the developer who is building complementary apps. The identified challenges in the shift towards app-developer centric APIs prove this fact. The elimination of these requires reconfiguration within the organization and its mindset (Svahn et al. 2015). Furthermore, we approve action research as valid method for gathering deep insights in a real organization, its technologies and its processes. The approach enabled a comprehensive understanding of the app developer's initial needs and their positive evaluation of the initiated change. However, these findings reveal a short-range character. The long-term effects on the actual app development activities remain in the dark. Therefore, an investigation on the effects of the new APIs on the app development appears as promising extension of this study. Does the shift to user-centric API design actually foster the development of apps? This question could be addressed by a quantitative analysis of the utilization of the created APIs as well as by further qualitative research.
REFERENCES


Lead User Involvement for Automotive API Design


