



CHALLENGING PARADIGMS IN HEALTH INFORMATICS



By Vivek Singh

A software professional working in a mature software development organisation, with experience in the health sector for the last 5 of 17 years.

Summary

In this paper, we discuss e-health in the context of developing countries. We acknowledge that we ourselves have made many of the mistakes that we highlight, and our opinion is based on our learning from these experiences. We make two broad points relating to the development and implementation of Health Information Systems (HIS):

- We argue that we should focus on developing HIS that empower healthcare providers directly, so that our client organisations benefit from them. We prefer this over centralised data collection systems that solely seek information from providers in the field. By doing so, we can ensure that centralised data collection happens automatically and at better quality.
- Software is a small but integral part of the total information ecosystem. We believe that software partners will be relevant and yield results only when they engage with healthcare organisations and providers in the field where the service is delivered, instead of participating in the process as utility vendors.



Observations

In this paper, we will not delve into too much detail about the general challenges of developing and implementing HIS. We believe that much has been written about this. Instead, we will only describe a few less commonly cited systemic patterns that we have observed when it comes to the development and implementation of HIS for Governments and NGOs in particular.

- We have experienced that technology partners are brought to the table at later stages of e-health projects, as compared to other professionals like statisticians or management consultants. Moreover, the involvement of these technology partners typically ends much sooner in the project lifecycle. Cost is usually cited as the reason because IT vendors are not inexpensive, but the prevailing sentiment also appears to be that minimal vendor involvement is appropriate.

ThoughtWorks® Global Health

This has a detrimental effect on these projects, because in most cases, user needs only start manifesting when users receive the first version of the HIS. It is important to clarify that users only start taking the system seriously when they depend on the system — i.e., not when it is being piloted or is a mirror system. This happens when most of the planned effort is coming to an end, or as we on the software development side say, when the software enters the maintenance phase.

- If software development partners have been expected to participate merely as ‘utility’ vendors in projects, then they have mostly obliged. In such engagements, the vendors simply throw software over the wall* to their clients and users, leading to poor results.
- HIS depend on health care providers to input information. However, this usually ends up being an additional process, which is over and above — and often disconnected from — their daily work. Simply put, it is viewed as an additional job to do. The providers themselves perceive no benefit from doing “data entry,” but end up doing it owing to pressure from their supervisors. The information needs of top management are different from the information needs at more junior levels of the organisation. This problem is easiest to identify when we notice that the user only feeds in the information, but does not use it otherwise. The basic question of “what is in it for the user?” is never explicitly asked during the lifetime of the project.
- We have observed that Open Source Software (OSS**), driven by community and individuals, has worked well in the technological domain. In fact, it has worked even better in the creation of software development tools. In comparison, it has not been as successful in non-software domains like Public Health. One reason for this is that such domains are not as easily understood by the developer community — they simply do not feature that often in the life of the typical software professional. Hence, it is hard for a developer sitting at home to develop an OSS-based solution for Public Health. As for implementing such systems, developers would need to do so very carefully — usually over weeks and even months — as compared to the mere seconds or minutes needed to download and use OSS tools in the technological domain.



ThoughtWorks® Global Health

- Creating solutions that are taken off the Internet or from managed data centres, which serve affluent consumers and IT literate enterprise users, are not useful in low-resource settings. Most IT professionals themselves are consumers or enterprise users at their workplaces; hence, the requisite empathy required for the Public Health domain remains largely missing.
- Sadly, the very business of software development perpetuates this proximity bias — it pays best to serve consumers followed by business enterprises, and finally, the Government & NGOs. Reality is that the most talented IT professionals are attracted by projects that pay them better.



Proposition

We have worked on a wide range of e-health projects over the past five years. We follow these simple propositions, based on our experience in the consumer, enterprise and (now) Government and NGO sectors.

- Any project that depends significantly on IT for its success, should have IT professionals involved earlier in the project cycle and continue on until much later. This would provide valuable context to the technology partners and allow them to choose the right solution fit for the purpose. This would also be useful for the other domain professionals involved in such projects; rather than presenting their idea of a solution without a technology partner, it would be better to jointly state their problems, examine them, and develop iterative solutions. For instance, rather than saying “we need to solve a certain problem with human resource management,” they might end up stating that “we need a human resource information system”. The second statement is not really a requirement, but a solution in itself.
- HIS are most successful when they are integrated into one’s work, instead of becoming an additional work-stream of their own. Designing such systems requires working alongside providers and local program managers, understanding their needs and helping them uncover new possibilities. Admittedly, creating such designs and training each user to incorporate them into their work-flow is a slower process, but it leads to higher quality results. For one, it creates better ownership on the part of the user and, as a result, ensures better quality of feedback for the

ThoughtWorks® Global Health

system under use. If pursued over time, this can create a virtual cycle — one of highly productive providers and corresponding qualitative solutions. This is a much better situation for the client organisation to be in, instead of having to rely on enforcement. Ultimately, the information needs of top management should be predicated on enabling providers to do their work better.

- Effective design comes from empathy. Empathy is created by direct human experience — whether it has to do with the very purpose of the system, the setting in which it is used, or the situation of the users or providers themselves. A typical IT organisation’s location and work environment does not provide the opportunity to develop this empathy, especially for domains like Public Health. This empathy can be developed only when technologists venture out of their comfortable workspaces into service delivery environments. By this, we do not mean occasional site visits, as we have often seen, but a deeper commitment to building empathy by working alongside health care providers in their various work settings.
- We have observed that there is a shortage of IT skills deployed in the Public Health space. Anyone who works in this domain will attest to the fact that the use of the phrase “a small world” is quite common. We have pointed out the underlying commercial explanation for this, and we do not expect this situation to change. Our hope is that those IT organisations that can afford to, should consider contributing some of their available IT skills as part of their Corporate Social Responsibility efforts, as skills are more scarce than funds.
- When requirements are dynamic, and software development has to be responsive to experience gained during the process of implementation, two features become mandatory: the software must be Open Source and it must be Agile***. Bahmni is a software product that we have developed using both methodologies.
 - With OSS, you can avoid getting tied down to a single vendor who might not be able to partner over the long-term. We also advocate the use of complete OSS and not just for layers of software that are developed in that particular project, so that this principle applies more widely, and in the process, creates greater public utility.



ThoughtWorks® Global Health

- Agile encourages participants to embrace change and not view it as failure to predict the future accurately. Agile teams embrace change instead of focusing on surfacing the root causes of the changes needed. In a dynamic world, pre-analysing the outcome is extremely hard. Agile values being effective over being efficient. One can be efficient only when one possesses great knowledge of the problem as well as the solution, which happens but rarely in real-world situations.

* “Throwing deliverables over the wall” is a common problem in the Waterfall methodology of software development. The metaphor speaks of an idea where the receiver of a deliverable has no access to the deliverer, and cheap mechanisms are used to provide feedback on the quality and/or appropriateness of the delivery.

** Open source - In this software, the code is available for modification or enhancement by anyone. Its authors make this available to others who would like to view that code, copy it, learn from it, alter it, or share it. They can do so as long as they let others do the same when they share their work.

*** Agile software development encourages providing feedback and makes processes iterative between the gathering of requirements, analysis, development, testing, and operation of the software. It encourages teams to break down the metaphoric walls described for Waterfall software development.

ThoughtWorks® Global Health

ThoughtWorks Global Health works with organisations that are advancing the cause of equity in health, in the social and government sectors and operating at all levels of the health system. A dedicated team of physicians, public health specialists, and technologists supports these clients through 34 offices worldwide.

Our focus is on improving quality and expanding access to care in low-resource settings. ThoughtWorks Global Health delivers technology solutions, tools, consulting and open source expertise to help clients empower workers in the field, deliver higher quality facility-based care, and leverage system scale to improve outcomes.

ThoughtWorks is a global technology company and a community of passionate, purpose-led individuals. Our teams think disruptively to deliver empowering technology that addresses clients' toughest challenges, all while seeking to revolutionise the IT industry and create positive social change.

thoughtworks.com/global-health