LEVERAGING FROM CUSTOMIZATION TO INFORM GENERIC SOFTWARE SYSTEMS DEVELOPMENT: DHIS TRACKER AND ITS INTRODUCTION IN HEALTHCARE

1. INTRODUCTION

The World Health Organization indicates that the lack of reliable data on several health related issues is what will hinder most developing countries not to meet the Millennium Development Goals. YEAGER (2012), for instance, argues that the lack of post-partum haemorrhage and pre-eclampsia/eclampsia data, together with the lack of skilled personnel hinders the accurate identification of gaps in service coverage. To this situation, it is also coupled the inability of health information systems to effectively provide reliable data.

To help countries with data problems, organization such as Health Information System Program (HISP), Regenstrief Institute and Partners In Health have developed Generic Software Systems (GSS). For instance, through the District Health Information Software (DHIS) developed by HISP it is possible to record and manage data of health programmes and information of patients. The DHIS framework offers also other potential advantages to clinicians including the monitoring and tracking of patients enrolled in health programmes (e.g., HIV/AIDS, Tuberculosis and Maternal Health).

However, it is a fact that customers of these generic products have needs that are quite heterogeneous(Franke & Hippel, 2003). It is also true that software vendors try to address these needs by designing systems around basic organizational functionalities, usually using 'generic' principles. Built to meet the demands of a mass of users, geographically spread across countries and living in completely different contexts from each other, the benefits of most GSS are various. These benefits are, generally associated with access to accumulated knowledge about organizational practices and the community of stakeholders sharing these practices(Michaud, 1999; Randell, 2003), and specifically, to the freedom in accessing the tools source code, and making changes to it.

In order to benefit from them, organizations need almost always, to undertake to the process of customization(Michaud, 1999). This process, apart from organizational and management factors(Sahay & Walsham, 2006), is influenced by GSS customizability. Swaminathan (2001) asserts that customizability of Information Technology (IT) systems depends on three factors: modularity, standardization, and predictability of demands and operations. While modularity is a principle that ensures that 'complex' and 'larger' software systems are divided into simpler and more manageable modules, thus making them more adaptable(Sullivan, Griswold, Cai, & Hallen, 2001); standardization allows multiple customizations to be performed on the same line without major changes, and predictability refers to the degree to which software developers can predict what the software will do(Bell, 1966).

Few software systems have, however, been released at their first glance with the above properties. Because of that, myriads of disruptions are observed during their customization. Likewise, the diverse amount of requirements associated with the fact that customization responsibility fall on use organizations, may lead to distinct versions of customized products(Iivari, 2010; Pollock, Williams, & D'Adderio, 2007). Their harmonization can be challenging to software developers, specifically when they have to collect the requirements expressed by the features added in the use nodes and include them into the generic tool. Pollock et al. (2007), for instance, state that too much diversity influences the ability to make software products generic.

Addressing most of these disruptions usually require global - local negotiations between customization players(Pozzebon & Pinsonneault, 2005). These players involve not only internal people (from different departments and hierarchical levels), but also a network of external individuals such as software developers, customizers and champion users(Pozzebon & Pinsonneault, 2005). The different interveners meet, discuss and share customization experiences and lessons. Therefore, it is during these negotiations that GSS gain properties that allow them to become mobile. These arguments point out to a discourse relating customizability as dependent on interconnections between domains, sites and organization, usually referred to as boundary spanning. Drawing from Levina and Vaast (2005) work, the paper intend to (a) build an understanding of the kind of boundaries spanned in making GSS customizable; and (b) discuss weather organizations should encourage or restrict the growth of practice during GSS customization.

To answer these questions, the paper builds on material collected during the customization of DHIS Tracker framework for use in the Malawian and Bissau Guinean healthcare information systems. Data was collected using artefact examination and participant observations, as well as document review and reflexive discussions. Findings of the study illustrate that adoption of DHIS Tracker by different healthcare service providers has resulted in instances distinct from the standard configuration of the framework, i.e., customization have resulted in fragmented versions of DHIS Tracker. In turn, this fragmentation posed challenges to both developers in harmonizing and satisfying user needs, and the end-users in their continuation of benefiting from the improvements being added to new framework versions.

The paper proceeds as follows: after introducing the main discussion that will be undertaken in this paper, the next section presents the theoretical framework. The third section presents the research approach. Within this section the contexts and the data collection and analysis techniques are described. The fourth section explores empirically the study research questions contrasting them to the research perspectives. This is followed by the analysis and discussion section where the contributions are presented. The conclusion of the study is presented in the later section.

2. THEORETICAL FRAME OF REFERENCE

GSS offer the possibility of adapting systems' functionality to users changing needs and requirements over time and thus stay current, applicable, and useful longer(Dourish, 1995). These applications predict that some maintenance tasks are offloaded to 'local developers'. The offloading of tasks is based on the principle that the mechanisms for resolving possible conflicts in the needs are in place and they enable the software application to meet undocumented users' requirements, thereby increasing the longevity of software system. To effectively perform software customization 'local developers' take advantage of the experienced actors (with different background) including 'marginal' maintainers and developers. This process involves also accessing a large pool of domain and professional knowledge that will free organizations of dramatic changes(Schmidt, Gokhale, & Natarajan, 2004).

However, the execution and sustainability of software customization projects has been a major concern in developing countries. According to Claeson and Waldman (2000) sustainability in healthcare affects several issues including design, management and execution of projects. Moreover, since GSS allow software functionalities to be changed, added, and deleted as required (Dourish & Edwards, 2000; Fischer & Scharff, 2000), sustainability includes also the ability of 'local developers' to change the software to fit local business processes. Fischer and Scharff (2000), for instance, asserts that customizable software system "allows users to invest

the world with their meaning, to enrich the environment with the fruits of their vision and to use them for the accomplishment of a purpose they have chosen"(Fischer & Scharff, 2000, pg. 398).

Besides bringing comfort, reinforcing a sense of individuality(Fischer & Scharff, 2000) and empowering 'local developers'(Michaud, 1999), the 'fruits of users vision' can lead GSS to completely separate paths. Several factors may influence the tool during its journey that might not allow the resulting tool to be easily integrated with its original version or the evolutions. For instance, 'local developers' might change the source code without following the guidelines described by the global team, global developers might decide to remove features that are already institutionalized in some local contexts, and so on. Understanding or easing these problems requires construction of knowledge about the software. The importance of collaboration between developers and users of the tools cannot be underestimated. This cooperation ensures knowledge to be shared and relevant information to be properly considered and acted upon by the tools' stakeholders, and is raised by considering customization as work spanning boundaries.

Boundary spanning is considered by Information System (IS) literature as a process of connecting two or more sides. Since it involves spanning professional and organizational settings, Levina and Vaast (2005) assert that the process benefits organizations. And, its study is suggested by boundary theorists as a way forward toward building an understanding of the nature of actions occurring at the boundaries. Boundaries, in this context, are considered as instruments for both splitting and connecting sides. Levina and Vaast (2005), for instance, while studying the issues surrounding the adoption of IT artefacts in organizations suggested paying special attention to boundary. To do that they posed four questions, namely which 'boundaries should be spanned?'; 'should boundary spanners be nominated or emerge from practice?'; 'how should boundary spanners be nominated?' and 'should organizations encourage or restrict the growth of practices?'

To answer these and other questions, boundary spanning literature relies on boundary-spanners and boundary-objects. While boundary-spanners are the individuals or groups of people that establish and maintain inter-organizational relations(Stock, 2006), relating the organization with elements outside it"(Lindgren, Andersson, & Henfridsson, 2008); boundary-objects allow different groups to work together(Star, 2010), thus articulating meanings and addressing multiple boundary perspectives(Akkerman & Bakker, 2011).

However, the practical assessment of GSS boundary spanning is whether the tool meets end-user needs and demands, and whether it is sustainable. These properties are attained when GSS are customizable, which as described in the introduction is dependent on modularity and standardization and predictability of demands and operations. In line with that, Daniels, Edwards, Engeström, Gallagher, and Ludvigsen (2010) stress that customizability can be attained as a result of general development, i.e., result from a software engineering work leading to versions of the IT artefacts; and as result of incremental changes made to it during customization work. The last is possible due to involvement of people from different backgrounds, institutions or organizations that contribute to the production of a share IT artefact.

Therefore the importance of understanding boundary issues through healthcare projects is driven by the fact of healthcare being a global subject, involving several bodies and organizations. And also because the majority of GSS tools become mature thanks to several lessons learned in each of its adoption sites, contexts and countries. This cross-country, cross-organization and cross-customization perspective might help to answer some of the above questions posed by Levina and Vaast (2005).

3. RESEARCH DESIGN

This research was performed under the umbrella of an action research network known as HISP and conducted from an interpretive perspective(Walsham, 1995). The study is empirically developed within the GSS context through DHIS Tracker and its adoption for the management of healthcare services in Malawi and Guinea Bissau.

The data used in this paper was collected from interactions with customizers, developers and end-users. The techniques adopted rely on a qualitative research tradition(Iivari, 2010) and has used software artefact examination and participant observations(Mason, 2002) as the primary data source. This process was done firstly in Malawi from 2010 to 2011, and subsequently in Guinea Bissau between 2011 and 2012. During this period the author was an 'involved researcher'(Walsham, 2006) acting as a customizer and performing activities such as requirement analysis, adaptation of the tools to meet those requirements and end-user training. Additionally, the author positioned himself as a resource, for both local and global teams, and collected local requirements and forwarded them to the developers. Through these activities it was possible to gain first-hand knowledge about several issues including system features and the customization disrupts. The author has also had the opportunity to inspect different versions of DHIS Tracker, and participated in the discussions around DHIS Tracker as well as test beta (unreleased) versions of the framework.

Moreover, document reviews and reflexive discussions(Jacobs-Huey, 2002) provided a secondary data source. Several documents including DHIS user and implementer manuals were analysed. In Malawi, for instance, notes taken by researchers that had been interacting with healthcare users and managers of mother and child health programmes were accessed. Through this, it was possible to draw the mother and child process model diagrams (Figure 1 and 2) and translate them later into DHIS Tracker. While performing customization work and observing the practices, the author interacted with end-users and local customizers, and observed the evolution of DHIS Tracker. The later was done by maintaining continuous contact with researchers working in other settings and with software developers. For example, several thoughts have been exchanged with customizers including those involved with DHIS Tracker in Malawi and Tanzania. Additionally, the author participated in several reflective discussions at local contexts, workshops at regional and at HISP global level. Through these workshops and discussions it was possible to reflect on DHIS Tracker features, and on end-users requirements and challenges.

To analyse the findings an interpretive cross-case approach was adopted. The approach was supported by Miles and Huberman (1994) data analysis framework steps of data reduction, data display and conclusion and verification. Through these steps, intra-case and inter-case analysis was adopted as data reduction techniques. By applying these techniques the data sources were examined and relevant information extracted. The data was visited several times for cross-checking with emerging conclusions, a process called interim analysis (Curro, Craig, Vena, & Thompson, 1995) which is characterized by cyclical practice of collecting and analysing the data. For instance, since the author participated in the customization of DHIS Tracker in the local contexts, he had the opportunity to participate in all the discussions related to the disturbances or challenges of the tools and also recalled and analysed them later with software developers.

4. EMPIRICAL INSIGHTS

HISP is a global south-south-north collaborative network(Braa, Monteiro, Sahay, Staring, & Titlestad, 2007) coordinated by the Global Infrastructures Research Group at Department of

Informatics of the University of Oslo. The group aims at supporting the improvement of health care systems in (the) developing countries by increasing the capacity of health care workers to make decisions. As a part of its activities HISP developed the DHIS software application framework. Over 15 countries, in Africa and Asia, use DHIS to capture, collate and analyse primary health care and hospital related information. The software is implemented at the national level in many countries including India, Vietnam, Kenya, Sierra Leone, Tanzania and Zanzibar. To ensure this flexibility, DHIS is provided as a generic tool with an open meta-data model and an adaptable user interface that allows the users to design the contents of their specific information system without the need for programming. The software is composed of four main modules, namely: *OpenHealthMapper* for mapping or spatial data display and management, *Routine Data* for the management of routine data, *Tracker* for community data management, and *mobiHealth* for patient-based data management using mobile devices. These four modules are designed to provide support to health workers and managers at all administrative levels through a balance between flexibility and standardization, and with a strong emphasis on using information for local action.

The paper investigated the DHIS Tracker module. The module was first developed in India as a context-specific module. After a while the request for similar features in other contexts triggered its integration within the global DHIS software framework. As a community data management, DHIS Tracker primary features include tracking patient enrolled to health programs, as well as the managing of single and anonymous events such as in-patient morbidity and mortality. Overall features available in the module include:

- Managing individual records: this feature permits the user to add details of persons including their relative's information;
- Managing Health Programmes: the tool also permits the creation of health programmes such as mother health, child health, HIV/AIDS and Tuberculosis (TB). This process includes the definition of the programme represented by its stages and their data elements;
- Enrolling persons into programmes: after having created the individual records and also the programmes this feature permits to relate each individual to a number of health programmes;
- Tracking action: the tool also allows the user to perform data entry related to specific programme stages;
- Run aggregation queries to produce data for other modules. i.e., data entered to the DHIS database using the Tracker module can be afterwards aggregated and presented using the different data/information presentation tools available within DHIS application such as data visualizer and GIS.

Several countries have tried to adjust DHIS Tracker to suite their local needs. Among those countries are Tanzania, Malawi and Guinea Bissau. Experiences gained from these three countries are outlined in the two sections below. Although presented as customization within countries the sections discusses the customization of DHIS Tracker in two phases. With this, the paper aim to describe how the maturity of the module was nurtured as well as the challenges faced during this process.

4.1 DHIS Tracker for Maternal Healthcare Management in Malawi

The customization of DHIS Tracker in this setting started with assessment of the Maternal and Child Health (MCH) programme data and requirements collected by a fellow researcher. This process was followed by the design of process flow diagrams (e.g., Figures 1 and 2). User

requirements were identified and specified during the design of these diagrams. With the requirements specified, the customization began.

To support a primary focus on functionality the three most important programs within the MCH comprising of antenatal care, delivery and immunization were included in the first customization plan. This decision brought two general advantages namely: it kept the focus on very few functionalities and it also reduced complexity related to number of specificities of the system. The customization process started by identifying and defining and creating the data elements needed in each of the selected programs into DHIS. After having all of them, the data elements were mapped (associated) with its respective programme (and programme stage), including the creation of client data (relationships, etc.). However, because the programmes are organized into stages it was required to define for each programme the stages and assign data elements to be collected in each program stage. With all these activities completed, data entry screens were created.

The process has not been as consistent as it should have been. The customizability of DHIS Tracker was assumed to make local adaptation easier; this could reduce time and complexity compared to developing a new system that performs almost the same as the generic DHIS Tracker. However, much time was spent in identifying which features are available in the tool; and at the end it was found that the module did not support all the required functionalities from the MCH. The problems encountered in trying to get DHIS Tracker into operation, include:

(a) The tool was not allowing mimicking the workflow of the MCH business processes (e.g., see Anti-natal care (ANC) in Figure 1; and Delivery in Figure 2). This process required decisions from the end-users. However, since the introduction of computer system at low level of Health Management Information System was not asked by the users, instead suggested by the researchers, it was difficult to assure the quality of the system through validation and test. Weaknesses were also identified with the tool including difficulty in sharing data elements between different programmes, i.e., the prediction that DHIS allow the sharing of data elements between different programmes stages belonging to the same program was not valid. The sharing of data elements feature is characterized as the possibility of allowing, for example, HIV test result of a woman captured by the HIV program to be displayed at the woman's interface of ANC or delivery services.
(b) The system was not allowing adding calculated data elements. Some of these data elements include date of next ANC appointment that could be calculated from the gestation age, and next programme stage and expected date of delivery from the last menstrual period date. Procedures to fill data on these data elements could not be automatized.
(c) The tool was released first with very weak documentation. The nonexistence of a critical mass using the system caused the lack of documentation to be even more resentful by customizers. For instance, customizers needed guidance in defining the best way of organizing the complexity of the MCH program within DHIS Tracker, which could not get through documentation.

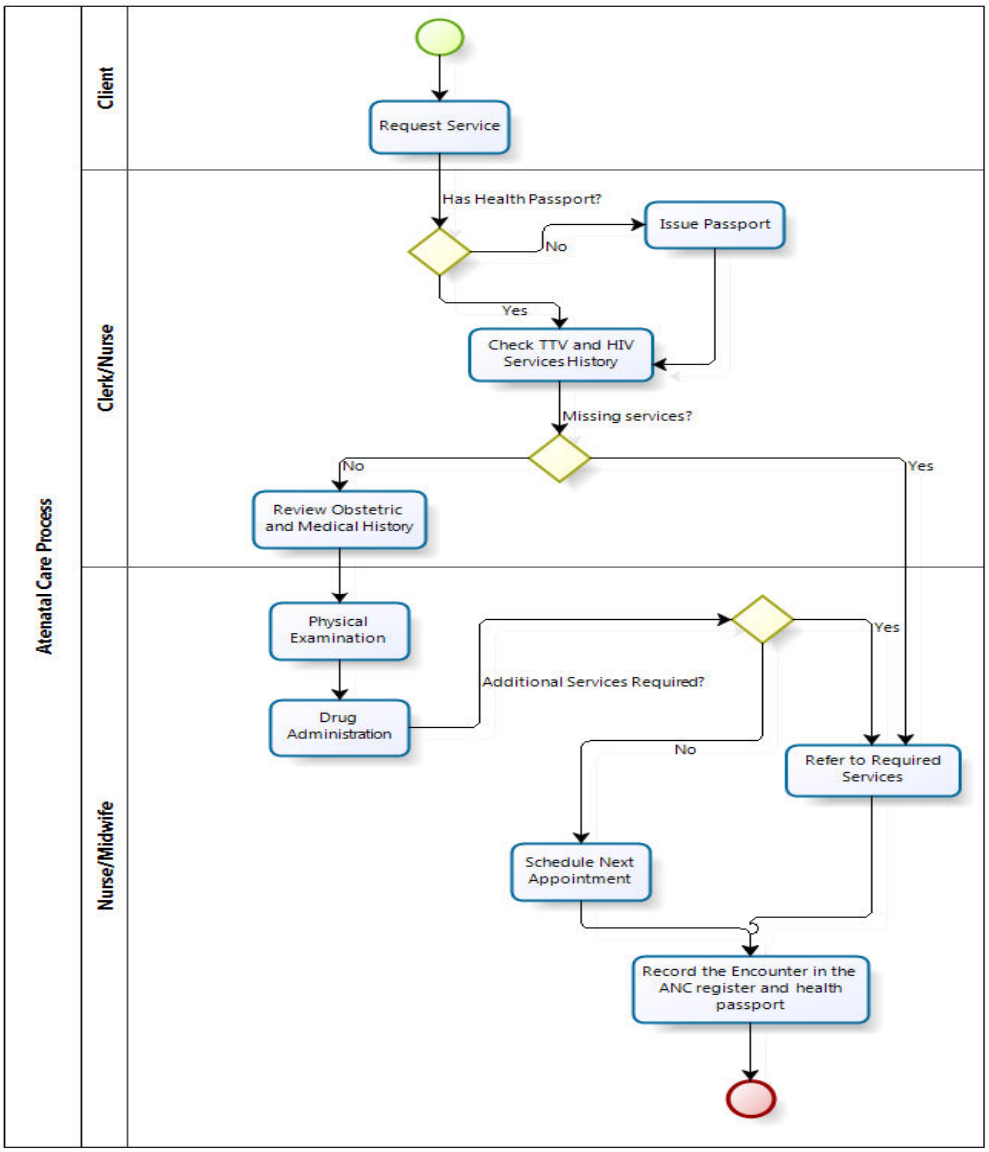

Figure1: ANC Process Flow

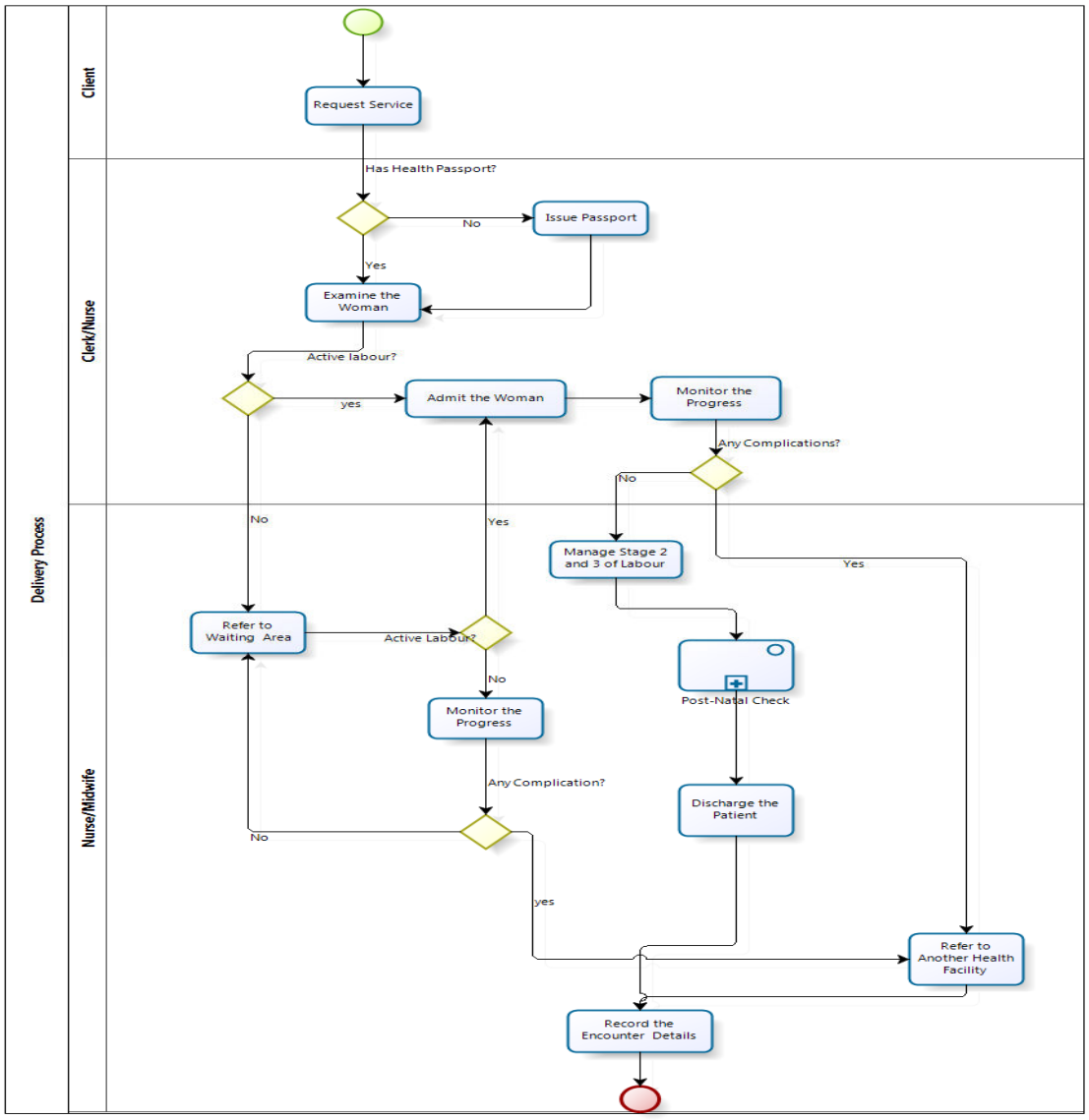

Figure 2: Delivery Process Flow

Furthermore, the customization has benefited from ideas and information of other individuals within and across the customization settings. For example, the local team had exchanged ideas, lessons and experiences with customizers of DHIS Tracker in Tanzania. It is through this process that the local team discovered that similar challenges had been faced in Tanzania. To solve those problems different approaches had been followed. While in Malawi customizers forwarded their problems to developers, in Tanzania customizer team decided to hire a developer to adjust the DHIS source code in order to match the local requirements.

4.2 DHIS Tracker for Management of Tuberculosis Patients in Guinea Bissau

Similar to Malawi, the customization of DHIS Tracker in Guinea Bissau was done in two broad stages: requirement analysis and software adaptation.

Requirement Analysis: Unlike Malawi where requirements were provided by a fellow researcher, here the TB program was assessed by the author. This process involved evaluating

the data collection, analysis and presentation tools. Subsequently, as in Malawi, process flow diagrams were designed and requirements identified.

Software Customization: According to its characteristics, of people involved and the time that was performed, this project can be regarded as a follow-up of the Malawi case. Due to the fact that customizers developed expertise during the customization in Malawi, lessons were transferred and applied in Guinea Bissau. Because of that, some of the troubles experienced in Malawi were no longer regarded as such in Guinea Bissau. This was also possible due to the improvements that DHIS Tracker framework has undergone since its introduction in India. Nevertheless, other problems were faced as follows:

(a) Since it was not possible to import and export data from/to DHIS Tracker, it was imperative to be connected to the server in online mode. Moreover, when the offline data entry feature was introduced within DHIS, it did not include Tracker data. Because of this, the suggestion to ease the introduction of historical data whereby central level staff was supposed to go to the hospital where TB patients were treated with their laptop with DHIS Tracker installed was discouraged.

(b) To benefit from features of new versions of the system, customization was to be done almost from scratch at least three times. Also changes made in the framework after the customized software was being used by data managers, required the redesigning of considerable parts of it. For example, considerable customization work had been rebuilt when DHIS Tracker framework allowed the definition of repeatable stages.

When the customization was complete, end-users were trained and given the opportunity to test the tool. It is, therefore, important to note that some of the problems highlighted earlier in Guinea Bissau were not verified in settings like Tanzania where changes have been made at the DHIS Tracker source code. In Tanzania customizers were able to export/import data from one DHIS Tracker version to another, and also to make DHIS Tracker to meet the end-users needs by changing the source code. However, they had other challenges. One of the most prominent challenge was motivated by the non-sharing of their source code with the framework developers. Because of that DHIS Tracker improvements could not match the changes made locally, thus they could not benefit from the periodically released enhancements. Figure 3 presents a screenshot of the customized application for tracking TB patients including its demographic information, stages and data entry screen.

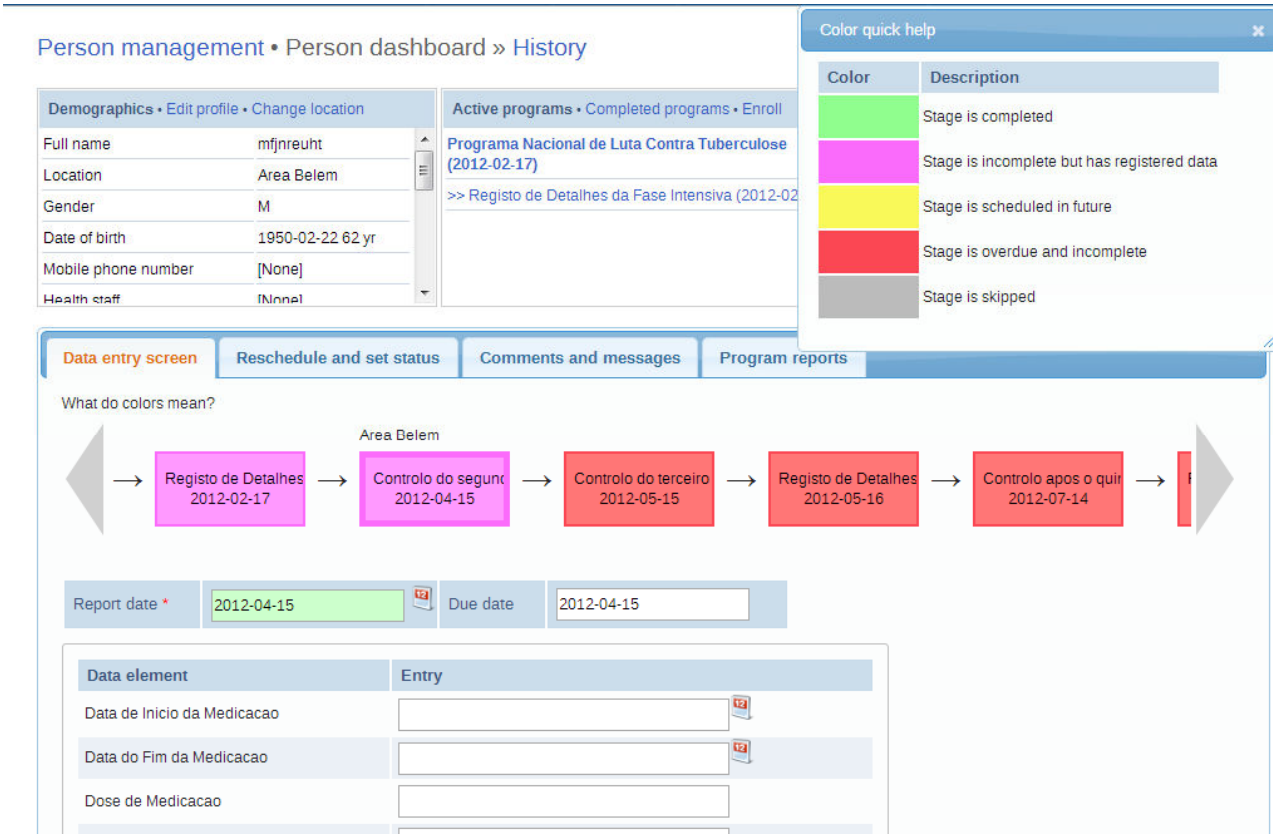

Figure 3. Screenshot of a Customized DHIS Tracker showing an example of a TB patient Dashboard

5. ANALYSIS AND DISCUSSION

Besides emphasizing the importance of health promotion, Primary Health Care has moved toward the use of appropriate information technology. Thus, to allow an unbroken decision support process that could support the very bottom level of Health Management Information System, DHIS Tracker was introduced. As a module developed to manage community based data, DHIS Tracker allows among other activities, the traceability of health data. This traceability increases the reliability as compared to aggregated data and leads to better quality. The implementation of DHIS Tracker in the settings evaluated in this paper, like many other studies of technology adoption, report success and failures (see e.g. Heeks (2006) and Pozzebon and Pinsonneault (2005)). While attempting to answer this paper research questions, the study generates two important lessons.

5.1 GSS can Challenge the Adaptation Process

One of the fundamental benefits of GSS is the possibility of allowing users to perform changes to the software to meet their own needs. Customization is one of the processes being, increasingly, adopted toward this aim. So far only few organizations benefit from the adaptability of GSS. Reasons are many and range from capacity and competence to complexity of the tools. Findings of this research have illustrated the existence of several barriers during the spanning process. For instance, in Malawi DHIS Tracker did not allow the mimicking of the business processes workflow, as well as the sharing of data elements between different health programmes. Likewise, in Guinea Bissau users could not use the offline data entry feature. This

problem became more challenging because of the absence of import/export functionality. To benefit from features of new versions of the framework, customization had to be done several times almost from scratch. Therefore, some of the customization could have been alleviated if documentation of the tool was available.

This ambiguity and complexity with boundary spanning work was highlighted by scholars studying boundaries (see e.g., Akkerman and Bakker (2011), Barrett and Oborn (2010), Levina and Vaast (2005)). In the healthcare settings this complexity is due to the existence of different work routines in each setting (see Ngoma et al. (2012)). In attempt to contribute to the discussion aiming at providing answer to the question of 'should the organization encourage or restrict the growth of practice surrounding IT?', analysis of the findings reveals that there is not a straight answer to this. For instance, as health systems are highly context-specific, there is no single set of practices that can be put forward as a model for process performance. The study findings indicate that customizers in Malawi were not able to align DHIS Tracker with some of the user requirements through source code. They chose to forward those requirements to the developers. Tanzanian customizers, on the other hand, performed changes on the source code level. Since the development was not coordinated with the developers the decision led to additional challenges. As presented in the findings Tanzania could not benefit from the enhancements of the generic DHIS Tracker. Thus, even though restrictions could eliminate challenges faced by customizers like the Tanzanians that decided to make changes at the source code, encouraging can therefore increase local developers' ownership and freedom and help to address challenges that were faced by customizers.

Thus, it can be asserted that there is a need to create balance between assigning the flexibility and controlling inclusion of new practices. And if customizers follow paths like those of customizers in Tanzania, the resulting code should be shared with the developers to allow future benefits from the enhancements of the tools, especially for systems that are still evolving such as the DHIS Tracker.

5.2 Development of GSS is informed by the Customization Process

As described in this study, DHIS Tracker was initiated in India and then expanded, first to Malawi and Tanzania and then to Guinea Bissau. The previous section have illustrated that across all sites, the generic framework provoked 'headaches' to customizers. However, after reporting the problems to the developers, Tracker failures were reduced. The result of this effort was observed in Guinea Bissau were customizers did not face the same challenges faced in Malawi and Tanzania, but faced other type of challenges. Some of the problems like the 'form name' were reported to developers and fixed later on. Others faced in Guinea Bissau were not within the developers' scope and had to be work-around by customizers.

It was, however, thanks to these pains that DHIS Tracker succeeded in reaching the maturity that showed both in Guinea Bissau and also in settings where it is currently being implemented. In the attempt to contribute to the discussion aiming at providing answers to the question of 'which boundaries should be spanned?', analysis of the findings reveals that DHIS Tracker development and customization was possible due to the collaborative arrangements available within the HISP network. Customization is here recognised as an important phase in the development of generic IT artefacts. It is, thus, during this phase that collaborative arrangements occur. People from different backgrounds, institutions or organizations collaborate in producing the tools and share ownership of the final products(Kamensky, Burlin, & Abramson, 2004). For instance, effective development, customization and use of software systems like DHIS is no longer possible if organizations work in isolation, but it requires joint

effort whereby professionals from many fields (e.g., healthcare, software development) and institutions (e.g., health authorities, universities, researchers) work together.

The findings of this study show that the collaborative arrangements besides helping in building systems that meet the needs of end-users permits the reuse of knowledge. This process contrary to the traditional view of customization where it is performed by vendors' representatives or the relatively recent view where the task is left to local people was facilitated by a technical middle-man acting as an intermediary between developers (experts about the system) and the end-users (experts about the domain).

Moreover, the process was not only between the domains of use and development, it also involved boundary spanning between different customization nodes. In between customizations, both the spanners and the developers accumulate expertise which allows them to operate the technology more efficiently as well as enhance or improve the software features. As observed in this study, technical middle-men were shared between domains, settings, and customizations (e.g., from Malawi to Guinea Bissau), thus providing a common ground for mutual understanding between diverse organization members including end-users, customizers and developers in a multi-cultural network. And, their understandings of the multiple cultures were imperative when features of the generic tools were discussed.

6. CONCLUSION

In the context of generic software systems, where software is developed envisioning a wide range of users, it is critical to have the tools customized to each particular use context. Lessons from this paper indicate that this process is influenced by the software customizability. An endeavour is then made to outline a basic mechanism through which customizability of generic software systems is achieved. Customization is, however, used as the mode through which boundaries and spanning activities are analysed. Software customizability is in this case a sequence of boundary spanning activities, i.e., customizability is seen to emerge as materialization of lessons learned after spanning domains, settings, and customizations nodes.

Moreover, in attaining customizability the aspect of encouragement or restriction of growth of the generic software system need to be taken into account. The paper argues that customization is to be expected in the development of working systems, as an outcome of a 'learning-by-trying' process. Through this process there is a need of balancing decisions, i.e., developers need to encourage local development with some restrictions on the tasks to be performed. If properly performed this can facilitate the handling of uncertainty involved in customer needs and design changes.

By evaluating the boundaries spanned during software customization, together with the issue of encouragement or restriction of the growth of practice surrounding IT artefacts', the paper provides important contributions to the boundary theory that is particularly relevant for institutions that will seek to enhance customizability of generic software systems through the mechanism of customization.

7. REFERENCES

Akkerman, S. F., & Bakker, A. (2011). Boundary Crossing and Boundary Objects. *Review of Educational Research, 81*(2), 132-169. doi: 10.3102/0034654311404435

Barrett, M., & Oborn, E. (2010). Boundary object use in cross-cultural software development teams. *Human Relations, 63*(8), 1199-1221. doi: 10.1177/0018726709355657

Bell, G. D. (1966). Predictability of Work Demands and Professionalization as Determinants of Workers' Discretion. *The Academy of Management Journal, 9*(1), 20-28.

Braa, J., Monteiro, E., Sahay, S., Staring, K., & Titlestad, O. H. (2007). *SCALING UP LOCAL LEARNING - EXPERIENCES FROM SOUTH-SOUTH-NORTH NETWORKS OF SHARED SOFTWARE DEVELOPMENT*. Paper presented at the 9th International Conference on Social Implications of Computers in Developing Countries, São Paulo, Brazil.

Claeson, M., & Waldman, R. J. (2000). The evolution of child health programmes in developing countries: from targeting diseases to targeting people. *Bulletin of the World Health Organization, 78*(10), 1234 - 1245.

Curro, F. A., Craig, R. G., Vena, D., & Thompson, V. P. (1995). The role of interim analysis as a quality assurance function in practice-based research network clinical studies. *Compendium of continuing education in dentistry, 30*(6), 352-354.

Daniels, H., Edwards, A., Engeström, Y., Gallagher, T., & Ludvigsen, S. R. (2010). *Activity Theory in Practice: Promoting Learning Across Boundaries and Agencies*. London and New York: Routledge Taylor & Francis Group.

Dourish, P. (1995). Developing a Reflective Model of Collaborative Systems. *ACM Transactions on Computer-Human Interaction, 2*(1), 40-63.

Dourish, P., & Edwards, W. K. (2000). A Tale of Two Toolkits: Relating Infrastructure and Use in Flexible CSCW Toolkits. *Computer Supported Cooperative Work: The Journal of Collaborative Computing, 9*(1), 33-51.

Fischer, G., & Scharff, E. (2000). *Meta-design: design for designers.* Paper presented at the Conference on Designing interactive systems, processes, practices, methods and techniques 2000, New York, New, York.

Franke, N., & Hippel, E. v. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research Policy, 32*(7), 1199-1215. doi: Doi: 10.1016/s0048-7333(03)00049-0

Heeks, R. (2006). Health information systems: Failure, success and improvisation. *International Journal of Medical Informatics, 75*(2), 125-137. doi: DOI: 10.1016/j.ijmedinf.2005.07.024

Iivari, N. (2010). Discursive construction of 'user innovations' in the open source software development context. *Information and Organization, 20*(2), 111-132. doi: http://dx.doi.org/10.1016/j.infoandorg.2010.03.002

Jacobs-Huey, L. (2002). The Natives Are Gazing and Talking Back: Reviewing the Problematics of Positionality, Voice, and Accountability among "Native" Anthropologists. *American Anthropologist, 104*(3), 791-804. doi: 10.2307/3567257

Kamensky, J. M., Burlin, T. J., & Abramson, M. A. (2004). Networks and Partnerships: Collaborating to Achieve Results That No One Can Achieve Alone. In J. M. Kamensky & T. J. Burlin (Eds.), *Collaboration Using Networks and Partnerships* (pp. 3-20). USA: Rowman & Littlefield Publishers, Inc.

Levina, N., & Vaast, E. (2005). The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems. *MIS Quarterly, 29*(2), 335-363.

Lindgren, R., Andersson, M., & Henfridsson, O. (2008). Multi-contextuality in boundary-spanning practices. *Information Systems Journal, 18*(6), 641-661. doi: 10.1111/j.1365-2575.2007.00245.x

Mason, J. (2002). *Qualitative Researching* (Vol. 2nd Ed.): London.

Michaud, J. W. (1999). *A Software Customization Framework.* (MASTER), University of Victoria.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis* (Vol. 2nd Ed.): Newbury Park.

Ngoma, C. A., Kimaro, H. C., Igira, F. T., Mukulu, J., Kaasbøll, J. J., Yona, A., . . . Pilyimo, R. (2012). *Challenges in Implementing a Computerized Name-based Information Tracking System: Practical Experiences from Maternal Health Care*.

Pollock, N., Williams, R., & D'Adderio, L. (2007). Global Software and its Provenance: Generification Work in the Production of Organizational Software Packages. *Social Studies of Science, 37*(2), 254-280. doi: 10.1177/0306312706066022

Pozzebon, M., & Pinsonneault, A. (2005). Global-local negotiations for implementing configurable packages: The power of initial organizational decisions. *The Journal of Strategic Information Systems, 14*(2), 121-145. doi: DOI: 10.1016/j.jsis.2005.04.004

Randell, R. (2003). User customisation of medical devices: the reality and the possibilities. *Cognition, Technology & Work, 5*(3), 163-170. doi: 10.1007/s10111-003-0124-0

Sahay, S., & Walsham, G. (2006). Scaling of health information systems in India: Challenges and approaches. *Information Technology for Development, 12*(3), 185-200.

Schmidt, D. C., Gokhale, A., & Natarajan, B. (2004). Leveraging Application Frameworks. *Queue, 2*(5), 66-75. doi: 10.1145/1016998.1017005

Star, S. L. (2010). This is Not a Boundary Object: Reflections on the Origin of a Concept. *Science, Technology & Human Values, 35*(5), 601-617. doi: 10.1177/0162243910377624

Stock, R. (2006). Interorganizational teams as boundary spanners between supplier and customer companies. *Journal of the Academy of Marketing Science, 34*(4), 588-599. doi: 10.1177/0092070306288765

Sullivan, K. J., Griswold, W. G., Cai, Y., & Hallen, B. (2001). The structure and value of modularity in software design. *SIGSOFT Softw. Eng. Notes, 26*(5), 99-108. doi: 10.1145/503271.503224

Swaminathan, J. M. (2001). Enabling Customization Using Standardized Operations. *CALIFORNIA MANAGEMENT REVIEW, 43*(3), 125-135.

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems, 4*(2), 7.

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems, 15*, 320-330.

YEAGER, B. (2012). Improving Access to Maternal Health Commodities: A Systems Approach. Retrieved from http://blog.msh.org/2012/09/05/improving-access-to-maternal-health-commodities-a-systems-approach/