Software Principles for the Public Sector with focus on Public Education

February 1-2, 2010, Bengaluru

Guiding Principles for Public Software

IT for Change
Bengaluru

Table of Contents

UNESCO, e-Governance Department, Department of Public Instruction and Sarva Shiksha Abhiyaan (Government of Karnataka), Karnataka Jnana Aayoga (Karnataka Knowledge Commission) and IT for Change (ITfC), organised a workshop “Software Principles for the Public Sector, with focus on Public Education” in Bangalore on February 1 and 2, 2010. This note is a key output from the workshop discussions.

For a digital version of this note as well as to access the workshop presentations, case studies, concept note etc, please visit www.Public-Sofware.in

Guiding principles for software for the Public Sector

1. There are certain political and philosophical imperatives of the public sector that any software developed for its use, needs to be in alignment with, in terms of its own architecture/ philosophy. These imperatives could be termed as 'Guiding principles for policy on software for the Public Sector', or 'Guiding Principles for Public Software' and form the subject matter of this note.

2. Definition of terms:
    1. Public software can be defined as software developed for the public good, which is publicly owned. Public ownership also implies that it cannot be privatized or privately owned. Public software is of two kinds:
        1. Public Software developed to promote public good - that helps government (public sector) to fulfill goals of government - software that supports NREGA transactions in a transparent manner
        2. Public software itself as a public good - new digital goods such as Wikipedia where the software itself performs public interest functions
    2. Public Sector for the purpose of these guidelines, is defined broadly as comprising of institutions working for public interest. While this includes not only governments but also academic institutions, civil society (NGO/CBO), community media institutions etc, there is a special role for the government as the key public sector actor in promoting public software.

Software and the public sector

3. Software has often been seen as a value neutral 'technology' issue and its procurement akin to that of purchasing furniture or such office or infrastructural item. However we are living in an increasingly digital world and the role of software, which is at the core of new ICTs, is becoming critical in our lives. Whether it is in using search engines, accessing public services over the internet or through tele-centers, communicating with colleagues and friends, participating in virtual professional and social networks, banking, use of the net for by political parties and NGOs for campaigns; computers and the internet have become critical to our lives.

4. Software is a basic building block of our increasingly digital world and its nature has important implications for public interest. Its procurement thus has important implications far beyond that of procuring office equipment etc.

5. In the case of the public sector, which has clear principles that underlie its own work, the choice of software needs to be aligned with these principles. Public Sector guiding principles include:

    1. Working towards ensuring universal access to basic facilities and resources such as public education, public health provisions, employment nets etc, to address structural issues of poverty, illiteracy, malnutrition / ill health, unemployment etc. With the goal of providing access to water resources, governments design policies and programs to create and maintain with public funding, public water bodies. Likewise governments set up schools and primary health centres across geographies, and especially in remote areas and especially to marginalized groups in the interest of social justice and equity. In these cases – water, health education services constitute 'public goods' which is essential to all but does not provide provide sufficient incentive for markets to provide[1] and hence becomes a public sector responsibility to ensure universal provision.

    2. Promote a culture of transparency, as well as public participation in its working, with accountability for its work and decisions to larger society. Structures such as Community water user groups or School Development Committees, processes of social audits of government programs or the increasing use of RTI is a characteristic of participation and transparency needs of public systems.

Public Sector Principles[2] and implications for software architecture

6. ICTs are also beginning to play an important role in the information and communication processes within governance systems, and e- governance is acknowledged as having the potential to transform governance through greater transparency, accountability and participation.

Software developed for public service, and especially in government, has a unique context and objectives deriving from those of public service; with its imperative of providing public goods and ensuring equity and social justice. It is well known that private and commercial actions have very different context, motives and considerations than public actions. For instance, the largest possible reach and diffusion as well as transparency of actions are basic to public service, which are not necessarily values espoused by private and commercial players.

Generic issues faced in the public software arena

7. There has largely been a complete lack of distinction between the software needs of the private and public

1 In each of these areas, private providers exist, however, being market driven, they would clearly not be able to ensure universal access which is the cornerstone of government vision and policy.

2 UNESCO, e-Governance Department, Department of Public Instruction and Sarva Shiksha Abhiyaan (Government of Karnataka), Karnataka Jnana Aayoga (Karnataka Knowledge Commission) and IT for Change (ITfC), organised a workshop “Software Principles for the Public Sector, with focus on Public Education” recently. This note is a key output from the workshop discussions.

sectors, and the private sector software principles and methods have largely been adopted in the public sector as well, with its attendant drawbacks. In addition, the issues faced in attempting to implement public software include

1. Proliferation of proprietary software in some domains
2. Lack of readily available solutions conforming to public software requirements (also lack of awareness of existing applications that do)
3. Lack of adequate commitment to Open Source solutions
4. There is no common database on Public Softwares and also there is often a lack of willingness to share public software applications and content. There are few or no mechanisms for providing suggestions for software improvement / updation
5. Lack of support and maintenance infrastructure in the public sector arena
6. Testing, Maintenance and support (maintenance) aspects are traditionally neglected in the public sector and in the case of software, which continuously evolves to meet current and future requirements, treating a software as available in its current state for posterity (as furniture would) can create huge difficulties.
7. The private sector technology companies largely promote licensed and priced softwares
8. Software developing agencies do not adhere to established standards

8. Based on these underlying principles of Public sector and their implications for Public Software, a set of principles that can guide decision makers in the procurement, development and use of Public Software, and which can help address some of the identified challenges can be drawn up covering following areas

Guidelines for public software

Ownership and sharing (nature of social contract)

9. Anything developed using public money should be available widely to everyone for furthering wider public interest. This is even truer of digital products where the marginal cost of replication is nearly zero. It is also important that all such digital products – software, applications and content – are widely available to citizens to use – including through modifying – for various useful societal purposes.

10. If the source code is owned by a private vendor, dependence on this vendor for support and maintenance can be both expensive and dangerous. This 'vendor lock-in' can create dangerous dependencies of public system on a private vendor. If the vendor keeps increasing the price of the software and associated services, the public system will have little option but to pay. Switching to a new system would impose huge and duplicate costs. If the vendor shuts shop, the application will run the risk of becoming non-operational, since software is usually perennially in need of being upgraded, either to fix defects or to add new functionality arising from new requirements and new environments.

11. This means that the source code for any software used by the government cannot be proprietary. In case of desktop/client software, this guideline is easy to follow, since there are several client based software applications for operating system, document processing, video/image editing etc that can be used by the public sector. Already many states in India are preferring public software applications on the desktops and this needs to be mandated by policy.

12. In case of larger applications which require back-end application development, it is important that the software be purchased outright at the time of its development by the vendor. In most cases, the person making the RFP does not specify the ownership of the application and there is a danger that though public monies are paid to the vendor, the ownership remains with the private vendor. This means that the public sector entity does not have the right to make changes or corrections to the applications[3]

3 The case of the AP NREGA is important to note. Though most of the knowledge for building this application is that of the government officials in the RPDR who provided the complex knowledge of the functioning of the government department and the scheme, but the ownership is the with private software vendor who provided the software development knowledge which is much more easily available in India

13. There may be cases where the software on an outright purchase basis may seem too expensive. In most cases such applications are likely to be usable across many state governments. In such a case a larger contract with multiple states can be considered, in which the application can be multi-lingual interface to begin with.

14. To this end, it must therefore be ensured that:
    1. The public agency commissioning a software take up complete ownership with full rights on it. This ownership should be real and not merely nominal providing the concerned public agency with full ability to share and modify the software as required.
    2. These rights should be shared with public to enable further development of software and applications for different societal uses, including through collaborative work.

15. Specific operational issues:
    1. There needs be an agency, which could be part of the state eGovernance department or DPAR if there is no eGovernance department at the state level and at the central government level, which needs to take custody of the software developed and plan for its maintenance, either in-house or through vendors. This will create 'internal ownership' of the application and help create 'substantial ownership' and not just 'token ownership'.
    2. The code should be made available on a GPL kind of licensing which permits software enterprises to be able to modify and enhance the applications and release it again under GPL, which would further public welfare. This will also allow the government department having custody of the application to work with a number of FOSS enterprises to maintain and enhance the application.
    3. In exceptional cases, when proprietary application (closed code, royalty based) is required to be procurred, the bid process should provide clear and adequate justification for the same and this would also need to be verified as a part of the audit checklist/processes.

Ensuring highest public interest and realization of citizen's right (purpose)

16. Public or governance software and applications are nothing but a new set of governance processes and systems. These should be developed with general larger canons of public service in mind, as they serve specific purposes of imperative and action etc of a particular agency developing the software of application. Very often new digital possibilities enable higher realization of public interest and citizen's rights than was possible in pre-digital processes. This is especially true with regard to transparency, accountability and monitoring of governance processes and active ongoing participation, including through co-creation and collaboration. It is important that all software and applications specifically take note of these imperatives of governance reform in India and include all possible features that enable them. There should a clear process of ensuring that all such new possibilities are incorporated in any software and application that is developed as one of its central design principles, and any exception is duly noted with clear reasons thereof.

17. Universal access to government services requires universal access to software required to access those services, specially as more and more such services would be provided through digital processes.

18. The concept of social audit of government programs/processes is seen as a critical component of transparency and is already being implemented[4] things like socio-political pre-audit, inviting comments from the community,

Specific operational issues:

19. Typically, governments invite comments/participation of the public in any new measures they undertake, which can be in terms of accepting written representations, public hearings etc. In case of software applications (which in a sense implements government policy), too, such public processes are required to ensure that the requirements of the community are also catered to in the design. This will help applications

4 Mandated in the NREGP

to not just incorporate 'Management Information System' but also 'Public Information System' or 'Janatha Information System'. Such public processes must be a part of the user requirement phase and documented in an appropriate manner. In addition, representatives of the public should be invited to beta test applications to ensure that important community needs have been incorporated.

Core technology issues (methods)

20. The government is usually lays down standards that are used in all areas of life – weights and measures, distances, ISI/Agmark standards etc. These standards are publicly owned and free for use by all in society. In the case of Information Technology arena, since it has witnessed rapid changes in the evolution of different applications that even change the nature of the application area, such standards have not been similarly created/adopted. In many cases, the default standards are proprietary and citizens need to pay royalties for use of these standards, which is inherently against public interest, and especially iniquitous. As technologies mature (as for instance in the area of text documents), there is a need to establish and use established open standards (eg the Open Document Format which is a ISO standard endorsed by Indian government) and avoid proprietary standards, which are also usually royalty extracting.

21. Open Architecture is a type of computer architecture or software architecture that allows adding, upgrading and swapping components which needs to be mandated. The eGovernance function should actively pursue standards in different technology areas, along with specialist agencies like NIC. Open standards would certainly be required in areas such as fonts, document formats, web accessibility, such that citizen have least difficulty and incur no or least cost in accessing public information and in participating in its audit/creation as applicable.

22. As a part of public software, government should create, through in house or outsourced efforts, large number of free and open fonts that are compliant with the current standards (UNICODE). Since India has several scripts, the use of free and open fonts is essential to encourage the local language computing. However, there are cases, where proprietary fonts are popular since free and open fonts are not available or not easily available or are not good. Since these fonts are clearly a case of public good (required by all, but may not provide incentive to any one individual or institution to create it and make it freely available to all), the government must create and ensure free and open fonts compliant with standards[5]

23. In any procurement of software or hardware, government must insist that the vendor should provide the required drivers etc to ensure compatibility with the public software already being used

24. Wherever the technology area is still evolving, it may be difficult to adopt a standard. Even in these cases, where specific applications become more and more commonly used/popular, government needs to actively pursue the creation of public standards. Also, there are very important areas such as GIS (Geographical Information Systems) which will become critical in government work and it is important for the government to actively work to create open standards in this area

Specific operational issues:

25. Adherence to standards by the developing agencies such as W3C guidelines on usability, accessibility and SCORM kind of standards for content development should form part of the checklist for any software development or procurement project.

Software development process

26. The arena of ICTs has several distinct components – provision of hardware, hardware maintenance, software development, software implementation, software maintenance and support etc. Different aspects may need to be treated differently. While hardware purchase and maintenance is usually done on a tender/bid mode,

5 UNICODE is the current global standard for fonts

software development and implementation need not be combined with the same. There are advantages in in-sourcing the core area of software making, and NIC for instance has developed some applications widely used across states. Where the government outsources software development, this should be done in a planned manner, in most cases, the domain expertise required for the software needs to be identified and made available for the project, the relationship between the domain group and the software development agency needs to be made clear etc.

27. There is a need to check if similar software already developed which can be customized and used. Not only the issue of cost of development but the cost of perennial maintenance. Re-using the code/components means support efforts are also shared which is the big saving.

28. For applications that need to be developed since there are no existing applications that can be used/adapted/customized, modular design and development is essential, even more so for public sector applications. While this may create an overhead in the design and development aspect, it makes the maintenance, reuse and enhancement aspects easier and allows the government to not be locked into the vendor who developed the application for its support and enhancement.

29. Community input in the specification and design processes and community audit during the testing processes are essential. The term community has two components here – one is the Free and Open Source Community that has thousands of volunteers who can participate in the design, development and testing of software applications. The second is the larger community who are also the stakeholders in the use of these applications.

30. Multi-lingual capabilities to be necessary factor in design (even if not developed in multiple languages, any application should have a design that would allow new languages to be easily incorporated through simple modification of the text messages used in the user interface.

Specific operational issues:

31. The budgetary provisioning for software procurement/development needs to factor in the entire software development/use life cycle, catering to continuing maintenance/enhancement over a longer period of time.

32. Modifications to software also need to be meticulously documented to ensure continued support.

Education

33. The issue of public ownership over the software application is even more critical in case of the education system. Just as articles in text books publicly owned, permitting the public education system to make available millions of copies to schools year after year, digital learning material (in terms of both static or dynamic 'content' and software tools) too needs to be publicly owned. No digital learning material should be procured by the education system on a royalty basis, since this would entail huge outflows from the system which would be avoided by public ownership. Of course the public ownership may be created by an outright purchase for which money may needed to be paid to the producer. However there should be no reason for procuring on royalty mode

34. The processes of creating digital learning material, need to conform to the same stringent quality control processes, that text books are subject to. This is necessary to ensure that the learning material promotes educational aims and is in line with accepted educational policies and perspectives and does not subvert these. Similarly digital learning material too needs to be produced under similar safeguards and quality assurance measures. Practices of procuring digital learning material from vendors without authenticating the same through the processes similar to text book creation is a violation of curricular design and development policies and should be stopped.

35. The digital learning material itself should be made available by the government on a free sharing licensing

model such as creative commons. Digital learning material is amenable to changes/enhancement. This will allow teachers and teacher educators to create locally relevant, custom versions of the material, which is one of the key educational policy goals (decentralized curricular design/development), which has so far not happened, but which ICTs can facilitate by supporting far easier processes of sharing and customizing and at much lower costs that text based material would allow.

36. Government must encourage the processes of local curriculum design and development through the DIET-BRC-CRC-School structures which would be a significant step in itself for teacher professional development.

37. The syllabus should avoid as far as possible mentioning specific brands or application names, since learning is of concepts and principles, and not of specific applications. Proprietary applications should not be mentioned in syllabi.

Specific operational issues:

38. Even more than in other areas, there is enormous pressures from vendors on the education department to purchase (outright or on royalty basis) their resources – either software tools or digital material. 'Conference' kind of events are regularly organized to 'showcase' products of vendors to government officials. The processes of curriculum design/development need to shield the officials from these huge pressures.

Other issues

39. This section looks at some basic requirements of public software
    1. Audit trail required of any public sector application from accountability perspective. For instance, no record, once authorized should be allowed to be deleted since that would compromise accountability, instead system should require additional transactions to modify a recorded one, with its own audit trail
    2. Security of public data – data backups to be facilitated in the software, this is essential to truly digitize the work of the government
    3. Investment is required in capacity building of public sector officials – lack of awareness and fear/inhibitions block the adoption of public software. In most cases, those afraid of adopting public software have never used it, hence it is largely an apprehension without basis, which needs adequate capacity building / orientation processes to address
    4. Associated policy changes are also required to require officials to use computers and public software. Kerala already has a policy requiring FOSS to be preferred and Tamil Nadu is currently preparing a similar policy.

Role of specialist agencies (structure)

40. There are quite a few public sector agencies in this area, including NIC, NRCFOSS, CDAC. There is a need for close coordination amongst agencies, while they are doing excellent work, there is need to work on following aspects.

41. There is a need to establish an agency within DPAR for coordinating the activities of agencies working in different aspects of public software – research and development (CDAC), design and development (NIC), support and maintenance (NIC), governance process design (NISG) etc. This agency could also setup and maintain the National repository for public software.

42. Government needs to create an empaneled list of vendors who are competent in using free and open platforms for software design and development

43. Government needs to ensure that vendors have no conflict of interest possibilities. In many cases, the same vendor is in both the consulting as well as in the service provision spaces and using the same vendor for both purposes is clearly a conflict of interest issue.

44. PPP is often used in software projects. These agreements should be subject to the guidelines discussed in this note – the ownership over the source code should not be with the private vendor. The risks of vendor lock-in need to be carefully studied and documented to ensure that the PPP term does not create dependencies which can harm the public interest

Specific operational issues:

45. The role of the agencies in different activities – software design and development, implementation, support and maintenance, training and documentation need to be clearly spelt out

46. This area is a 'new work process' one in which traditional bureaucratic approaches are not often succesful. There is a need to allow agencies working in this space additional autonomy to support greater flexibilities of operation as well as enable them to provide support services at high levels of quality

Overall aspects of public software and its importance

Re-engineering government processes

47. Digital processes enable completely new levels of work integration and thus outcome effectiveness. Every time a software and application is developed for governance processes there should be a clear process of identifying other agencies working in the same or related areas; the nature of such connection and possible overlaps; possibilities of sharing, collaborating, synergies etc; and common resources and possibilities of action, like databases, citizen interface etc. Even if not possible because of current government structuring to leverage all these possibilities right away, a mapping and thorough analysis of these issues, and connecting with the concerned agencies at the state of software/ application design and development will have immense advantages – both current, and in the future. Such mapping process would include
    1. identifying all other agencies with overlapping and interfacing remits and areas of work – including non-govermental ones
    2. checking for existing applications, common databases, common information formats that can be used by the department for the current requirement. Even if reuse of the code is not possible, the requirements, design specifications definitely need to be studied and used as relevant.
    3. Convergence of applications from overall functionality perspective needs to be seen as a clear component of Government process re-engineering. Currently smart card applications are being made by many departments at the same time for the same citizen, which is a waste of public resources and creates potential issues of duplication of records etc, apart from inconveniencing the citizen

48. Ensuring better and more effective working across government agencies, and with outside partners. while the previous section takes into account the best realization of all new possibilities enabled by digital processes for citizen's right, this section pertains to such possibilities with regard to internal working of the government agencies with the clear and single/ convergent objective of serving citizen's needs.

Strengthen the front line government worker

49. Systems should empower the front line staff. The dilemma has been to provide adequate autonomy to the front line officials while keeping them in line with policy requirements. In absence of regular interaction possibilities, the method has been to usually severely curtail autonomy and flexibility of the front line personnel and make them largely implementers within rigid boundaries. However with new ICTs, networks are possible to be setup which can give far more leeway to the front line staff while alerting any significant deviations from policy. This liberation of the operational staff from micro management while still keeping them within overall norms is what has allowed the growth of transnationals, who can allow significant local

flexibilities at local levels but retain overall control from their head offices

Universal access to ICTs

50. Promotion of all ICT P&S in Indian Languages. Government should actively procure software and related tools that can help citizens and communities to work in local languages

51. Government should actively work with both government agencies and private sector companies to introduce the low cost computers, containing all possibly useful public software to the public

Annexure A. Public Sector Principles and Software architecture

Universal access

52. The universal access to ICTs and the new possibilities they offer has been an important goal of the government. The CSC scheme aims to establish tele-centres all over the country to enable access to computers and internet to all citizens. Different government departments are also computerizing their operations so that the information on their work can be available in digital formats on an anywhere, anytime basis.

53. Software that is proprietary (privately owned) where the right to share and modify is not publicly available are a serious obstacle to universal access to ICTs. Specially software that runs the personal computers used by all (desktop software), comprising of operating system, applications that process text documents, spreadsheets, images, audio/visuals etc.

54. Proprietary software also often uses document formats that are proprietary. To open or edit these documents in proprietary formats, the user is compelled to procure a copy of the same proprietary software, at a cost of the license fees.

55. To make ICTs universally accessible, it is essential that the public sector only uses documents in free and open formats, so that no citizen is compelled to purchase software that can open or edit these formats. Similarly, the software used should also be one that is freely shareable, so that the citizen has an option to use the same software that is used within the public sector. The Public sector should avoid using applications that cannot be freely acquired by the citizens too.

56. The principle of equity/social justice requires that the Public sector should specially focus on the needs of the poor and marginalized. Since any 'license fee' restriction can be a severe impediment to the possibilities of the poor and marginalized to use software, equity consideration requires that the public sector uses only freely shareable software.

Transparency

57. The public sector is committed to far higher levels of transparency than the private sector. The RTI applies to public sector and requires that unless there are reasons to hold back, all information relating to its work would need to be shared when asked for

58. On similar grounds of transparency, the source code used in all public sector applications needs to be transparent to the public (unless there are valid reasons for non disclosure akin to the RTI, like national security etc). Commercial reasons for non transparency are not applicable to the public sector. In a way, this transparency is essential to also permit social audit of the software used in public sector applications.

59. The lack of transparency of source code may seem a trivial issue. However, there are several cases, where applications that are closed have had malicious code inserted which carried out illegal or criminal activities for the application developer. The case where World Bank (a public sector institution) blacklisted software vendors on the grounds that the software supplied by them had code that carried a surveillance on their activities

60. Closed code has danger at both ends – that the vendor can introduce malicious code that affects public interest or that government can carry out (without following due course of law) surveillance on citizens using closed code applications.

Participation

61. Proprietary applications allow only for their application to be used. Software released under the GPL can be modified by users as well as freely shared with others. This increased extent of participation has following implications

62. Public software can be customized with changes designed, developed and tested with wider community participation. This helps in greater ownership over the software application and the processes surrounding its development and use. This can give a fillip to groups of volunteers who can participate in development, testing, audit etc. In ICT programs in public schools, teachers and students can customize the software and tools for their local needs[6]

63. Also small enterprises that can make locally relevant customizations and provide these at a price can also be supported.

64. Typically, we see the ICT field dominated by large vendors, many of who are monopolies and the encouragement to small enterprises can bring about greater public participation in the technology sector, as entrepreneurs who can create local jobs and services. This extends participation in software creation activities.

Security

65. Certain government information requires to be kept confidential, such as those relating to national security. Software that is used to process or store information or transactions pertaining to such issues needs to fulfill certain specific needs. The source code of such applications need to be available with the government, since there is always a danger that malicious code could be introduced into the application that can compromise public interest. Hence government cannot use code provided by vendor in a closed form.
66. Similarly on security grounds, there may be a need for source code not to be publicly available. Governments may prohibit the availability of source code for applications developed to crack encryption software. This may be one of the rare cases where public software would not be free and open source software

6 As is being done in Kerala under the IT@Schools program