



Good Practice Note No. 20

Development, Acquisition, Implementation, and Maintenance of Application Systems

Part of a series of notes to help Centers review their own Center internal management processes from the point of view of managing risks and promoting value for money, and to identify where improvement efforts could be focused. The good practices described in this series of notes should not be interpreted as minimum standards as not all may be applicable to every Center.

SUMMARY

The acquisition, development, implementation and maintenance of new or revised applications systems (simply referred to as systems development in this note) makes significant demands upon the financial, human and IT resources of the research centers. It is essential for efficiency reasons and to ensure the quality of the system implementation that system development processes are managed in accordance with information technology industry good practice.

This Good Practice Note (GPN) supplements the earlier released overview GPN on the Management of IT Risks and analyzes the risks, opportunities and critical success factors in the management of the systems development processes. This note also identifies the following subsidiary good practices in relation to systems development:

System development project management framework

- Conduct project risk analysis
- Develop a project charter and plan
- Hire knowledgeable implementation partner where internal resources are insufficient



- Define criteria for selection of implementation partner
- Define responsibilities of parties involved in the system development process

System development life cycle (SDLC) methodology

- Define system requirements to support user and business needs
- Design and implement cost-effective security control
- Ensure integrity of licensed application software
- Ensure comprehensive contract application programming
- Develop and monitor the implementation of training plan
- Conversion Data
- Control promotion of the new system to production
- Conduct a post-implementation review
- Establish program change management process



Good Practice Note No. 20

Development, Acquisition, Implementation, and Maintenance of Application Systems

INTRODUCTION

The acquisition, development, implementation and maintenance of new or revised applications systems (simply referred to as systems development in this note) makes significant demands upon the financial, human and IT resources of the research centers. It is essential for efficiency reasons and to ensure the quality of the system implementation that system development processes are managed in accordance with information technology industry good practice.

The process of managing the systems development of application systems is a component of the overall management of IT processes. The overall organizational environment has significant impact on the success of applications systems implemented.

This Note supplements the overview Good Practice Note on the Management of IT Risks, where a summary of good practices affecting systems development was identified. In summary these comprise:

- Adopt a general project management framework for IT projects
- Adopt a system development life cycle (SDLC) methodology for developing, acquiring, implementing and maintaining IT systems and related technology.
- Ensure that the SDLC methodology fully addresses the process of identifying, specifying and approving requirements
- Apply standard procurement procedures, consistent with CGIAR guidelines, to IT related procurement
- Design and implement test plans and retain documentation of results
- Ensure all system development is accompanied by adequate user support materials



- Prepare and monitor implementation plans
- Prepare and monitor data conversion plans
- Apply formal processes for new system acceptance and transfer to production
- Implement a formal approach to system change management

This focused note provides an analysis of the potential risks and critical success factors for the systems development processes, and identifies some subsidiary good practices to elaborate on certain aspects of those good practices listed in the overview note. The information in this note has been drawn from the COBIT framework, good practices identified during the reviews of system implementation conducted by the CGIAR Internal Audit and from external sources of good practices.

RISKS, OPPORTUNITIES AND CRITICAL SUCCESS FACTORS FOR THE SYSTEMS DEVELOPMENT PROCESSES

Application systems go through a lifecycle of processes that need to be managed. The key objectives of the management of systems development processes are:

- An effective and efficient approach of identifying automated solution and implementation partners is in place to satisfy user and business requirements
- Timely and effective design and implementation of automated solutions which support the business processes
- Adequate business procedures are developed and maintained to ensure proper use of the application systems
- System changes to meet the evolving business requirements are implemented while preventing business disruption, unauthorized alterations and errors.

The table below summarizes some of the significant opportunities, risks and critical success factors in the management of the system development process.



Opportunities	Risks	Critical Success Factor
<p>The Centers' business processes are improved with the deployment of automated solutions, thus facilitate the achievement of goals and objectives.</p>	<p>Solutions identified do not effectively support the business operation due to following factors:</p> <ul style="list-style-type: none"> ● Lack of knowledge of possible solution available ● System requirements were poorly defined ● The possible solutions were not evaluated based on appropriate business and user requirements. 	<ul style="list-style-type: none"> ● Project management controls such as project organization, planning, and monitoring ● Employment and utilization of a system development methodology. ● User and management involvement in vendor and product selection ● Established criteria for selecting vendor and vendor products
	<p>Poor quality of application system deployed, as:</p> <ul style="list-style-type: none"> ● Incompatibility with existing systems and infrastructure ● Inability to evolve ● Inappropriate technology and architecture ● Not user friendly. 	<ul style="list-style-type: none"> ● Clear definition of business, user, and technical requirements ● Active involvement of appropriate parties, i.e. senior management, users, IT.
	<p>System data and programs are not free from error, such as:</p> <ul style="list-style-type: none"> ● inaccurate system processing ● invalid, inaccurate and incomplete transactions processed ● incorrect data stored in the system ● lack of system audit trail. 	<ul style="list-style-type: none"> ● Development and implementation of comprehensive test plan and strategy. ● Thorough testing of systems, to include appropriate test databases, program libraries, and operating procedures. ● Conducting adequate user training ● Implementing program change procedures.



Opportunities	Risks	Critical Success Factor
	<p>System may not be performing as intended, as:</p> <ul style="list-style-type: none">● System not configured based on user and business requirements● Inadequate testing● Unauthorized changes made on system programs.	
	<p>System may not be used for intended purpose, for the following reasons:</p> <ul style="list-style-type: none">● Inadequate user training● Insufficient user procedures● lack of post implementation evaluation.	<ul style="list-style-type: none">● Post-implementation evaluation.● Change management process.● Formulation and implementation of complementary control procedures.
	<p>Business processes may be disrupted by:</p> <ul style="list-style-type: none">● the unavailability of system● unauthorized access to system.	<p>Integration of the information security program into the systems development methodology, in terms of :</p> <ul style="list-style-type: none">● Business continuity planning● Assigning appropriate access to production environment, i.e. application security and database administration.
	<p>System cannot be modified easily to meet the changing needs of the organization for:</p> <ul style="list-style-type: none">● lack of vendor support● incomplete system documentation● inadequate in-house skills to maintain the system.	<ul style="list-style-type: none">● Employment and utilization of a system development methodology● Project management controls.



Opportunities	Risks	Critical Success Factor
<p>Solutions are completed and deployed using reasonable allocated resources and within appropriate timeframe to address the operational needs of the Center.</p>	<p>Excessive cost and delay in developing, implementing or maintaining application systems, for the following reasons:</p> <ul style="list-style-type: none">● Lack of management support● Inadequate stakeholders involvement● Inappropriate SDLC methodology applied● Inadequate or inappropriate resources allocated● Poor project management● Inadequate controls in the SDLC process● Inadequate contractual protection● Scope variations● Misuse of systems development methodologies● Mismanaged costs	<ul style="list-style-type: none">● Employment and utilization of a system development methodology● Project management controls.
<p>Historical data from the old system are accessible in the deployed solution.</p>	<p>Historical data may be inaccurate and incomplete, due to:</p> <ul style="list-style-type: none">● insufficient planning for data conversion/migration● Master data and transactions may not be completely and accurately converted.● Migration was not properly tested.● Fields may not be properly mapped.	<ul style="list-style-type: none">● Data migration planning.● Validation and control of migrated data.



ELABORATION ON THE OVERVIEW NOTE'S GOOD PRACTICES FOR SYSTEM DEVELOPMENT

System Development Project Management Framework

Good practice

Conduct Project Risk Analysis

The earlier overview Good Practice Note on Management of IT Risks underscored the significance of adopting general project management framework and system development life cycle (SDLC) methodology for systems development.

The adopted SDLC methodology should provide, in each proposed system development project for an analysis and documentation of the security threats, potential vulnerabilities and impacts and the feasible security and internal control safeguards for reducing or eliminating the identified risk.

Good practice

Develop a Project Charter and Plan

The development of Project Charter and Plan should be a salient feature of the adopted SDLC methodology. The Project Charter defines the project objectives, manner in which the project will be managed and the governance surrounding the project. While the Plan defines the micro level activities and deliverables for each phase.

This prevents the project from getting unfocused, as it may go off track because of recurring or additional functionality changes made along the way. This inevitably results in time and cost overruns and can even lead to a project shut down which occurs when an implementation team loses sight of the original business objectives of the application. A checklist of milestones, based on those objectives, helps ensure timely implementation.

Good practice

Hire a knowledgeable implementation partner where internal resources are insufficient

Few Centers have the internal resources required to ensure a successful system implementation. To bridge that gap, Centers should turn to an implementation partner, typically a technology firm with



implementation experience that has the knowledge, industry expertise, and professional staff to ensure a successful implementation.

Centers should take the time to hire the right implementation partner because their fees often run higher than the cost of the software itself. In fact, some organizations have taken as much or more time to select their implementation partners as they have to find the right application. The selection process should conform to CGIAR procurement guidelines in relation to selection of consultants, which will generally involve a competitive process.

An effective implementation partner is:

- Not a hardware or software vendor
- Experienced in overseeing large-scale implementations
- Well-versed in the capabilities of the software
- Technically skilled to provide the integration required with other systems within the organization
- Equipped with the business acumen to maximize the opportunities provided by the software in terms of fulfilling the business need.

Good practice

Define criteria for the selection of implementation partners

Below are some items to consider in selecting implementation partners:

- Experience with the software and with large-scale implementation
- Knowledge of the centers' business and their business needs
- Financial and technical viability and ability to provide post-implementation support and maintenance
- Years of operations, clients services, product offerings and geographic reach.
- History of post-sales service and support.



Additional Guide for Contents of Request for Proposal (RFP) for System Selection	
Item	Description
Product versus system requirements	The chosen vendor's product should come as close as possible to meeting the defined requirements of the system. If no vendor's product meets all of the defined requirements, the project team, especially the users, will have to decide whether to accept the deficiencies. An alternative to living with a product's deficiencies is for the vendor or the purchaser to make customized changes to the product.
Vendor viability/financial stability	The vendor supplying or supporting the product should be reputable and therefore, should be able to provide evidence of financial stability. New vendor may not be able to prove financial stability. New vendors, especially if the product is new and or you are the first customer, present substantially higher risk to the organization.
Customer references	Project management should check vendor-supplied references to validate the customer's claims of product performance and completion of work by the vendor.
Availability to complete and reliable documentation	The vendor should be willing and able to provide a complete set of system documentation for review prior to acquisition. The level of detail and precision found in the documentation may be an indicator of the attention to detail and precision utilized within the design and programming of the system itself.
Vendor support	The vendor should have available a complete line of support products for the software package. This may include a 24 hour, 7 days a week help line, onsite training during implementation, product upgrades, automatic new version notification and onsite maintenance if requested.
Source code availability	The source code should either be received from the vendor initially or there should be provisions for acquiring the source code in the event that the vendor goes out of business. Usually these clauses are part of software escrow agreement in which a third party holds the software en escrow should an event occur. The acquiring company should ensure that product updates and program fixes are included in the escrow agreement.
Number of years experience in offering the product	More years indicates stability and familiarity with the business the product supports.



A list of recent or planned enhancements to the product, with dates	A short list suggests the product is not being kept current.
Number of client sites using the product with a list of current users	A large number suggests wide acceptance of the product in the marketplace.
Acceptance testing of the product	Such testing is crucial in determining whether the product really satisfies your system requirements. This is allowed before a purchasing commitment must be made.

Good practice

Define responsibilities of parties involved in the system development process

Apart from having a defined project management and SDLC methodology, it is advisable that key players are involved in the system development process. The various roles and responsibilities of groups/individuals that may be involved should be properly defined. These are summarized as follows:

Senior management

Commits to the project and approves the necessary resources to complete the project. This commitment from senior management helps ensure involvement by those needed to complete the project.

User management

Assumes ownership of the project and resulting system, allocates qualified representatives to the team, and actively participates in system requirements definition, acceptance testing and user training. User management should review and approve system deliverables as they are defined and accomplished.

Project steering committee

Provides overall direction and ensures appropriate representation of affected parties. The project steering committee is ultimately responsible for all costs and timetables. This committee should be comprised of a senior representative from each function that will be significantly affected by the proposed new system or system modification. Each member must be authorized to make decisions relating to system design that will affect their respective departments. The project manager must be a member of this committee and in some cases may serve as chair. The project steering committee should perform the following functions:



- Review project progress regularly and hold emergency meetings when required.
- Serve as coordinator and advisor.
- Members of the committee should be available to answer questions and make user-related decisions about system and program design.
- Take corrective action. The committee should evaluate progress and take action or make recommendations regarding personnel changes on the project team, replanning budgets or schedules, changes in project objectives and the need for redesign.

Project sponsor

Provides funding for the project. Works closely with the project manager to define the success measurement for the project. It is crucial that success is translated into measurable and quantifiable terms. Data and application ownership are assigned to a project sponsor. A project sponsor is typically the senior manager in charge of the primary business function the application will support.

Project manager

Provides day-to-day management of the project, ensuring that it remains in line with the overall direction, ensures appropriate representation of affected departments, ensures the project adheres to local standards, ensures that deliverables meet agreed quality standards, resolves inter-departmental conflicts and monitors and control costs and project time table. This person can be an actual end-user or a member of the system development team. Where projects are staffed by personnel dedicated wholly to the project, the project manager will have a line responsibility for such personnel.

Systems development project team

Completes assigned tasks, communicates effectively with users by actively involving them in the development process, works according to local standards and advises project manager of necessary project plan deviations.

User project team

Completes assigned tasks, communicates effectively with the system developers by actively involving themselves in the development process, works according to local standards and advises project manager of expected and actual project plan deviations.

Security officer

Ensures that system controls and supporting processes provide an effective level of protection, based on data classification set in accordance with corporate security policies and procedures, consults throughout the life cycle on appropriate security measures that should be incorporated into the system, reviews



security test plans and reports prior to implementation; evaluate security-related documents developed in reporting the system's security effectiveness for accreditation; and monitors the security system's effectiveness periodically during its operational life.

Quality Assurance

Reviews results and deliverables within each phase and at the end of each phase confirm compliance with requirements. The points where reviews occur depend on the system development life cycle methodology used, the structure and magnitude of the system and the impact of potential deviation.

System Development Life Cycle (SDLC) Methodology

Good practice

Define system requirements to support user and business needs

Typical SDLC methodology provides for a clear definition of user and business requirements. The following should be considered during this phase of system development:

- System requirements should be defined prior to deciding on the acquisition strategy for the system, i.e. whether acquired off the shelf, developed internally, through contract, by enhancing existing system or combination of all these.
- Requirements should be focused not only on the functional and operational requirements but also include performance, safety, reliability, compatibility, scalability, security and legislation.
- Requirements should be solicited from cross functional teams of the center, not only from the direct users or project sponsor.
- Teams defining system requirements should be advised to go beyond their current state and envision the "could be" set-up. Realignment or re-engineering of processes should be considered.
- All external and internal interfaces should be specified and designed.
- Refine the system requirement before starting on the software design.

The organization's SDLC methodology should ensure that the system design is reassessed whenever significant technical and /or logical discrepancies occur during system development or maintenance.

Good practice

Design and implement cost-effective security controls



The Center's SDLC methodology should require that the basic security and internal control aspects of a system to be developed or modified should be assessed along with the conceptual design of the system in order to integrate security concepts in the design as early as possible.

Application controls should:

- guarantee the validity, accuracy, completeness, timeliness and authorization of inputs, processing and outputs
- be carefully examined so that the costs of implementing security and controls do not exceed the benefits.

The decision requires formal management sign-off. All security requirements should be identified at the requirements phase of a project and justified, agreed and documented as part of the overall business case for an information system. Security requirements for business continuity management should be defined to ensure that planned activation, fallback and resumption processes are supported by the proposed solution.

Good practice

Ensure integrity of licensed application software

Centers should ensure that for licensed software acquired from third-party providers, the providers have appropriate procedures to validate, protect and maintain the software product's integrity rights. Considerations should be given to the support of the product in any maintenance agreement related to the delivered product.

Good practice

Ensure comprehensive contract application programming

The Centers' SDLC methodology should provide that the procurement of contract programming services be justified with a written request for services from a designated member of the IT function. The contract should stipulate that the software, documentation and other deliverables are subject to testing and review prior to acceptance. In addition, it should require that the end products of completed contract programming services be tested and reviewed according to the related standards by the Center's IT function quality assurance group or equivalent before payment for the work and approval of the end product.

The contract should include provisions for IP rights over developed programs to be retained by the contracting Center. Further, if program source codes will not be turned over to the Center, an escrow



agreement should be considered, to ensure that codes will be available even if the developer ceases operations.

Good practice

Develop and monitor the implementation of a Training Plan

Staff of the affected user departments and the operations group of the IT function should be trained in accordance with the defined training plan and associated materials, as part of every information systems development, implementation or modification project.

Educated users take advantage of the benefits of an application by using it more often and more fully. Their satisfaction on the job and with the technology increases because they have the knowledge and skills to work better, not harder. Centers should be aware of the crucial role new applications play in employee productivity and design and deliver effective training so users understand:

- How the application works
- How it helps them do their jobs better
- How it contributes to the company's overall success

Training should also continue to support end-users as long as the application remains in place at the organization because as technology becomes increasingly sophisticated, it correspondingly becomes more complicated to maintain and understand. By training and supporting users over the long-term, Centers gain the increase in productivity and ultimately, the competitive edge they seek in a software application.

Good practice

Data Conversion

Management should require that a data conversion plan be prepared, defining the methods of collecting and verifying the data to be converted and identifying and resolving errors found during conversion. Tests to be performed include comparing the original and converted files, checking the compatibility of the converted data with the new system, checking master files after conversion to ensure the accuracy of master file data and ensuring that transactions affecting master file update both the old and the new master files during the period between initial conversion and final implementation. A detailed verification of the initial processing of the new system should be performed to confirm successful implementation. Management should ensure that the responsibility for successful conversion of data lies with the system owner.



Good practice

Control promotion of the new system to production

Management should define and implement formal procedures to control the handover of the systems from development to testing to operations. Management should require that system owner authorization is obtained before a new system is moved into production and that, before the old system is discontinued, the new system will have successfully operated through all daily, monthly and quarterly production cycles. The respective environments should be segregated and properly protected.

Good practice

Conduct a Post-implementation Review

The Center's system development life cycle methodology should require that a post-implementation review/evaluation be conducted to assess whether the operational information system requirements/benefits (e.g. capacity, throughput, etc.) are delivered and whether the development process was conducted in the most cost effective and efficient manner.

Benchmarks, in terms of anticipated improvements are defined as well as how these can be measured. Although the metrics vary, they fall into one or more of the three broad categories:

- Time savings
- Cost savings
- Quality improvements.

The evaluation of project activities should be linked to an overall learning process. Lessons learned should be accounted for, documented and made available for planning future projects.

Good practice

Establish program change management process

In order to ensure the integrity of system processing, despite modifications that need to be implemented to address the changes in business requirements, a program change management process should:

- ensure that all requests for change are approved and assessed in a structured way for all possible impacts on the operational system and its functionality.



- monitor changes to application systems
- establish parameters defining emergency changes and procedures to control these changes when they circumvent the normal process of technical, operational and management assessment prior to implementation. The emergency changes should be recorded and authorized by IT management prior to implementation
- ensure that associated documentation and procedures are updated accordingly.
- ensure that the release of software is governed by formal procedures ensuring sign-off.



Exposure Draft: November 2005
(Adopted without Change)
Author: Vima Salazar