

# A Framework to Assess and Address Human Capacities Needed to Leverage Open Source Software Platforms in Developing Countries

Brown Msiska<sup>(✉)</sup> and Petter Nielsen

Department of Informatics, University of Oslo, Oslo, Norway  
bmsiska@gmail.com, pnielsen@ifi.uio.no

**Abstract.** While open source health information software platforms provide developing countries a low-cost, quick and less risky way to build health information systems as compared to in-house solutions, human resource capacity challenges can limit their ability to leverage such platforms. Drawing from a case study focusing on the deployment and operation phases of the DHIS2 platform in Malawi, we observe open source software platforms require a range of human resource capacities that go beyond capacity to use the platform. To fully leverage open source health information software platforms entails the availability of platform usage capacity, platform deployment capacity, platform customisation capacity and platform module development capacity. Most capacity building initiatives for information systems in developing countries have been short-term efforts focused on initial end user capacity to use such systems. However, to cope with rapid innovations and evolution associated with open source software platforms, capacity building ought to be a continuous process encompassing a range of human resource capacities not only use of the platform.

**Keywords:** Health information systems · Open source software · Software platforms · Software ecosystems · Developing countries

## 1 Introduction

Health Information System (HIS) integrates data collection, processing, reporting, and use of the information necessary for improving health service effectiveness and efficiency through better management at all levels of the health services [1]. In most developing countries, there are major problems with the quality of health information which is often incomplete, inaccurate and not available at the right time for the right people. Consequently, strengthening HIS is recognised as one of the key activities required to improve the effectiveness and efficiency of health services in developing countries [2].

Resource constraints that characterise most developing countries form a key barrier to global HIS strengthening efforts. Due to high budget deficits, there is underinvestment towards HIS strengthening efforts which limits their ability to acquire appropriate software to drive their HIS operations [2, 3]. To mitigate this, a common strategy in developing countries is to adopt and adapt a generic health information system for use

in the local context. This is different than starting from scratch. Adopting generic systems is cheaper, quicker and less risky compared to in-house software development [4]. By adopting generic health information systems, developing countries reduce the cost and more importantly the development time of the systems. Generic systems contain a standardised core made up of common components that remain stable across different contexts and customisable components that vary across contexts and over time [5]. Developing countries leverage these customizable components of generic health information systems to make them fit for use in the local context.

The extent of customisation afforded by generic systems varies. Some generic systems provide resources for developing third party applications (or components) which allow the end user community to extend the functionality of the systems in order to meet specific needs within context use. Such generic systems act as software platforms on which other applications required within context of use can be built by local developers or other third parties. A software platform is a software-based system with an extensible codebase that provides shared core functionality and resources with which derivative application can be created [6, 7]. In this paper, the term health information software platform is used to refer to an extensible generic health information system that provides resources with which derivative applications can be built. This is typical of some contemporary generic health information systems being used in developing countries; such as Open Medical Record System (OpenMRS), District Health Information Software (DHIS2), and Open Logistics Information System (OpenLMIS) to mention a few.

Traditionally, software is distributed under a proprietary license which requires clients to pay annual license fees for using the software and prevents them from modifying and redistributing the software [8]. With the prevailing resource constraints, the cost of ownership and freedom to modify software to fit local settings are major concerns for developing countries [3]. Open-source software provides an opportunity to build low-cost health information systems allowing countries with modest resources access to modern data analysis and visualization tools [9]. Open-source software, abbreviated as OSS, is software that is made available with source code and is provided under a software license that permits users to study, change, and improve the software [10]. By granting the user community access to the source code, open source software allows them to create and extend the software with local innovations that fit the local context. Furthermore, open source software is viewed as a tool for technical self-sufficiency as it eliminates the need for outside consultants and reduces costs of ownership of software [11]. Consequently, the use of generic open source health information systems in developing countries is common [2, 3]. The convergence of open source software and software platform approaches in the design and development of health information systems has given rise to open source health information software platforms with DHIS and OpenMRS as popular examples.

HIS strengthening initiatives in developing countries have a history of failure and unsustainability resulting from a number of factors. One important contributing factor is the lack of human resource capacity to use, develop and maintain the systems [12]. Furthermore, finding skilled developers in developing countries is a struggle due to lack of training and brain drain [13]. Other studies also show that developers participating in and contributing to open source projects are predominantly from developed

countries [14, 15]. However, HIS cannot deliver expected benefits unless they are supported by appropriate human resource capacity locally [12]. Thus, to fully leverage open source health information software platforms appropriate human resource capacity must exist.

Software platforms are to some extent “half products” which have to be customised and/or extended before they can be fit for use in a particular context [16]. They constitute a departure from an era where end-users got a fully-fledged solution from a software vendor to an era where software vendors deliver solutions that must be completed by the end-user community within the context of use. Because of this, the capacity requirements for leveraging open source health information platforms vary from those of traditional HIS software. Platforms do not only allow for, support and encourage, but mandates local initiatives and innovations. The purpose of this paper is therefore to empirically address the question: what human resource capacities are required to leverage open source health information software platforms within context of use? By addressing this question the paper aims to strengthen our understanding of the implications open source health information software platforms have on HIS capacity building initiatives in developing countries. Beyond identifying concrete capacities in the case of HIS in Malawi, we also contribute with a general framework to assess and address human capacities needed to leverage open source software platforms in developing countries.

## 2 Software Platforms and Ecosystems

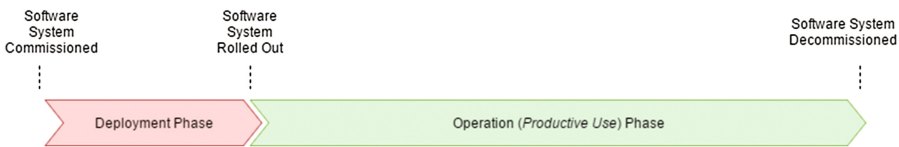
A software platform is an extensible codebase of a software based system that provides core functionality as well as an interface shared by the modules that interoperate with it [17]. Building on this definition others have gone further to describe a software platform as a software based system with extensible codebase that provides shared core functionality and resources from which derivative applications are generated [6, 7]. A software ecosystem is the software platform and the collection of modules specific to that platform [17]. Thus, the software platform is just one among other parts that make up a software ecosystem. Within the ecosystem, the software platform is complemented by modules (or applications). Modules are add-on software subsystems that connect and add new functionality to the software platform [17]. Associated with software platforms and software ecosystems is a shift from a closed software product-line based development approach to an ecosystem based development approach [16].

With the ecosystem based development approach, the software platform is rarely a solution itself; its functionality is exposed to end users through applications running on top of it. This shift in development approach comes with its own challenges. A key challenge emerges from the separation between the platform developer and platform’s context of use, as well as the diversity of user requirements. Challenges faced by platform developers include making an informed decision on what applications or services to develop [18] and how to effectively respond to turbulent information processing requirements within context of use [7]. These challenges make the involvement of third party application developers close to context of use increasingly attractive for software platform developers [19, 20] as it allows them to focus on the core extensible

base of the platform and defer satisfying specific user needs to third party developers. The proximity of third party developers to context of use enables them to make informed decisions on add-on modules and to develop them in response to particular needs. At the same time, deferring development to third parties creates new local capacity requirements within contexts of use. This makes the human resources capacity challenge prominent in initiatives aimed at assuring sustainability of health information system platforms in developing countries. This is the key challenge addressed in this paper.

### 3 Software Platforms and Human Resource Capacity Requirements

Once commissioned, a software system undergoes a number of phases within its context of use. For the purpose of this paper, distinction is made between two major phases: deployment phase, and operation phase. Software deployment is a process comprising all activities carried out in order to make a software system available for use [21]. This includes among other things setting up the required hardware and software environment; installing the software system in question; piloting and adapting it for local use; and testing it against functional and nonfunctional requirements to determine its readiness for use. Once deemed ready for use, the software system is rolled out into operation; ushering it into the operation (or productive use) phase (Fig. 1).



**Fig. 1.** Software system timeline within context of use

Once the software system is put into operation, anomalies are discovered, operating environments change, and new user requirements emerge and the software system must change accordingly [22]. The process of changing a software system or its component to correct faults, improve performance or other attributes, or adapt to a changed environment after it has been put into operation is called software maintenance [23, 24]. Software maintenance activities span a system's productive life cycle and 70% of all effort on a software system is estimated to be expended on maintenance alone [25]. A range of human resource capacities are therefore required during both the deployment and operation phases of a software system. For software platforms, in particular, the lack of appropriate human resource capacities in the deployment and operation phases can constrain their implementation, maintenance and hence sustainability within context of use. Understanding what human resource capacities are required for open source software platforms within each of these phases can therefore be instrumental in ensuring their success and sustainability in developing countries.

## 4 Methodology

This paper is based on a qualitative case study carried out in Malawi with DHIS2 as the focal software platform under study. DHIS2 is a web based, free, generic and open source health information software platform. It is currently the leading solution for aggregate health data and is being used in more than 50 developing countries by government ministries, donor agencies and NGOs [26]. DHIS2 is a flexible metadata-driven software that affords implementers the flexibility to customize its data model to fit the data needs of particular end-users. Besides the flexible data model, DHIS2 also allows customisation of data entry forms and reports. In addition, DHIS2 comes with a RESTful Web API that allows extension of its functionality through third party innovations built using common web technologies such as JavaScript, CSS and HTML5. In addition to applications running on top of it through the Web API, DHIS2 also allows development of custom software modules that sit side by side with its core modules. The introduction of such extensibility features into DHIS2 has seen it evolve from a traditional health information system to a software platform supporting open innovation through extensions by other developers within its ecosystem other than the core software development team at the University of Oslo [26, 27].

DHIS2, in Malawi, falls under the custody of the Central Monitoring and Evaluation Division (CMED) in the Ministry of Health (MoH). CMED is in charge of collecting and analyzing aggregate data used to monitor and evaluate various health programmes run by MoH. The DHIS2 platform is used by CMED and other stakeholders to collect, store, analyse and visualize aggregated data for decision making purposes. Therefore, the case study largely focused on CMED and key stakeholders involved in the deployment and operation of DHIS2 in Malawi. The aim of the case study was to address the question what human resource capacities are required to leverage open source health information software platforms within context of use. To address this question, we started by purposively sampling [28] key personnel from CMED and other stakeholders involved in the deployment and operation phases of DHIS2 as respondents for the study. The respondents were drawn from Ministry of Health and CMED, HISP Malawi, University of Malawi (UNIMA) and Baobab Health Trust. We then conducted semi-structured interviews with the selected respondents in order to establish prevailing human resource capacity requirements and challenges during the deployment phase and the subsequent operation phase of DHIS2 (Table 1).

**Table 1.** Respondents

Organisation	No. respondents	Designation
Ministry of Health/CMED	5	Director, CMED Chief Technical Assistant, HMIS DHIS2 Technical Assistants (3)
HISP Malawi	3	Board Members (3)
Baobab Health Trust (BHT)	4	Executive Director Software Development Manager DHIS2 Integration Team Members (2)
UNIMA	1	HISP Malawi Representative

Further data was collected through participatory observation which involved attending stakeholder meetings, training workshops, and working with the DHIS2 implementation team in Malawi comprising of staff from CMED/HISP Malawi and other strategic local partners. As part of these observations, one of the researchers worked as part of the DHIS2 implementation team in Malawi in addition to being a member of the team of trainers in a DHIS2 Application Development Workshop that took place in March 2016. In addition, data was also collected through document reviews comprising of reports on DHIS2 in Malawi by practitioners and other researchers.

The data yielded from the interviews, participatory observations and document reviews was thematically analysed [28] resulting in grouping of human resource capacities and challenges identified according to the two phases: deployment and operation. The capacities and challenges were further analysed with respect to the kind of human resource capacity; resulting in a framework which we present in our discussion. In the next section, we describe the human resource capacity requirements and challenges during the deployment and operation of DHIS2 in Malawi identified during the case study. This is followed by the discussion and later on, concluding remarks.

## **5 DHIS2 Software Platform in Malawi**

A case study was carried out in Malawi focusing on the deployment and operation phases on the DHIS2 software platform. The aim of the case study was to establish prevailing human resource requirements and challenges in each of the phases. The results of the case study are presented in Sects. 5.1 and 5.2.

### **5.1 Human Resource Capacity Requirements in the Deployment Phase of the DHIS2 Platform in Malawi**

DHIS2 deployment in Malawi commenced in 2009 as a pilot involving three districts; Blantyre, Zomba and Lilongwe. The pilot project ran for a period of three years and resulted in DHIS2 being rolled out to all districts and all health programmes in Malawi in 2012. The deployment of DHIS2 in Malawi involved: setting up a web server on which to run the platform; installing the DHIS2 platform on the web server; defining metadata for the platform in terms organizational units, users and user roles, data elements and indicators for data sets selected for the pilot. The deployment of DHIS2 therefore required appropriate human resources capacity to set up the web server, installing DHIS2, and define required metadata for selected data sets. Once the platform was deployed the default data entry forms and reports generated by DHIS2 turned out significantly different from the paper based forms and reports that were currently in use by end users. To preserve familiarity and ease the transition towards using DHIS2, custom reports and forms had to be designed and implemented. Implementing such custom forms and reports required the availability of human resource capacity to create custom forms and reports in DHIS2.

The human resource capacity requirements mentioned above came with a challenge: the lack of ICT personnel in CMED to carry out the deployment and customization tasks

required for the deployment of DHIS2. To make ICT human resources available to CMED for the deployment of DHIS2 a local HISP node, HISP Malawi, was established. With funding acquired through the University of Oslo, HISP Malawi recruited two programmers on contract and placed them on secondment to CMED. These were joined by ICT staff from the University of Malawi constituent colleges: Chancellor College, College of Medicine and the Malawi Polytechnic. The pool of ICT personnel from HISP Malawi and University of Malawi underwent training on DHIS2 to enable them implement the deployment and customization tasks related to the pilot. In addition to the technical human resource capacities, there was a need for end users participating in the pilot study to be able to use DHIS2. Therefore, end users participating in the pilot study underwent training to enable them use DHIS2.

## **5.2 Human Resource Capacity Requirements in the Operation Phase of the DHIS2 in Malawi**

Once a decision was made to roll out DHIS2 to all districts in Malawi, the demand for human resource capacity to use the software platform grew. As a result, a series of training workshops have been conducted targeting end-users of the software platform. Such trainings included, among others, a training of trainers workshop in August 2012 and a series of DHIS2 mobile trainings [29]. “The end user training employed a cascade approach” (Director, CMED). Trainer of Trainers (TOTs) at national level were identified and trained and the TOTs in turn trained district trainers who undertook training of health workers in their respective districts. However, we found out during the case study that end-user training has not been done regularly. As a result, there is still a backlog on untrained staff. Furthermore, changes in newer versions of DHIS2 have left a number of end-users requiring re-training.

At the same time, rolling out DHIS2 to all health programmes under the ministry of health required further deployment and customisation tasks in terms of metadata and implementation of custom forms and reports for the programmes that were not part of the pilot. Furthermore, frequent releases of newer versions of DHIS2 necessitated deploying and transitioning to newer versions of DHIS2. As a result, the demand for human resource capacity to deploy and customize the software platform grew as well. To catch up with advances in DHIS2 that came with each release members of the DHIS2 technical team have attended a number of regional DHIS2 Academies. Through the academies they acquired a range of knowledge required to install and customize DHIS2.

During the period DHIS2 has been in operation there have been a number human resource capacity challenges. First of all, staff turnover involving technical staff at HISP Malawi has threatened the day to day operations of DHIS2. All technical assistants recruited by HISP Malawi and placed on secondment to CMED between 2012 and 2015 have left for various reasons including funding for their retention. Currently, HISP Malawi has two technical assistants recruited towards the end of 2015. These were fresh graduates from the University of Malawi and have had to attend a series of DHIS2 academies to bring them up to speed with DHIS2.

In addition, the lack of ICT staff in CMED has left it dependent on external expertise, both local and foreign, to keep DHIS2 in operation. The MoH, like all other

ministries in the Malawi Government, has an ICT department which has an allocation of ICT staff from the Department of e-Government in the Office of the President and Cabinet. However, until recently none of the ICT staff in the ICT department have been working with CMED on DHIS2. CMED has therefore relied on HISP Malawi and other external expertise to keep DHIS2 in operation. CMED sees the continued reliance on external expertise as a threat to the sustainability of DHIS2 and has been lobbying government to adjust its staff establishment to include ICT personnel. “We came up with a position paper requesting for ICT personnel under CMED ... but things take time in government so we are not sure when that will happen” (Chief Technical Assistant, CMED). The decision to adjust CMED’s staff establishment is however still outstanding.

Furthermore, some problems and requirements CMED has had with respect to DHIS2 required changing or developing new DHIS2 modules. Handling such requirements and problems required human resource capacity to develop or modify DHIS2 modules. These have usually been reported to DHIS2 core developers in Norway because of lack of necessary human resource capacity locally. As part of efforts to address this gap, In March 2016, a DHIS2 Application Development workshop took place at University of Malawi, Chancellor College from 7th March to 16th March 2016 [30]. The training introduced participants to the DHIS2 API and how to develop DHIS2 applications using the API. The training was funded by University of Oslo with support from UNICEF. It attracted participants from Kenya, Ethiopia, Zambia and Malawi including the two technical assistants at CMED.

## 6 Discussion

Analysing the deployment and operation phases of DHIS2 in Malawi reveals four categories of human resource capacity needed to leverage open source software platforms. Before a software platform can be put into operation it must be deployed first and this requires the availability of platform deployment capacity which includes ability to setup the platform operating environment and install the software platform. Once the software platform is installed, its customisable components have to be customised to fit local needs. This entails platform customisation capacity. Historically, donor driven HIS projects in developing have used foreign experts to mitigate the gap in deployment and customisation capacity in the deployment phase.

Once the system is put into operation it requires human capacity for its maintenance, including the development of new platform modules. This entails platform module development capacity. Such capacity is instrumental to third party innovations which respond to specific needs within context of use. Maintenance often happens at a time when donor and external support is no longer available leading to sustainability challenges where local capacity is lacking. Both during deployment and maintenance the software system is subject to use by end users. In the deployment phase this happens as a result of piloting and testing of the system against end user requirements. Therefore in both phases there is a requirement for human capacity to use the software platform, hereby referred to as platform usage capacity. Furthermore, during the

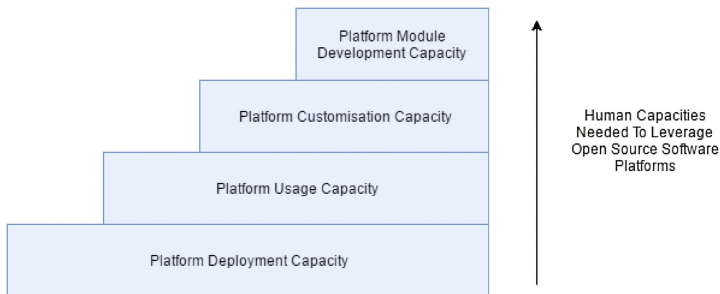


**Table 2.** Human capacity requirement for software platforms within context of use

Phase	Key activities	Human capacities required
Deployment	<ul style="list-style-type: none"> <li>– Setting up hardware and software environment</li> <li>– Installation of the software platform</li> <li>– Customisation</li> <li>– Testing/Piloting system</li> </ul>	<ul style="list-style-type: none"> <li>– Platform deployment capacity</li> <li>– Platform customisation capacity</li> <li>– Platform usage capacity</li> </ul>
Operation	<ul style="list-style-type: none"> <li>– Correcting system faults</li> <li>– Upgrading to newer software versions</li> <li>– Customising new versions</li> <li>– Develop platform modules</li> <li>– Test implemented changes</li> </ul>	<ul style="list-style-type: none"> <li>– Platform deployment capacity</li> <li>– Platform customisation capacity</li> <li>– Platform module development capacity</li> <li>– Platform usage capacity</li> </ul>

maintenance phase there are episodes of deployment where the software system is upgraded to a newer version. These capacity needs are summarized in Table 2.

The extent to which developing countries are able to leverage open source software platforms is, therefore, subject to the availability of human capacities to deploy, customize and use the platforms; complemented by capacity to develop platform modules in response to new requirements or those not adequately addressed by platform developers. This we summarise in the model in Fig. 2, showing a ladder of human resource capacities required to leverage software platforms within context of use. The model can act as a framework informing stakeholders implementing of open source health information software platforms in developing countries what human resource capacities to put in place in order to fully leverage the software platforms. At the same time, it could be used to inform efforts to assess and address gaps in human resource capacities needed to leverage open source platforms where they have been implemented.

**Fig. 2.** A model for human capacities needed to leverage open source software platforms

Many of donor-driven HIS in developing countries have ended up as unsustainable and/or failures due to a lack of local human resource capacity left behind after implementation [31]. During implementation, donors and their agents have traditionally filled the human resources gap by engaging foreign experts at the expense of building local expertise [12]. While this works well in the short term it creates long

term challenges with maintenance and sustainability. Without requisite capacity, locals fail to support and maintain the solutions leading to gradual decay and obsolescence of the solutions as they fail to respond to emerging needs within context of use. As shown in the model above, open source software platforms come with an implicit demand for local capacity not only in terms of use but also in terms of customizing and extending the platform to meet local needs. This supports the argument made by Sahay and Walsham [32] who state that health information systems need to be accompanied by the scaling of local human resources capacity at least two levels: level of end users and level of the implementation team.

The introduction of health information systems in developing countries is often accompanied by short-term training aimed at building the capacity of end-users to use the systems. This is important, but not enough to address long-term learning needs of end users and the implementation and maintenance team to stay up to date as the health information system evolves overtime. Capacity building is a continuous process [12]. To cope with rapid innovations and evolution associated with health information software platforms and open source software platforms in general, capacity building ought to be a continuous process and include more than the platform usage capacity. Associated with increasing complexity of health information systems is the need to scale the technical competence of users and that of the implementation team responsible for providing technical support to the users and the user organization [32]. With the shift towards health information software platforms, the issue of human resources capacity building in developing countries involves not only building capacity of local end-users and local implementation team but also ensuring that there is continuous learning to enable them cope with the rapid evolution and innovations that characterise software platforms.

## 7 Conclusion

Open source health information software platforms provide developing countries a low-cost, quick and less risky way to build health information systems compared to developing in-house solutions. However, software platforms also place new capacity requirements within context of use as compared to traditional off-the-shelves or commissioned software systems. The availability of human resource capacity to deploy, use and maintain a health information software platform can have an influence on the extent to which developing countries can leverage the potential of these platforms. Thus, capacity building initiatives around health information software platforms should strive to build this range of capacities. Short-term training aimed at building the capacity of end-users to use the systems is important, but not enough to address long-term learning needs of end users and the implementation and maintenance team to stay up to date as the health information system evolves overtime. To cope with rapid innovations and evolution associated with open source software platforms, capacity building ought to be a continuous process covering a range of human resource capacities not only use of the platform.

## References

1. WHO: Developing Health Management Information Systems: A Practical Guide For Developing Countries. World Health Organisation, Geneva (2004)
2. Karuri, J., Waiganjo, P., Orwa, D., Manya, A.: DHIS2: the tool to improve health data demand and use in Kenya. *J. Health Inform. Dev. Countries* **8** (2014)
3. Sheikh, Y.H., Bakar, A.D.: Open source software solution for healthcare: the case of health information system in Zanzibar. In: Popescu-Zeletin, R., Jonas, K., Rai, Idris A., Glitho, R., Villafiorita, A. (eds.) AFRICOMM 2011. LNICSSITE, vol. 92, pp. 146–155. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-29093-0\\_14](https://doi.org/10.1007/978-3-642-29093-0_14)
4. Bansler, J.P., Havn, E.C.: Information systems development with generic systems. In: ECIS 1994, pp. 707–718 (1994)
5. Silsand, L., Ellingsen, G.: Generification by translation: designing generic systems in context of the local. *J. Assoc. Inf. Syst.* **15**, 177–196 (2014)
6. Eck, A., Uebernickel, F., Brenner, W.: The generative capacity of digital artifacts: a mapping of the field. In: Proceedings of PACIS 2015 (2015)
7. Ghazawneh, A., Henfridsson, O.: Balancing platform control and external contribution in third-party development: the boundary resources model. *Inf. Syst. J.* **23**, 173–192 (2013)
8. Fogel, K.: Producing Open Source Software, 1st edn. O’Reilly, Sebastopol (2005)
9. Yi, Q., Hoskins, R.E., Hillringhouse, E.A., Sorensen, S.S., Oberle, M.W., Fuller, S.S., et al.: Integrating open-source technologies to build low-cost information systems for improved access to public health data. *Int. J. Health Geogr.* **7**, 29 (2008)
10. Kandar, S., Mondal, S., Ray, P.: A review of open source software and open source movement in developing countries. *Int. J. Comput. Sci. Inform.* **1**, 89–93 (2011)
11. Vital Wave Consulting: mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World, Washington, D.C. (2009)
12. Kimaro, H.C.: Strategies for developing human resource capacity to support sustainability of ICT based health information systems: a case study from Tanzania. *Electron. J. Inf. Syst. Dev. Countries* **26**, 1–23 (2006)
13. Roets, R., Minnaar, M., Wright, K.: Open source: towards successful systems development projects in developing countries. In: Proceedings of the 9th International Conference on Social Implications of Computers in Developing Countries, Sao Paulo, Brazil (2007)
14. Paudel, B., Harlalka, J., Shrestha, J.: Open technologies and developing economies. In: Proceedings of CAN InfoTech (IT) Conference (2010)
15. Weerawarana, S., Weeraratne, J.: Open Source in Developing Countries. Sida, Stockholm (2004)
16. Dittrich, Y.: Software engineering beyond the project – sustaining software ecosystems. *Inf. Softw. Technol.* **56**, 1436–1456 (2014)
17. Tiwana, A., Konsynski, B., Bush, A.A.: Research commentary—platform evolution: coevolution of platform architecture, governance, and environmental dynamics. *Inf. Syst. Res.* **21**, 675–687 (2010)
18. Henfridsson, O., Lindgren, R.: User involvement in developing mobile and temporarily interconnected systems. *Inf. Syst. J.* **20**, 119–135 (2010)
19. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, pp. 111–119. Carnegie Mellon University (2009)
20. Boudreau, K.J., Lakhani, K.R.: How to manage outside innovation. *MIT Sloan Manag. Rev.* **50**, 69–75 (2009)

21. Carzaniga, A., Fuggetta, A., Hall, R.S., Heimbigner, D., Van Der Hoek, A., Wolf, A.L.: A characterization framework for software deployment technologies. DTIC Document (1998)
22. Pigoski, T.M.: *Practical Software Maintenance: Best Practices for Managing Your Software Investment*. Wiley, New York (1997)
23. Sommerville, I.: *Software Engineering*, 9th edn. Pearson Education, Boston (2011)
24. Van Vliet, H.: *Software Engineering: Principles and Practice*. Wiley, New York (2007)
25. Ogheneovo, E.E.: On the relationship between software complexity and maintenance costs. *J. Comput. Commun.* **02**, 1–16 (2014)
26. PATH: *An Interim Review of the Health Information Systems Programme - University of Oslo - with Recommendations for Future Action*, Seattle, September 2016
27. Polak, M.: *Platformisation of an Open Source Software Product: Growing up to be a Generative Software Platform*. University of Oslo, Oslo (2015)
28. Creswell, J.W.: *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage, Los Angeles (2009)
29. HISP Malawi: DHIS2 TOT Training [Internet]. DHIS2 TOT Training. <http://hispmalawi.org.mw/index.php?page=pages&pid=39>. Accessed 7 Nov 2016
30. University of Malawi: Computer Science Department holds DHIS2 Workshop - University of Malawi|Chancellor College [Internet]. Computer Science Department holds DHIS2 Workshop (2016). <http://cc.ac.mw/news/computer-science-department-holds-dhis2-workshop-17-03-2016>
31. Kimaro, H.C., Nhampossa, J.L.: Analyzing the problem of unsustainable health information systems in less-developed economies: case studies from Tanzania and Mozambique. *Inf. Technol. Dev.* **11**, 273–298 (2005)
32. Sahay, S., Walsham, G.: Scaling of health information systems in India: challenges and approaches. *Inf. Technol. Dev.* **12**, 185–200 (2006)