THE ROLE OF CONTEXT IN THE CREATION OF DIGITAL TECHNOLOGY FOR LOCAL MARKETS

Kari Koskinen

Abstract

The ongoing digitalization and increased connectivity across the world have made application development technologically feasible in almost any part of the globe. No matter where one is, most of the resources such as application programming interfaces (APIs), software development kits (SDKs) and required skills can be acquired via Internet. There is an increasing interest towards promoting software development and creation of app economies to serve local markets. What remains more unclear is how the context, where application development takes place, is relevant. This research aims to answer that question by studying application development in East Africa, concentrating to Kampala, Uganda. The data has been collected by interviewing entrepreneurs and software developers in the region and doing participant observation in the local tech hubs where many of the start-ups reside. Using structuration theory and the concept of habit formation in the user space the research finds that even though the technologies and skills needed for the application development are there, the context heavily conditions what type of applications are created and how, as well as the ways applications are being pushed for targeted users. The contextual challenges for the entrepreneurs are not only technological, but also display themselves in the form of existing social habits that the entrepreneurs aim to replace with their technology-driven solutions. The entrepreneurs find themselves between two different types of structures, those that come with the technological resources and the ones that exist in the context where their applications are built and used.

Keywords: Application development, IS diffusion, Context, Habit, East Africa.

1 Introduction

The layered, modular architecture of smartphones and other consumer devices has enabled third-party application developers to produce content and applications to these devices. This has led to the creation of millions of applications varying in complexity and purpose (Tiwana *et al.* 2010). The application developers themselves are relatively independent individuals and companies, which often have no other formal linkage to the platform owners or device manufacturers than the technologically mediated user agreements, technological resources and distribution channels (Maurer and Tiwana 2012). Many of the resources needed in application development have been digitalised and can be offered directly over the network, and as a result, software development kits (SDKs), application programming interfaces (APIs) and even material for skills development can be accessed everywhere and at any time (Yoo *et al.* 2012).

In this sense, it can be argued that application development has become disconnected from location, since many of the resources needed in the process and channels to distribute the applications are digital and available online. Digitalisation overall leads to de-contextualization, where items that get digitalised become ubiquitous and detached from their original location and form (Bailey *et al.* 2011). In a similar vein, the resources that developers use to build their applications can be easily transferred from one location to another, and provided one has access to basic hardware and the internet, application development can take place practically everywhere.

However, this research is based on the premise that despite the near ubiquitous existence of the technological resources needed in application development, the context where application development occurs and where an application is used also plays a vital role in the process. The notion of context has been much discussed in the literature in terms of technology implementation (Avgerou and Walsham 2001; Avgerou 2002; Heeks 2002; Rai *et al.* 2009), yet in relation to development of technology, the meanings and implications of context have not been studied to a similarly large extent. The purpose of this research is therefore to study the role of context in application development. The research question it aims to answer is *how does the local context displays itself in the applications and affect their development?*

The topic is approached from the perspective of tech start-ups operating in a particular geographical and social setting, namely East Africa, concentrating mainly on Kampala, Uganda. All of the start-ups studied for this research work with applications that are meant for local markets. Overall, the research uses the concept of habit to operationalize the more complex notion of context, which allows to better understand the challenges that the application developers face in their push to make their applications successful. It enables to see application development as a process, where the developers constantly interact and try to alter the existing habits in a way that allows them to attract as large number of users as possible.

This paper is organized so that it first takes a closer look on what kind of technological artefact an application is, and what is the role of technological resources in enabling the creation of applications. From there the paper moves on to examine the meaning of context in the application development. Then context will be operationalised by using the concept of habit in explaining the contextual factors that affect the application development, after which the study is placed on a broader framework by using structuration theory. Then the data collection and analysis will be discussed, following a conclusion chapter on the findings and ideas for possible avenues to follow in completing the research.

2 Application Development as a Technological Process

> *'Applications consist of suites of IT capabilities. They are developed to meet a set of specified user needs within a select set of communities. They can grow amazingly complex in terms of*

> *effort and scope, but despite this, they still can be viewed as applications, if governed by a set of specifications through which their design scope remains bounded. An application is a priori determined by choice of design context, user groups and functional goals'* (Hanseth and Lyytinen, 2010, 2).

As can be seen from the description above, applications cannot be seen as merely technological products, but their creation is heavily affected by the non-technological factors. However, the overall base of any application is technological due to its digital nature. Yoo et al. argue that pervasive digitization has led to a new type of product architecture, which they label layered modular architecture (Yoo *et al.* 2010). They see it as a mix of modular architecture and layered architecture, where the former is about products that are compounded of loosely coupled components through interfaces, and the latter refers to digital technologies being embedded into physical products and thus bringing in capabilities in the form of software. Yoo et al. see that digital technology consists of four different layers of architecture: devices, networks, services and contents. The application developers find themselves in the service layer which is on the very top of this stack, and the applications rely heavily on the layers below in their functioning. They build their applications by using the application programming interfaces (APIs) and software development kits (SDKs), which enable them to use the resources provided by different sensors and functions of the actual mobile device.

Application platforms' SDKs and APIs can be seen as prime examples of generativity. Tilson et al. define it by saying that generativity *"denotes the ability of any self-contained system to create, generate, or produce a new output, structure, or behaviour without any input from the originator of the system"* (Tilson et al., 2010, 750). SDKs and APIs provide software developers a certain set of resources and tools, out of which the developers then choose the ones which are most appropriate for their own purposes in terms of building an application. Furthermore, many of the applications themselves offer developers additional tools in the form of APIs that other applications can connect to and draw resources on.

The main implication of technology to context becomes visible in the form of these technological resources, since it is those that ultimately decide what can or cannot be done. This is not to say that the resources decide deterministically what kind of applications will be built, yet whatever will be done has to take place within the limits set by the technology. Ghazawneh and Henfridsson talk about these as boundary resources and define them as *"the software tools and regulations that serve as the interface for the arm's-length relationship between the platform owner and the application developer"* (Ghazawneh and Henfridsson 2013). In other words, boundary resources function as toolkits that enable the transfer of application creation capabilities from the platform owners to the third-party developers (von Hippel and Katz 2002).

In sum, the technology itself functions as a constraint by simply not allowing certain things to be done and thus placing the ultimate limits to the development of an application. These constraints can be temporal however, as technological advancements and updates present new possibilities of usage (Corral et al., 2014). Furthermore, the constraints are also dependent on the developer's ability to use the technological resources. In some cases, a certain technology does not simply fit a particular context.

In sum, technology and technological resources define the limits for application development but not directly what will be done and how. This is where contextual factors become relevant and come to show that development of an application is not only a technological process but is also affected by the location where it takes place and where it will be used. The question remains though what is meant with context and how it can be operationalised in research.

3 Role of Context in Application Development

The concept of context is widely used but seems more difficult to define. In the literature some have approached the topic from a rather technological perspective by understanding context in the form of context-aware applications (Chen and Kotz 2000; Rahimian and Habibi 2008). Others have directed their attention to digitized context, Grudin for example argues that once context is transformed into a digital form, it also becomes altered and becomes a context in its own, leaving behind the parts of context that were not included into the digitalization process (Grudin 2001). This type of technological context can then be removed from its original framing, and once digitalized, appear anywhere on the planet at any point of time.

Other have linked context to user experience by looking how a device fits a particular context. For example, Ryan and Gonsalves differentiate between objective and subjective factors, where the former can be measured by testing how many mistakes users make or how quick they are in performing the given tasks. The subjective part consists of user perceptions on the device, such as how easy the users find the device to use (Ryan and Gonsalves 2005). Seen from a users' perspective, context links to the users' ability to perform certain functions by using the technology.

Context seen from a technological perspective ignores many relevant social and other factors, which provide a broader understanding of the term. Dey et al. argue that context is a crucial element in application development and define it as *'any information that can be used to characterize the situation of entities (i.e. whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and applications themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects'* (Dey et al., 2001, 106). From this definition it becomes clear that context is a concept that includes multiple agents, both human and material, and becomes visible in the interactions that occur between the different locations, objects, groups and individuals.

Others have noted that context is not necessarily stable but keeps on changing, also when transformed from one form to another. Dilley (2002) talks about the social life of contexts, and by that he refers to the dynamic nature of context as something that is being continuously interpreted and constructed. Context is an action of creating connections and disconnections between the relevant actors, objects and environments, and it is done by the person who interprets the relevant setting in a particular case. In this sense context displays itself as fluid and relational, subject to interpretations done by the relevant actors. Drawing on Dilley, Hayes and Westrup (2012) argue that context could be understood by using concepts of distal and proximal accounts (Cooper and Law 1995). Distal representations see entities as stable and discrete with clear boundaries. Proximal representations include the notion of time by concentrating on the processes that lead to certain distal representations. Context cannot be only narrowed down to a particular type of setting or state where events take place, instead it has to be seen as dynamic and relational (Huen 2009).

The problem with all of the above mentioned different definitions of context is that they are relatively difficult to operationalize. This is why the concept of habit is needed to better grasp the complexity of context and turn it into a more observable form.

4 Habit as an Indicator of Context

The concept of habit has received attention in IS literature in explaining especially the continuing usage of a particular IT artefact. Whereas intention plays an important role in understanding adoption of technology, habit can better explain the usage that continues after the initial adoption of an artefact. Habits have been largely missing also from models such as Technology Adoption Model (TAM) (Venkatesh and Bala 2008). However, during the past decade models have been created that aim also to explain the continued use of IT and the factors that contribute to this usage (De Guinea and Markus

2009; Ligun and Hui 2013). Similar to Schumpeter's views on the difference between invention and innovation (Schumpeter 1934), for an application to be a success, people have to have adopted it and also use it on a continued basis, in which case the usage can be seen as habitual.

Habits are often performed semi-automatic and goal-driven actions, that individuals repeat on a frequent basis. Limayem et al. define habit as *'the extent to which people tend to perform behaviours automatically because of learning'* (Limayem et al. 2007, 705). Overall, habits can be seen as products of particular contexts and as such automated responses to different situations. Verplanken et al. understand habits as *'learned sequences of acts that become automatic responses to specific situations, which may be functional in obtaining certain goals or end states'* (Verplanken et al. 1997, 540). This definition of habit further highlights their goal-oriented character. Overall, for a habit to form, it is believed that certain amount of repetition or practice is needed, as well as a stable context that guarantees that the same habit can be used over and over again to reach a particular goal.

The concept of habit also captures many of the characteristics that are discussed in relation to context. Following Ryan's and Gonsalves' thinking on context, habit can be seen as having both objective and subjective side. Objective side forms the general settings in which the subjective side is based on: the individuals need to reach a certain goal, and they choose a route of action to reach that goal that fits the context they find themselves in. In most cases, the ideas behind many mobile and other applications are based on the notion of subjective side of habits, i.e. they offer a different route of action to reach a goal that has been traditionally done in some other way. In order to make the applications succeed, the developers must not only build the application but also be able to restructure their intended users' current habits so that it includes the usage of the application. Dey's definition of context can be understood as a description of the process that has led to the existence of prevailing habits. The reasons why certain actions are performed in a particular way has been the result of the local context that has decided on its part which habits have become feasible modes of action. The offered technological form of performing a habit, i.e. using the application, is based on similar settings than the existing habits, but as Grudin notes when it is dressed into a technological form, the habit itself becomes altered. The new application might offer the same end result as the existing habit, but provides an alternative route that the intended users may either accept by using the application or then rejecting it.

As mentioned, the success of an application depends on its capability to satisfy a certain need or want that exists among the targeted users. Habits can be linked back to these needs and wants through habit formation. As seen in figure 1, needs and wants lead to goals, which on their part create an intention to reach that goal. Intensions on their end turn into behaviours. If these behaviours keep on being repeated long enough in a stable context, the argument goes that these behaviours will turn into habits (Ligun and Hui 2013).

Figure 1. The connection between needs and wants and habits

Within the confines of this study, habit is therefore used to shed light to the contextual factors that explain some of the challenges that the applications and their developers face in their work. The logic behind the applications is based on the idea of renewing or altering a certain existing habit of targeted users. Limayem and Hart talk of facilitating conditions (Limayem and Hirt 2003), which consist of different enabling factors that facilitate the birth of a habit and which have to be fulfilled in order for the new habit or the alternate form of it to develop. Following this line of thought, the challenges for the application developers exist in two different levels; firstly, they have to build their applications in a way that these applications can be reached and used by the users. Secondly, the developers have to be able to convince the users to change their current habits and to adopt the new or altered, technologically mediated way of performing the habit to reach the goals they have set for that particular habit. The

ability or inability to attract users can be caused by different reasons, but as a whole these reasons can be categorised to falling either to the category of technological or then contextual factors, which is also the categorisation that will be used in presenting the results. These technological and social factors form a set of structures that the developers try to accommodate so that the technological structure matches the existing social one, making it easier for the targeted users to adopt the application and use it.

5 Context and Technology as Structures

As discussed above, habits and their formation are linked to the overall context where they are situated. When habits are being performed, they reproduce on their part the structures they stem from. Overall habits are seen as relatively stable in terms of time (Jones and Karsten 2008), indicating that also the structures behind them are less amenable to change.

As structuration theory states, structure and agency should be viewed as a mutually interacting duality. The structure is being created and reproduced by agents via their actions, but structure also guides the actions of the agents for example through the norms and customs that exist within a certain context. Structure enables and constraints actions but is also being constantly recreated by agents through the actions they take (Rose and Scheepers 2001).

Structuration theory can be used to highlight further the interconnectedness that exists between technological resources and contextual factors. It is possible to view technological resources and local context as two sets of structures. When application developers aim to introduce new technologically mediated habits into the local context, they can be seen as trying to alter the existing contextual structure so that it would allow the technologies and applications to function in the context. In practice this happens if the applications are able to attract users that use the application and by doing so, accept the new habit that is offered by the developers via their application, which then becomes part of the contextual structure. This process is shown in figure 2.

What is noteworthy is that technology also functions as a certain type of structure, as it heavily conditions on what the developers can or cannot do but also enables the developers to impact which technologies prevail over others. However, the overall ability of the technological structure to be changed by the developers is limited, as the developers can only choose from the options that a technology or technologies provide, but not to modify or alter the technologies profoundly. One could argue that this also applies to context as a structure, but the main difference is that regarding the technology, the ultimate limits on what can be done are already decided by the creators of the technological resources that the application developers use, where as in the case of the social structure these limits are not entirely known.

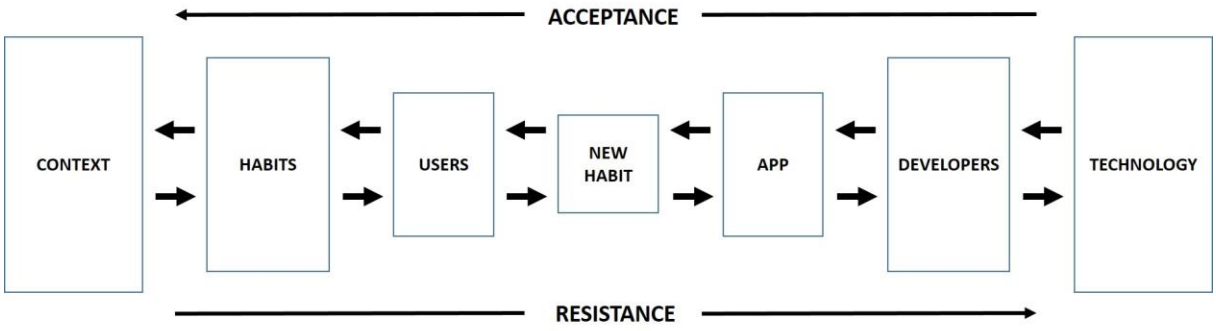

Figure 2. The connection between contextual and technological structure

These two types of structures provide a basis for discussing the role of context in application development. The developers aim to introduce a more technological form of performing a habit that stems

from the so called technological structure. The existing habit on the other hand is based on the local context and structure. If the users accept the new form of performing the habit, it ultimately also has implications to the existing contextual structure. In case the users reject the new technologically mediated way of performing the habit, the developers then have to go back and modify the application according to the possibilities offered by the technology. It is important to notice that technological resources and contextual factors are not exclusive as such, but can in the process merge to some extent if the users adopt the technology and that way recreate the context they reside in.

Within the confines of this paper, structuration theory functions as a link between technology and the context that exists in a particular location. The developers are seen as agents that are actively engaged in trying to shape both sets of structures, i.e. the technological resources and existing contextual factors, in a manner that allows them to attract the targeted users to use their application.

6 Data Collection and Analysis of Data

The data consists mainly on 35 longer interviews with application developers and other relevant actors in East Africa, concentrating mainly to the region of Kampala, Uganda. Additional data has been gathered by participating to pitching events for technology start-ups as well as workshops, and also by spending time on two different tech hubs in Kampala and performing participant observation. Furthermore, several informal discussions have been conducted with people belonging to intended user groups in order to find out their opinions about the applications.

The interviews were in most cases recorded and when not, notes have been taken. Notes were also taken from the participant observation taking place in the hubs as well as from the informal talks with the intended users and relevant stakeholders. The data was analyzed by using thematic analysis. Basically the in the analysis references were collected that were linked either to the adaptability of technology for the intended purpose, or then to the existing habits that existed in the user space and that the developers intended to tap into and change with their application. In addition, any kind of comments and references that were made regarding the overall context and technological resources were also categorized and analyzed, as they were seen as indicators of the technological and contextual structures that guided the developers' intentions in developing the apps and trying to make them successful. In the following the results from the analysis are presented.

7 Results

7.1 Technological Resources and Constraints

As noted above, technology, both in terms of providing a certain set of resources but also as setting requirements in terms of skills, establishes the first area to analyse the role of context in application development. Technological resources provide the building blocks for an application, and by doing so place the ultimate limits on what can be done. Their usage also requires certain skills, and the ability of a developer or a user to build and use the application depends on the skills they have or are able to learn. In other words, in order for the technology to function properly certain conditions have to be met, which may or may not be present in the context where it is to be built and used.

The data overall showed that the developers themselves did not seem to face difficulties in implementing the technologies or building the applications, nor had too many problems in shaping the available technological resources for the initial purposes. However, what became more relevant for the developers throughout the building process was taking the targeted users into account when choosing the correct technologies and ways of building the applications. As an example, some of the games created for the local context had to be built in a way that would run smoothly on smartphones with less processing power. Another more common example was that sometimes the applications had to work also on the

basic phones, which in practice resulted in reducing some of the functionalities of the application in question. This also meant that some of the processes could not then be automatized and had to be done manually, which demanded additional resources that the start-ups were often already short of. Furthermore, when this type of technological obstacles could be overcome, i.e. the technology could be used to reduce manual work, there were still often social factors that had to be solved. An example of reducing manual work were mobile payments integrations to the application, where for example for one start-up integrating the application to an existing mobile payment system turned out to be technologically demanding as it had not been done before. When it became possible to integrate the payments to their system, one of their intended target group rejected receiving payments in mobile money, preferring cash instead.

This also linked to another issue, namely the targeted users' understanding and usage of technology. Since having data on mobile subscriber plan was relatively expensive, many of the users who had this preferred to use their data on already known applications such as Facebook or WhatsApp. Overall the applications should not consume too much data also because the connections were sometimes slow. Another user related aspect was that the technological form of presenting things was not always understood by the users. An example of this were maps, where for some local users the way maps present reality did not make much sense.

Overall, many of the application developers had to make a sort of compromise between the technologies and the size of intended user groups. The trade-off was that if the application was built using more advanced technologies and devices, it easily led to a reduction in the size of the intended user groups. The alternative for not using these advanced technologies and settling for less than optional technology meant more manual work. In sum, these technological factors set the underlying conditions for any type of alteration to occur in the existing habits: if the technology for any reason did not reach or make sense for the intended users, there was very little chance for the application to gain traction among the users and altering the existing habits. Getting the technological aspects right and making them sufficiently match the existing realities and user preferences was a first step in creating a successful application.

7.2 The Social Implications of Context to Application Development

The ideas for the applications were drawn from the local context. The start-ups and the applications basically targeted a habit that existed within the location and built the application around it. Thus, many of the applications tried to streamline some habit by building an application that made performing the habit more efficient or somehow better.

However, in some cases the applications did seem to entirely match what was wanted by the intended user group. The solution in some cases was then to redefine the targeted user group, although that ran the risk of reducing the intended users to a too small of a group. An example of this was an app that provided safer motorcycle taxis via an app, where a member belonging to an original targeted user group stated the following:

> *"For me safety when taking a boda (a motorcycle taxi) is not such a big thing... I mean, I know they are unsafe and that there are many accidents, but that is the way they are, you just take them from the street… I don't really see myself using that app because of safety, you might even end up paying more..."* (A local person on whether he would use an application that allows the user to have a motorcycle driver that is safer and offers helmets for the passengers).

However, a person belonging to another user group did see safety as an issue:

> *"I try to take only those bodas since they are safer, and you get your own helmet... the traffic here is quite dangerous, lot of accidents and the other boda drivers drive like crazy"* (A foreigner visiting Kampala on why she uses the safer motorcycle taxi application).

The problem was that the market was not quite big enough for the company to be successful only in terms of relying only to customers belonging to the redefined user group. In a similar vein, in some cases the existing non-technological habits resisted the introduction of new habits, and the intended users preferred to continue using them over the suggested more technological approaches of performing the habit.

> *"We target students, cause they might have the money for this yet may not have clear how to do the laundry […] I mean if you are older, you have someone to do these thing […] Also we need to make sure they don't use the service just once to get a number of a washer and then bypass our system from there on..."* (A developer working on an app that creates a platform for people to get in contact with persons who do laundry in their area)

As a result, having built the application, many of the start-ups spent considerable time in persuading the intended users to adopt the application and getting rid of the old habit. In other words, since the applications were linked to some habit that people performed, a lot of the work of the start-ups became non-technical and went for persuading the intended users to renew their existing habits.

> *"I ain't doing almost any of that technological development anymore, all my time goes to the non-technical stuff, like getting the people to use the service, calling them and asking how was the experience, and then telling the person doing the service if the feedback was not good […] this is about a lot more than just the technological part"* (A developer on his changed role)

The result was that in many cases, the tech start-ups ended up as being not very technological, resembling more of traditional businesses where the application maybe only had a minor or no role at all. What had started as an application business, ended up being as a business where the developers and the start-ups had to increasingly find manual ways of doing this in order to meet the users' wants and needs and their preferred ways of doing things.

8 Conclusion

The common nominator between the findings presented above is their link to habits, either in the form of being the current habits of intended user groups, or then habits that the application developers tried to introduce to the intended users via the applications. Overall, defining context via habits that exist in a particular location and setting serves to connect context to application development and also allows the researcher to transform the difficult concept of context to a more usable form.

In terms of technological resources, the developers seemed to be able to build the applications the way they wanted by using these resources, as from a skills perspective there was no problem using resources produced elsewhere and then applying them to another and context. However, the issue became more complex when introducing the applications to the users. If the users did not have the required tools or skills to use the applications, it was enough to prevent them from adopting these new technology-driven habits. In many cases the developers tried overcome these obstacles by offering ways to use their service that did not include the usage of the application, but this also led the application to lose importance in the overall business and causing more manual work.

Therefore, to understand the chances of success for applications in a particular context, the process must not be seen only as a technological one but as heavily affected by the local context and the social structures that are in place. The applications have to be able to replace existing habits and become the preferred way or habit of performing certain functions, and building an application is only one part of the process. Habits stem from the local context, and the ideas for the applications and the challenges the developers face can be traced back to local habits. By creating applications that are closely linked and fit well with the local habits and their performance, the likelihood of success increase in the local

market. However, by building the application around a very particular habit may have consequences when scaling up to new markets as habits are likely to be at least somewhat different when moving from one context to another.

References

Avgerou, C. (2002) *Information Systems and Global Diversity*, Oxford University Press.

Avgerou, C., Walsham, G. (2001) *Information Technology in Context: Studies from the Perspective of Developing Countries*, Ashgate Publishing Company: Brookfield, VT, USA.

Bailey, D.E., Leonardi, P.M., Barley, S.R. (2011) 'The Lure of the Virtual', *Organization Science*, 23(5), 1485–1504.

Chen, G., Kotz, D. (2000) *A Survey of Context-Aware Mobile Computing Research*, Vol. 1. No. 2.1., Technical Report TR2000-381, Dartmouth College, Hanover, NH, USA.

Cooper, R., Law, J. (1995) 'Organization: Distal and proximal views', in Bacharach, S.B., Gagliardi, P. and Mundell, B., eds., *Research in the Sociology of Organizations: Studies of Organizations in the European Tradition*, 275–301.

Dey, A.K., Abowd, G.D., Salber, D. (2001) 'A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications', *Human-Computer Interaction*, 16(2), 97–166.

Dilley, R.M. (2002) 'The problem of context in social and cultural anthropology', *Language & Communication*, 22(4), 437–456.

Ghazawneh, A., Henfridsson, O. (2013) 'Balancing platform control and external contribution in third-party development: the boundary resources model', *Information Systems Journal*, 23(2), 173–192.

Grudin, J. (2001) 'Desituating Action: Digital Representation of Context', *Hum.-Comput. Interact.*, 16(2), 269–286.

De Guinea, A.O., Markus, M.L. (2009) 'Why Break the Habit of a Lifetime? Rethinking the Roles of Intention, Habit, and Emotion in Continuing Information Technology Use', *MIS Q.*, 33(3), 433–444.

Hanseth, O., Lyytinen, K. (2010) 'Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet', *Journal of Information Technology*, 25(1), 1–19.

Hayes, N., Westrup, C. (2012) 'Context and the processes of ICT for development', *Information and Organization*, 22(1), 23–36.

Heeks, R. (2002) 'Information Systems and Developing Countries: Failure, Success, and Local Improvisations', *The Information Society*, 18(2), 101–112.

Von Hippel, E., Katz, R. (2002) 'Shifting Innovation to Users via Toolkits', *Management Science*, 48(7), 821–833.

Huen, C.W. (2009) 'What is context? An ethnophilosophical account', *Anthropological Theory*, 9(2), 149–169.

Jones, M.R., Karsten, H. (2008) 'Giddens's Structuration Theory and Information Systems Research', *MIS Quarterly*, 32(1), 127–157.

Ligun, Z., Hui, L. (2013) 'How do the Contexts impact the Habit Formation of Information System Users?', *International Conference on Applied Social Science Research (ICASSR 2013)*.

Limayem, M., Hirt, S.G. (2003) 'Force of Habit and Information Systems Usage: Theory and Initial Validation', *Journal of the Association for Information Systems*, 4(1), available: http://aisel.aisnet.org/jais/vol4/iss1/3.

Maurer, C., Tiwana, A. (2012) 'Control in App Platforms: The Integration-Differentiation Paradox', *ICIS 2012 Proceedings*, available: http://aisel.aisnet.org/icis2012/proceedings/ResearchInProgress/5.

Rahimian, V., Habibi, J. (2008) 'Performance evaluation of mobile software systems: Challenges for a software engineer', in *5th International Conference on Electrical Engineering, Computing Science and Automatic Control, 2008. CCE 2008*, Presented at the 5th International Conference on Electrical Engineering, Computing Science and Automatic Control, 2008. CCE 2008, 346–351.

Rai, A., Maruping, L.M., Venkatesh, V. (2009) 'Offshore information systems project success: The role of social embeddedness and cultural characteristics', *MIS Quarterly*, 33(3), 617–641.

Rose, J., Scheepers, R. (2001) 'Structuration Theory and Information System Development - Frameworks for Practice', *ECIS 2001 Proceedings*, available: http://aisel.aisnet.org/ecis2001/80.

Ryan, C., Gonsalves, A. (2005) 'The Effect of Context and Application Type on Mobile Usability: An Empirical Study', in *Proceedings of the Twenty-Eighth Australasian Conference on Computer Science - Volume 38*, ACSC '05, Australian Computer Society, Inc.: Darlinghurst, Australia, Australia, 115–124, available: http://dl.acm.org/citation.cfm?id=1082161.1082174 [accessed 11 Nov 2015].

Schumpeter, J.A. (1934) *The Theory of Economic Development: An Inquiry Into Profits, Capital, Credit, Interest, and the Business Cycle*, Transaction Publishers.

Tilson, D., Lyytinen, K., Sørensen, C. (2010) 'Research Commentary—Digital Infrastructures: The Missing IS Research Agenda', *Information Systems Research*, 21(4), 748–759.

Tiwana, A., Konsynski, B., Bush, A.A. (2010) 'Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics', *Information Systems Research*, 21(4), 675–687.

Venkatesh, V., Bala, H. (2008) 'Technology Acceptance Model 3 and a Research Agenda on Interventions', *Decision Sciences*, 39(2), 273–315.

Yoo, Y., Henfridsson, O., Lyytinen, K. (2010) 'Research Commentary—The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research', *Information Systems Research*, 21(4), 724–735.

Yoo, Y., Jr., R.J.B., Lyytinen, K., Majchrzak, A. (2012) 'Organizing for Innovation in the Digitized World', *Organization Science*, 23(5), 1398–1408.