Transactions on Information and Communications Technologies vol 8, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Using the capability maturity model as a guide for transferring process technology to government organizations in Sub-Saharan Africa

W. Junk[a] & M. Parish[b]

[a] *Computer Science Department, University of Idaho, Moscow, ID, 83844 USA*

[b] *Thunder and Associates, P.O. Box 40874, Nairobi, Kenya*

ABSTRACT

In the late 1980s the Software Engineering Institute published the Capability Maturity Model (CMM) for Software. We have successfully used the CMM framework to guide the transfer of software development and quality assurance process capabilities to the Microcomputer Information Systems Department in the Ministry of Finance, Government of Kenya. In this paper we describe a three pronged technology transfer program integrating instruction, standards development, and guided practice which used the CMM to help identify where improvements were needed. We also present the results of a case study in which a software development life cycle was strictly applied, documentation developed, and quality assurance reviews performed on a real system under development for the Ministry.

INTRODUCTION

In the mid-1980's, the Ministry of Finance (MoF), Republic of Kenya undertook the modernization of their financial accounting and budgeting system. The existing applications were mainframe systems written in COBOL, running in a UNIX environment, and were supported by a staff of experienced COBOL programmers. However, as has been the case around the world, the resources of the mainframe operation and its ability to respond were not keeping up with the users' requirements.

The MoF realized it had an opportunity to take advantage of the new microcomputer technology. Due to its ease of support, durability, and

simple nature, microcomputer technology was believed to be uniquely suited to the needs of Lesser Developed Countries (LDCs). The enthusiasm of the government in this undertaking was mirrored by donor countries that provided funding for the purchase of hardware, software, training, and consulting services. However, the managers and technicians of the mainframe operation significantly resisted this initiative. As a result, a new organization, Microcomputer Information Systems Department (MISD), separate from the mainframe operation, was formed.

To head the new organization a Director was selected from within the civil service ranks. Then approximately 45 technicians were recruited and hired from outside of the civil service. Because of the lower pay scales for civil servants, the government was not able to hire from the private sector. The microcomputer technology they were undertaking was leading edge and there was not a large experience base even within the private sector. The Ministry then looked to the University of Nairobi for candidates. Many new technicians were recruited from the mathematics, engineering, and science disciplines. Consequently, most had no practical experience in system development.

The applications to be developed for the Ministry and the hardware and software environments available required a broad range of skills and experiences. Applications covered accounting, budgeting, planning, tracking, and revenue accounting. The operating system environments included DOS and UNIX, with some distributed processing on Novell networks. Systems were developed initially using dBase and then FoxPro and Clipper. MISD staff were also responsible for ongoing systems maintenance and user support.

From its inception, MISD was charged with establishing standards and guidelines for systems development, reviewing projects to ensure that standards were used, and developing systems to meet the requirements of the Ministry. However, there were no standards which could be immediately adopted for the development environment in which MISD staff would be operating.

STAFF AND ORGANIZATIONAL DEVELOPMENT ACTIVITIES

Immediately, MISD was faced with the need to establish itself, increase the technical skills of its staff, and develop systems for its eager users, all at the same time. To build the Ministry's confidence, MISD management established quick response to user requests as a high priority. A consultant group was brought in and co-located with the government staff to provide the higher skill levels needed to train government programmer/analysts and lead them in the high priority systems development.

A major training program was undertaken for government programmer/analysts. They received classroom training in a broad range of technical areas including dBase, FoxPro, and Clipper, as well as the DOS and UNIX operating systems. With this training and the assistance of the technical advisors, MISD experienced strong initial successes that established its credibility. Over the next few years, MISD and the technical advisors delivered more than 20 applications enjoying a high level of user satisfaction.

BARRIERS TO FURTHER SYSTEM DEVELOPMENT

After several years of experience in developing new systems as well as maintaining and enhancing existing systems, MISD Management began to notice that in many cases systems were understood by only the original developer and not easily sustained by another staff member. Systems were generally not well documented. The success or failure of development efforts seemed to be totally dependent on the skill and experience of the individual developer. Projects were more successful when the scope and delivery schedule permitted development by a single individual, rather than a team.

As the user organizations became more experienced, their appetite for more complex systems grew. In an attempt to respond to some requests, the government contracted with a major international developer to deliver automated solutions for the Income Tax and Value Added Tax Departments. Many difficulties were encountered during the contractual effort. MISD recognized the need to improve its own system development capabilities so that it would be able to participate in and support the development of these more complex systems. The Director recognized the need to expand programmer/analyst focus beyond individual technical skills. He felt that an examination of the processes used to develop systems would be a key to improving both capability and quality. He felt that a quality assurance program was needed to improve the capacity of the organization and he asked for assistance from the on-site consulting group.

TECHNOLOGY TRANSFER MECHANISMS

Before discuss our approach to improving the effectiveness of information technology transfer we want to briefly review some approaches that are commonly used by LDCs to upgrade their technology base. Although each situation has unique concerns, there are three distinct avenues for technology acquisition: (1) receive donations of, or contract for turnkey systems which will be quickly integrated into governmental operations, (2) have staff attend classroom instruction after which they return and apply the new technology in their work, and (3) co-locate expatriate advisors

with the government staff to advise government managers, help guide the organization in long term planning, provide technical direction, and serve as a technical resource in day-to-day operations. Each of these approaches has the potential for accomplishing technology transfer, but they are not universally or equally beneficial in all situations.

Turnkey Delivery of Systems

This method is characterized by the acquisition of technology in its completed form (all hardware, software, and documentation), fully developed and ready for implementation in the user's environment. Because of the complexities associated with the application, most turnkey systems require customized development by an outside organization experienced in system development. Typically, the developers are expatriates working in the country, but not in daily contact with government staff. In some cases development is performed outside of the country and delivered when completed. During system installation and implementation, government staff are trained to acquaint them with the system's hardware and software architecture, languages used, and operational procedures. After a short period of developer support, the government staff are then expected to assume operations, maintenance, and enhancement responsibilities.

Classroom Training

Classroom instruction can expose a large number of people to a specific technology. It is most effective for relatively simple technologies and fundamental concepts such as a new programming language or the use of a new software development tool. Laboratory or hands-on exercises enhance the technology transfer by providing concrete illustrations and opportunities for participants to perform structured practice in a situation that poses little personal risk and no operational risk to the government organization.

Teams of Expatriate Advisors and Government Staff

Another common approach to accomplishing technology transfer is to co-locate teams of expatriate advisors with members of the government staff. In this situation the advisors are not members of the government organization, but rather are individual consultants or staff members of private firms under contract to provide management and technical support. The expectations of LDC governments and the donor organizations that fund projects have been that the combination of appropriate technology with the use of expatriate "experts" to introduce the technology would result in a rapid proliferation of information technology systems.

In the context of information technology systems, the government personnel and consultant work in a team environment, usually co-located, to complete the analysis, design, construction, and implementation of the

information system. The role of the consultant is to provide direction, technical expertise, and training while actively participating in the project. At the conclusion of the development effort, when the consultant is no longer available, the government staff are expected to function independently to maintain and enhance the system.

Realities of Technology Transfer

We have seen situations were combinations of turnkey development, training, and teaming have been successful in the short run, allowing consultants and government staff to develop information technology systems. However, their success often depended on the heroic efforts of individual team members. The organization as a whole did not assimilated and institutionalized the technology. When a key staff member left the organization, a void was created and the technology transfer had to be started over. Consequently, it was difficult to maintain continuity and it was improbable that the organization would be able to sustain a long term, continued technology growth. It was becoming trapped in repeated cycles of learning and relearning basic skills.

Contractor developed turnkey systems present unique problems for the government. Representatives from MISD and user organizations are asked routinely to review and oversee the contractor's work. But, frequently the government staff members do not have the background to achieve effective management of the contractor's activities. In the end, a system may be delivered, but it is likely to fall short of the desired capabilities. It is also likely that the government staff charged with maintaining and supporting this system after the contractor's obligations are completed, will experience significant difficulty due to inadequate knowledge of the systems architecture and implementation details.

THE CAPABILITY MATURITY MODEL

After having seen the failure of many varied technology transfer activities we decided that a significant contributing factor was the lack of a defined, documented, and institutionalized software development life cycle. Many key activities associated with effective software methods were not practiced or were practiced inconsistently. Donor funds were used extensively to provide government staff with training in software development techniques and tools, but there was little opportunity and no framework within which they could return to the job and apply their new skills. It was almost as though the training had no relevance to what they were being asked to do. Generally, the training did not expose them to advances in software development methodologies, processes, or life cycles. There needed to be a shift in emphasis resulting in a focused and sustained effort at building a process infrastructure of effective software engineering and management practices.

In the late 1980s, the Software Engineering Institute under the leadership of Watts Humphrey developed a model, the Capability Maturity Model for Software (CMM) [1], delineating the characteristics of an organization with mature, capable software processes. The model is not a life cycle model, but rather a model that relates to how well the software development technologies are being practiced and institutionalized. The CMM presents five levels of maturity, from immature and unrepeatable to a mature, well-managed collection of processes. The CMM is illustrated in Figure 1.

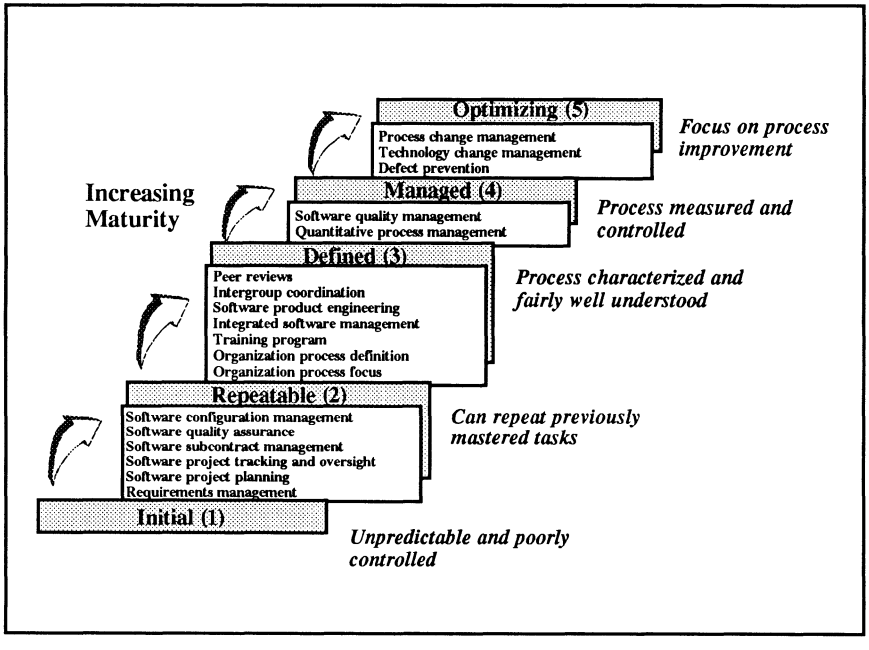

Figure 1: The Capability Maturity Model and Associated Key Practices

With the exception of Level 1, Initial, each CMM level also has associated with it several Key Practices. For example, one of the Key Practices in Level 2 is Software Quality Assurance. Each Key Practice consists of specialized processes and techniques which the organization must define and implement in its environment. In general, the higher the level on the CMM, the more Key Practices have been mastered and the more effective and predictable the organization will be in its system development activities.

Since its initial publication, the CMM has been widely used as a means for evaluating the state of progress in developing a stable and predictable

software development capability. It is now considered a benchmark against which an organization can evaluate its own processes and their effectiveness. The evaluation process is called a Software Process Assessment. From the assessment result a plan can be developed to improve critical areas or remove deficiencies.

THE MISD SOFTWARE PROCESS ASSESSMENT

In August, 1992, the consulting group introduced the CMM to the management and technicians of MISD. MISD wanted to use the model to assess its own software process maturity and give focus to areas needing improvement. An informal Software Process Assessment was completed by the consulting group and key MISD staff. As is true of the majority of software development organizations world wide, MISD exhibited the characteristics of a Level 1 (Initial) organization. In summary, the outcome of development efforts were not predictable. Some areas of development were performed well by some technicians but not performed consistently across the entire organization.

MISD, with the assistance of the consulting group, had developed a document which suggested the basis of a standardized system development life cycle. Although the document itself was not broadly used or accepted, it documented several approaches which were proven successful in the Kenyan environment and were broadly used by the developers. These included 1) a high level of user involvement throughout the life cycle achieved through workshops, interviews and training, 2) prototyping to provide additional requirements definition input, and 3) phased systems development starting with a small system scope easily delivered and then expanded.

Artifacts of the system development life cycle were limited to the completed system. Developers rarely completed formal documentation of user requirements, system design, or system usage. They delayed documenting the system until development was complete, but never had time to finish it because they were immediately assigned to a new development effort.

Management was grappling with the issue of project management of systems developed by MISD staff and systems being developed by contractors. It was difficult to get a good measure of what had been accomplished without project planning, tracking, and milestone reporting. Most projects were completed on short delivery schedules as operational prototypes. The software development process for each project seemed unique, rushed, amorphous, and difficult for management to control.

System development was a fluid process. Requirements and design changes were uncontrolled and based solely on conversations between developer and user. The user's requirements, both in terms of functionality and amount of data seemed to grow larger and larger as the development process continued. The developer accepted new requests from the user for functionality even during the late phases of development. The end result was confusion. Management did not know what was being promised. The developer lost sight of the original functional requirements and reacted by assuming that the most recent user request had the highest priority. System changes, even at the late stages of development were implemented without much testing and users were expected to "work out the bugs" after implementation.

GOALS FOR IMPROVEMENT

MISD recognized that in order to support the new more complex systems required by the Ministry, they must improve the quality and long-term maintainability of applications. Management's role in the processes would have to be strengthened and formalized. In developing the corresponding action plan, MISD management and the consulting group used the CMM Key Practices to identify areas in which to concentrate improvement efforts. Similar to what has been recognized by many government and commercial organization in other parts of the world, the software developers in MoF were being asked to perform without the guidance provided by a properly structured software development process. The following are the objectives established as a result of that evaluation:

Requirements Management (Level 2)
MISD needed to establish a formal requirements definition process, that included preparation of documentation. It needed to establish methods which would provide for clear understandings between the development and user organizations in order to guide both the developer's efforts and the user's expectations.

Software Project Planning (Level 2)
MISD needed to gain better control over project scope, delivery schedules, and deliverables. It needed to establish project planning and scheduling as a routine component of the system development life cycle. It needed standard milestones at logical points in the system development life cycle that would enable management to accomplish effective oversight.

Software Project Oversight (Level 2)
MISD needed to improve management oversight of projects by making the oversight and management activities routine. It needed to establish project management disciplines for management and the development teams. It

needed to establish oversight groups and responsible individuals for tracking and monitoring progress of development teams.

Software Quality Assurance (Level 2)
MISD needed to establish a quality assurance program, including developing and adopting written standards to be used in all systems development.

Organization Process Focus (Level 3)
MISD needed to establish an organizational structure that provided emphasis on the processes used to develop systems.

Organization Process Definition (Level 3)
MISD needed to develop and adopt written standards for system development. Since all standards could not be developed and institutionalized immediately, it would be important to identify and develop standards for areas which would provide the greatest opportunity for quality improvement in a reasonable time period.

Training Program (Level 3)
MISD needed to provide training for developers as the implementation of standard methods required. Training would focus on system development processes and on specific technical areas. It was considered essential to conduct training activities so that the material and timing are closely linked with the needs of the developers at their current stage in the life cycle.

TECHNOLOGY TRANSFER STRATEGY

In order to accomplish the MISD improvement objectives, there had to be an infusion of software development process technology from the consulting group. After observing staff performance following several earlier training activities and seeing that only marginal changes in abilities resulted, it was decided that a new strategy was needed. Since the government organization was clearly operating at the initial level in the CMM framework we decided to see if we could apply some of the underlying organizational development and technology transfer ideas on which the CMM is based.

MISD and the consultant settled on a three pronged approach as a strategy for beginning the technology transfer. First, the program would build upon the foundation of technical training already received by the developers. Secondly, a system development life cycle standard was needed in order to give the developers a guide to the processes they would be expected to follow in development. Thirdly, a series of intensive

training sessions and workshops were required to guide the developers each step of the way through the individual processes and deliverables.

MISD already had a document describing a software life cycle approach for building MoF systems, but it was not being used completely or consistently on projects. We decided that a key element in achieving an institutionalized process would be to define and document a software life cycle that was based on the processes currently applied by the most effective staff members. Important constituents of their approach were prototyping and significant user involvement. From this base, additional processes were added to increase effectiveness. A system development life cycle standard was documented by the consultants with the assistance of MISD personnel. The standard was conceptual in nature and was kept short and concise for the purposes of simplicity of implementation.

In addition, a collection of more detailed standards were developed in key technical areas such as user requirements definition, system design, user handbook, and security. These documents were written in a "how to" approach giving check lists, formats, and examples. The documents were written at a level that would be useful to developers. Again, to ease implementation, strong emphasis was placed on examples and formats which could be followed for specific project documentation and development. The documents were circulated for review both within the MISD organization and to consultants. The documents were then revised based upon the input received.

MISD management and the consultants anticipated that a structured approach to the development of systems would not be readily and voluntarily accepted. MISD programmers were concerned about the standards reducing their freedom to use a variety of technologies and approaches. Also, the use of standards introduced a new potential for oversight and review that had not been present before.

With a base of process documentation and guidelines available on what should be produced from the software development process, additional training was provided on specialized topics such as software life cycles, software quality assurance, software testing, and software reviews.

We realized that the availability of process documentation was not going to be sufficient to accomplish our objectives. Classroom training had its limitations too. It was time to switch to a mentoring model for training that would encourage a much closer association between the consultants and government staff in a real development situation. Use of the life cycle and standards on an actual Ministry development project was viewed as the most effective way to introduce the staff to the standard and

to gain their support for the more disciplined approach to system development. It was also a way to test the usability of the standards and gather input for revisions.

THE BUDGET MONITORING SYSTEM PROJECT

In April 1993 a new project, the Budget Monitoring System (BMS), was selected as a trial of the new MISD life cycle and standards. The team selected for the BMS project used the workshop technique which had been successfully applied on past MoF projects. Workshops are team meetings in which the participants review or discuss something relevant to the project and develop a consensus as to what course to follow. During startup of the project the workshops were used to review the project life cycle, as well as to identify the short term objectives and responsibilities. Once technical activities were underway, the workshops were used to address technical issues.

The BMS development team consisted of four developers from the government staff and two technical advisors from the consulting group. The government developers were the most senior staff members. They had received all the technical programming skills training that had been offered to date. Each of the developers had the experience of developing several stand alone applications in FoxPro.

The two technical advisors were senior technicians. Each had developed in excess of six systems. Both were host country nationals. One had been a civil servant working as an MISD member prior to being hired by the consulting group. The technical advisors were closely aligned with the government development team in background, native language, and understanding of the work environment. The role of the host country national technical advisors was to provide design and development input and to follow through with technical assistance to the government staff in the development of the artifacts.

The team was lead by two senior expatriate technical advisors. One advisor with standards development and process management experience took the role of workshop facilitator. The main role of the facilitator was to ensure that the development team followed and understood the processes associated with the development of the system. The other expatriate led the development team in the workshops and followed through with mentoring activities in the work area. The main role of this advisor was to provide technical skills assistance. The separation of roles of the two senior expatriates helped illustrate to the team, the working together and joining of the process and technical aspects of system development.

Team size was kept small to ensure coordination and the high participation levels of each developer in the workshop sessions. The workshops were also used to provide process training to programmers who were not BMS participants but who soon would be participating in other development projects. This facilitated institutionalization of standards beyond the immediate BMS team.

A key feature of each workshop was the participation of management. The MISD Deputy Director, who functions as the technical manager of the government programmer staff, participated in all workshops. He made management decisions in these workshops which helped guide the work of the teams. The Deputy also had the opportunity to apply project management discipline during the meetings.

The consultants structured the team's work both in and out of the workshops to follow the life cycle. The consultants also acted as mentors. The first workshop was a life cycle standard orientation. The second addressed project concept document preparation. Subsequent development phases progressed in this way, with workshops to review the life cycle phase and the deliverables, intervening time to complete the deliverables, and concluding with a deliverables review.

Workshop sessions were held on an average of one time each week. Initially workshops were planned to be conducted in sessions lasting a full day. This proved to be too long and the team did not maintain the high level of participation and concentration desired. The excessive amount of time allowed the group to divert to unproductive discussion. Subsequent sessions were shortened to no more than three or four hours each. This compression produced a higher level of urgency within the group in handling the matters at hand and resulted in a more disciplined approach to discussion.

The structure imposed by the life cycle identified several areas in which the staff required additional training because they were unable to complete a required deliverable. In previous projects, developers simply "slipped by" the areas they were not fully able to deal with, resulting in low quality systems. In this structured workshop approach, development was stopped at any point that the team required technical training.

The workshops were especially effective in reinforcing the expectations imposed by the software development life cycle and providing a forum for on-going training. They helped to clarify individual roles and responsibilities, created an environment which enhanced teamwork, and developed presentation skills. The workshops also helped to identify deficiencies in analytical skills, technical skills, writing skills, planning

skills, and organizational skills. Each of these was then addressed by training or mentoring. The workshops also were useful in providing insight into actual progress and became an essential project management activity.

CONCLUSIONS

Focusing on the key processes as outlined in the Capability Maturity Model provided the consultants and the government with specific areas in which to focus training and organizational development efforts. As a result of focusing on the model, MISD's goals have broadened from "developing information systems" to "developing high quality information systems using structured, manageable, and repeatable processes."

This focus has proven beneficial to the developers who have indicated strong support for the use of the MISD System Development Life Cycle and accompanying standards documents. They felt that the structure that was imposed helped them deliver a higher quality system. They also felt that the training interspersed throughout the development cycle as specific skill deficits were identified has greatly enhanced their abilities. They felt that additional projects using this approach will identify additional areas for training. The support that the developers had for the approach was strongly reflected in their regular attendance at workshops and their spirited participation.

This focus has proven beneficial to MISD management. Due to the structure provided by the life cycle and the high level project planning performed, management was able to track the progress of the system. Management has already seen an improvement in system documentation. It was also obvious from the material in the project notebooks that the writing skills of the developers have improved. Most importantly, this focus has proven beneficial to the users. The system delivered met user requirements. Training was conducted in a more organized fashion. At implementation, system "bugs" were reduced due to more thorough system testing prior to installation. BMS Phase 2, which will provide enhanced functionality, is already in the planning stage. The level of user satisfaction remains high.

On a negative note, the imposition of the structure of the workshops and standards raised the traditional questions of balance between "rapid" development and "delays" resulting from adherence to the process. The development of this system took longer than MISD management and the user organization anticipated. However, in reviewing the schedule, these delays were due more to the time required for training rather than the more disciplined approach. Future development efforts should progress more

quickly as the training requirements are reduced and developers become more knowledgeable of the processes. The longer term benefits of a higher quality system that is sustainable and maintainable have not yet been felt, but should be an important factor during BMS Phase 2.

The usability of the MISD System Development Life Cycle and accompanying standards has been validated. The standards are being revised slightly based on this experience. However, the series of workshops has pointed out the need for standards documents to be developed in additional areas, such as system design documentation, system performance, tender specifications, and user interface. MISD management has requested the development of standards for these additional areas.

MISD has concluded that the effort has certainly improved the development processes. However, the staff does not yet have the capacity to continue the workshops and follow the structured approach based on one team's experience. The workshops and mentoring approach will have to be used many more times, involving more projects and more developers before the process is institutionalized and repeatable.

The developers, consultants, and MISD management have agreed to pursue the next steps which will broaden workshop participation to include additional development efforts. Five new projects have been designated to follow the workshop-mentoring model.

REFERENCES

1. Humphrey, W. S. *Managing the Software Process* Addison Wesley, 1989.

2. Paulk, M. C., et al. *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24, Software Engineering Institute, 1993.

3. Glenday, G. 'Perspectives on the Computerization of Tax Systems in Sub-Saharan Africa,' in *Proceedings of the Conference on Information Technology and Fiscal Compliance*, Cambridge, MA, November 5-6, 1992.

4. Pradervand, P. *Listening to Africa: Developing Africa from the Grassroots*, Praeger Publishers, 1990

5. Broadman, J. 'A Summary of the Final Report on Microcomputer Adoption,' Harvard Institute for International Development, Harvard University, April 1986.