Anti-Retroviral Therapy Information System (ARTIS)
High Level Requirements

Date: December 2006

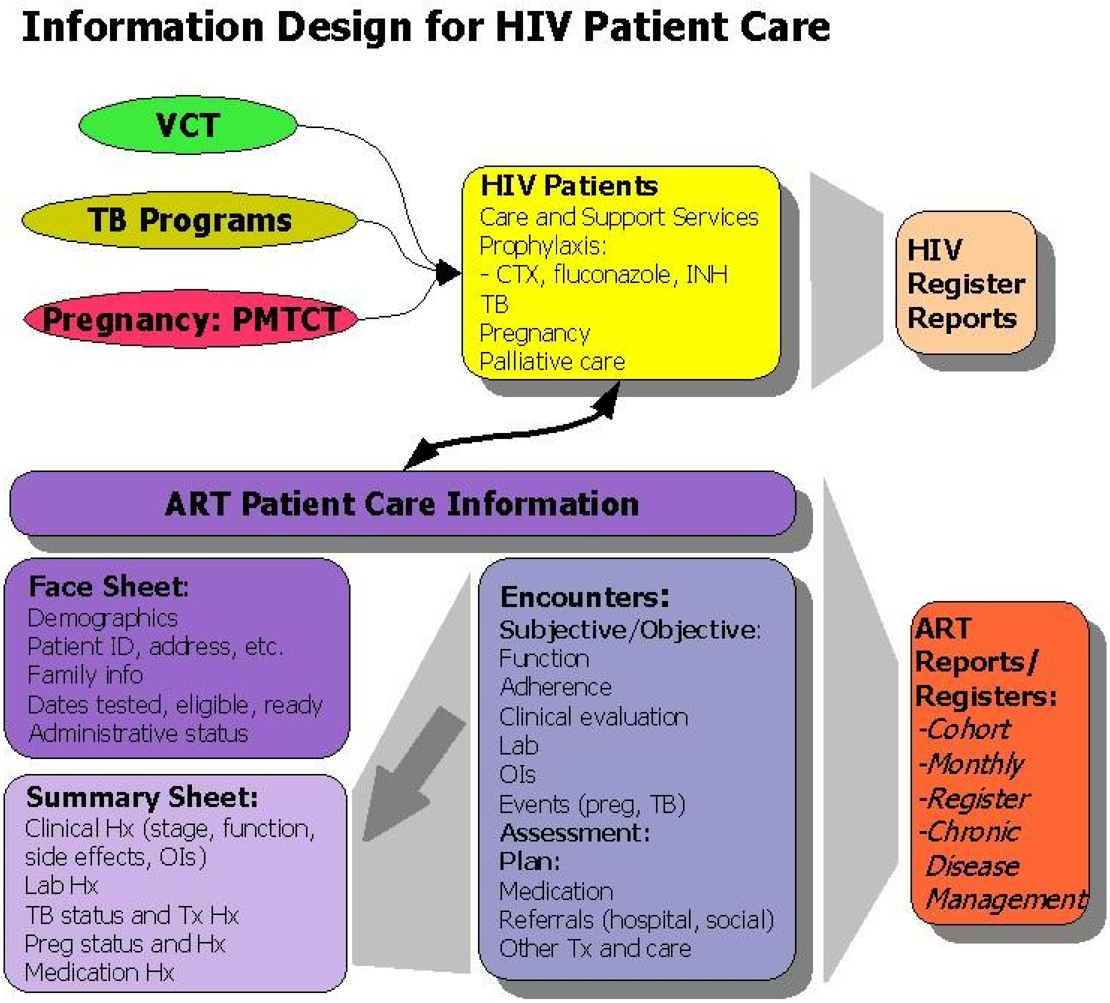
Version: Draft 5

Mark H. Spohr, MD

mspohr@nnk.com

1. Executive Summary

There is an immediate and pressing need to install electronic information systems to manage HIV/AIDS patients. This is due to the extent of the epidemic, the complexity of the disease, and the wide variety of responses that must be coordinated. This diagram gives an overview of the information that must be coordinated.



1.1. Current Information Systems

Current information systems are primarily paper-based and typically consist of documents created locally to meet the immediate need. The WHO has produced a set of paper reference documents (WHO Patient Monitoring Guidelines) which includes a minimum data set for registering HIV patients and monitoring their care. These forms are in use in some locations and have been adapted for use in others.

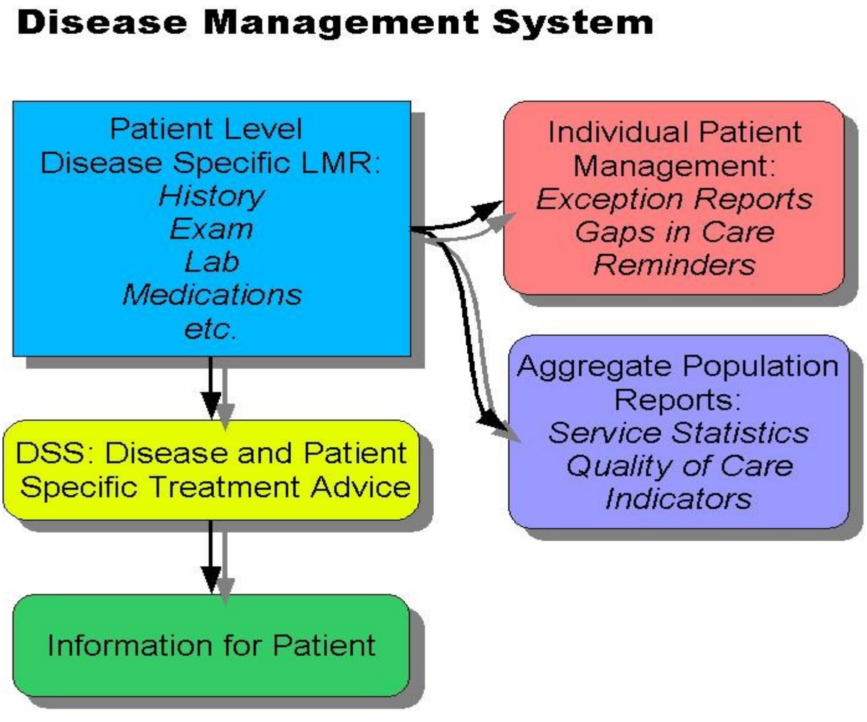
The limitations of paper based information systems are a constraint to patient care, resource management, program management, monitoring and evaluation and research. The limitations of paper based information systems are due to the difficulty in tracking patients over time (longitudinal medical record) and the difficulty of aggregating data.

A number of electronic information systems have been built and are in use and there is rapid development in this area. This document set includes an evaluation of the current systems which are suitable for use in developing countries.

1.2. Opportunities

There is a pressing need for electronic information systems to manage the HIV epidemic. Only electronic systems will be able to properly track the large number of patients over long periods of time and be able to report on the effectiveness of treatment, extent of the epidemic, and manage the diverse resources needed to address all aspects of the epidemic.

Information systems must be able to track individual patients from diagnosis of HIV infections through many years of care and support services and ART with complex drug regimens. In addition, there is a significant management problem of coordinating multiple care providers and services including laboratory. The logistics of drugs and supplies is a significant challenge. National and program managers must have good information on their activities and the effectiveness of their programs. International donors must also know how to direct resources.



Disease management (DM) functions are particularly useful for managing ART because of the complexity of treating HIV. Some examples of DM functions for ARV systems include:

- identifying contraindications among the ARV combinations or with drugs for treating OIs;
- issuing clinicians reminders or warnings about side effects and resistance associated with specific antiretrovirals;
- synthesizing multi-factorial laboratory and clinical data to monitor patient progress.

Some electronic LMRs include additional software intelligence that provides DM functions. DM (also sometimes called a Disease Registry) is a specific type of LMR that is focused on recording individual patient information for one or more diseases and provides the staff with the following points:

• The point of care to provide patient specific information and advice to support decision making for patient care;

- Between care visits for identifying gaps in patient care (exception reports);

- Periodically provide status reports providing aggregate information on population progress for purposes of monitoring care and improving care delivery.

1.3.Goals

The following are the objectives of this document:

- Capture the high level requirements for the ART Information Systems (ARTIS) in the PEPFAR supported countries.
- Identify a number of applications designed to support ART in PEPFAR supported countries.
- Provide the tools needed to select appropriate developers/providers of ART information systems.
- Provide the tools needed to select and modify as needed a specific ART solution to meet country requirements.

The following are the goals for this High Level Requirements Document:

a)Deliniate the functional system and b)operational requirements for ART at a high level.

The goal of this document is to systematically list required functions of ART information systems to enable effective project definition and timely implementation of an appropriate ART information system. These are called High Level Requirements documents as they provide a broad overview of 'what' the ART information system should be able to do and not 'how' these functions will be performed. The ART information system selected can be a Commercial-Of-The-Shelf (COTS) product, a built-to-specification solution or a combination of these two that will meet specific program needs and be compatible with other systems.

1.4.Limitations of this document

The following lists what this project/document is *not* intended to do:

- Identify requirements that are not directly relevant to ART such as hospital management.
- Identify detailed operating procedures for clinics or ancillary services.
- List all possible vendors that provide software systems that might meet the requirements of some appropriate solutions, in whole or in part.
- Recommend specific vendors for specific GAP country ART solutions.
- Provide in-depth functional and operational characteristics of an ART information system.
- Describe design aspects of the ART information system including object models or database schemas.

2. Functional Requirements

This section describes the functions that should be included in ART software and the rationale for each. The ART software list evaluates each software system using these terms for ease of reference.

2.1.Patient Care Tracking

Functions that support individual patient care.

2.1.1.Patient ID

Description of the patient ID features and formats supported. Also, tools that are available to prevent duplicates and search for records.

Identifying individual patients accurately is central to the proper functioning of the software. The main tool for this is a patient identification number. Identification numbers may be assigned by various clinics and agencies. Patients may have more than one identification number (clinic, national, HIV register, etc.) The software should offer flexible support of multiple patient identification numbers to accommodate the current schemes.

2.1.2.Unique ID Tools

Describe tools available to prevent duplicate Ids, prevent split records, and locate patients in DB.

It is crucial that each patient have a unique record and that each patient has only one record. The consequences of duplicate identification numbers and multiple records per patient (split records) are incomplete and inaccurate information which can lead to inappropriate care.

Central to unique ID tools are methods to locate patients using various criteria such as name, address, identifying numbers, age, sex. More the factors that can be included, better the chance of uniquely identifying a patient.

2.1.3.Family or Contact Data Record Linking

Links to other records in the system to record family groups or disease contacts.

Type of linking.

Family and contact groups are important for purposes of providing care and support services and also for contract tracking for HIV and TB as well as other communicable diseases. A system that supports the linking of individuals in the medical record can help coordinate care and improve outcome. It is preferable that this linking be done by forming associations between individual patient records with a provision to indicate the type of relationship. This type of linking maintains the relational integrity of the databases. A less preferable method is to create separate lists of family and contacts within individual patient records. This method is difficult to keep up to date as individuals change.

2.1.4.Demographics

Demographic fields such as address, contact information, birth date, sex, other ID numbers, etc. List fields that are supported.

Demographics refers to patient identifying and location information. It is essential to adequately identify the patient and the patient's location. The software should allow for a wide variety of identifying information. Some of this information will need to be customized to the region of the patient since methods of identifying location vary and the format of identification numbers and other information such as dates also have regional differences. The software should be checked carefully to ensure that it accommodates necessary informations or that it can be easily modified.

2.1.5.Patient status

Current patient status indicator and values that are supported. Examples include eligible for treatment, receiving treatment, deceased, lost.

Patients will have a variety of administrative states at various times from registration as HIV positive, through eligibility and preparation for treatment, and through various programs of care. The definition of these states will often be defined for a specific program or region. The software should support a flexible definition of multiple patient states that can be updated for regional and other program requirements.

2.1.6.Contact tracking

Fields to list contacts for communicable disease contact tracking.

Case contract tracking is particularly important for communicable diseases such as HIV and tuberculosis. It is important to ensure that contacts receive appropriate information and care. If the software has specific fields to facilitate contract tracking, it can improve management.

2.1.7.Appointment scheduling

Scheduling of patient appointments. Indicate capabilities such as multiple referrals, flags and reminders for overdue clients.

In the management of HIV/AIDS which is a chronic disease which requires continuous treatment, the management of patient appointments is crucial. Patients must be scheduled at appropriate intervals to ensure that they receive proper care and medication continuity. In addition, the important function of patient follow-up can be greatly facilitated by an electronic appointment system. Appointment scheduling systems facilitate the smooth flow of patients by matching patients to staff availability. In addition, patients who miss appointments should be flagged for personal contact to ensure they receive continuous care and do not run out of medication.

2.1.8.Medical History

Indicate capabilities. A complete medical history should include social history, family history, patient’s past medical history, immunizations, toxic exposures, habits, review of systems, and history of present illness.

Each patient's past medical history is important and must be readily available to ensure that they are receiving appropriate care. Often the most pertinent facts of the patient's medical history are presented on a “summary sheet” for review at the point of care. The software should allow for a complete and flexible recording of each patient's medical history. The history is typically recorded in categories such as social history, family history, patient’s past medical history, immunizations, toxic exposures, habits, and review of systems. The system that is defined for use in any region will depend on local practices. Software selection should be dependent on ensuring that it can support the required format and information.

2.1.9.Behavioral history (incl. STD risk factors)

Specific behavioral history relating to STD/HIV risk factors. Indicate risk factors tracked.

Since HIV is a sexually transmitted disease and can also be transmitted by contact with bodily fluids, it is important to track specific behavioral risk factors so that patients and contacts can be properly educated to prevent transmission of disease. In addition, certain risk factors can also alert the clinician to important co-morbidities such as hepatitis and tuberculosis. The software should have a section where behavioral history can be recorded in a flexible format that will meet regional and program requirements.

2.1.10.Allergies

Includes drug, non-drug, and environmental allergies. Indicate if support flags to warn on Rx?.

Allergies are important to ensure patients receive proper care. The software should support recording and appropriate display of allergy information. This is especially important for drug allergies which may require modification of treatment protocols. Typically drug allergies are displayed on a patient summary sheet for use at the point of care.

2.1.11.Physical exam

Recording of the physical exam by organ system. Indicate system for recording exam and number of elements. Ensure the neurological exam includes Karnofsky score.

The physical exam is essential to take care of HIV patients since it guides clinical therapy and can alert the clinician to important opportunistic infections. The exam should be performed and recorded at each clinical encounter. The software should permit fast and easy recording of significant positive and negative clinical findings. The software should also permit an easy review of prior exams so that a patient's clinical progress can be monitored. Software should permit the flexible recording of priority exam information and this facility should be easily customized to meet regional requirements and protocols. The neurological Karnofsky score is important for tracking the neurological impact of HIV and treatment and the software should support this explicitly.

2.1.12.Pregnancy

Pregnancy history (Gravida/Parity), complications, current pregnancy dates and exam/lab tracking, EDC.

Pregnancy has important implications for patients who are HIV positive. The most crucial is to initiate a protocol to prevent mother to child transmission (PMTCT) of the HIV virus. In addition, pregnancy is a significant stress and an important time to ensure that both mother and child receive the best care and support services to achieve the best possible outcome. In order to do this, it is crucial to record the pregnancy history (including prior complications and outcomes) and the details of the current pregnancy including detailed exam and laboratory results. Access to this information will help ensure that clinicians can provide the best care. The software should provide specific features to ensure that the PMTCT protocol is being followed including prompts to ensure that the appropriate peri-delivery treatment and patient education is being followed.

2.1.13.HIV staging

HIV staging by WHO staging system or other staging system used.

The WHO HIV staging protocol is crucial to patient management and an important factor in monitoring the progress of the disease and treatment. The software should support this staging system explicitly. In addition, other HIV staging systems to meet regional requirements may be optionally supported.

2.1.14.TB specific Dx and Tx

Tuberculosis screening and management features including symptoms screen, lab orders and results, resistance testing, and treatment.

It is imperative that the software include features to support the diagnosis and treatment of tuberculosis. Since estimates of coexisting TB and HIV infection run in the neighborhood of 50%, all HIV patients must be monitored closely for tuberculosis and treated appropriately should include that for active disease as well as prophylaxis. The TB status of the patient should be assessed at each clinical visit and appropriate treatment and prophylaxis should be recorded.

2.1.15.STD specific Dx and Tx

Sexually transmitted disease (STD) exam, lab, diagnosis and treatment.

The existence of other sexually transmitted disease is important in the management of HIV patients since this can affect disease transmission and can be significant opportunistic infections. The software should record the results of STD exams, laboratory, diagnosis and treatment.

2.1.16.Malaria specific Dx and Tx

Malaria tracking including lab, resistance, and treatment.

Malaria is an important opportunistic infection for HIV. In malaria endemic areas, HIV patients are often co-infected with malaria and this can affect their ARV treatment response. Malaria infection needs to be monitored throughout ARV treatments as it may not be apparent until the patient has been on ARV medication. Malaria prophylaxis may also be part of the treatment protocol. The software should include specific features to ensure that the patient's malaria diagnosis, treatment and prophylaxis status is monitored and recorded continuously at clinical visits.

2.1.17.ARV Tx specific functionality

Describe specific functionality to address ARV treatment and advice.

HIV positive patients will progress to the point where they receive anti-retroviral (ARV) therapy (ART). Current protocols for ART demand a high degree of clinician and patient communication and patient education. The protocols also specify criteria for monitoring treatment and changing medications when appropriate. It is essential that the details of patient ART be recorded and available for review and assessment. Software should permit treatment regimens to be easily recorded and reviewed. It should allow regimens to be modified as protocols change. Adverse reactions, complications, and resistance to medications be recorded.

A fertile area for advanced software is to incorporate treatment advice into the software so that the software can evaluate the patient's condition and response and make suggestions of appropriate treatment based on approved protocols. These advanced treatment advice systems can improve patient care. However, they are difficult to design and implement and they must be modifiable to adapt as protocols change.

2.1.18.PMTCT information tracking

Specific functionality to track Prevention of Mother to Child Transmission status and indicators. Indicate information supported such as initial information, counseling and testing, post test counseling, treatment at delivery, and child post-partum testing.

The opportunity to prevent mother to child transmission (PMTCT) of the HIV virus during pregnancy and delivery is an important intervention. PMTCT programs have been developed with a specific protocol of patient testing, education, and treatment. If the PMTCT protocol is followed carefully, the transmission of HIV to the child is dramatically reduced. Therefore, HIV patients who are pregnant should be enrolled in the PMTCT protocol. The software should monitor the specific steps of the PMTCT protocol to ensure that they are followed properly and that patients receive appropriate care and instruction. Since the various actions of the protocol take place in a period of time before, during and after delivery, software can play an important role in ensuring that all of the steps are followed properly.

2.1.19.Other disease specific exam info

Indicate any other specific diseases that are explicitly covered and the information that is recorded for each.

Software may include features for monitoring diseases other than those covered above. These may include opportunistic infections or related conditions. Software should be evaluated to determine if these additional features can be of use in the program. The flexibility and customization features of the software will be important factors in making it useful.

2.1.20.Problem list (conditions and status)

Medical problem list with date of onset, resolution. List any specific nomenclature supported.

The concept of a "problem list" is important in the long term care of patients and is significant in chronic disease management. The problem list is an accounting of an individual patients current medical problems (diagnoses) and can also include resolved problems. It is important to maintain this and update it at each clinical visit. It is an efficient method of making sure that all of a patient's problems are assessed at each clinical visit. The problem list is often presented as part of the patient summary sheet for use at the point of care. Software that supports a patient problem list can lead to improved patient care. The problem list software should allow for easy entry and update of the conditions on the list and should keep track of the dates of the activity of each patient's problem.

Problem list software can use free text or problem information coded with a specific coding system such as the WHO ICD or the Read codes. Free text is generally discouraged since it can be difficult to aggregate for analysis but often the diagnosis codes are too cumbersome for efficient use. When dealing with a specific chronic disease such as HIV infection, it is often possible to put together a short list of specific "problems" that can be selected in the software. This can lead to efficient software use and uniform coding which facilitates analysis. Any coding system should be carefully evaluated for use in the specific environment.

2.1.21.Referrals (counseling, ancillary care)

Referral for specialized treatment tracking, including results of referral.

Since HIV infection is a long term disease which requires a variety of responses over the duration, a system that can manage referrals for counseling or ancillary care can be very useful. A good HIV management program will include a wide variety of services including counseling, contact follow-up, family support services, specialist medical services and others. It can be difficult to track and manage these services without the use of software. Software that has the capability of scheduling, tracking, and following up on these additional services will improve the patient's outcome. The software should include the ability to classify the category of service, the response, responsible agencies or individuals, and outcome.

2.1.22.Consultations (medical)

Medical consultations with results and recommendations.

As HIV progresses, various specialist medical consultations may be required which go beyond the primary clinical caregiver. Often these consultations may be conducted at a time and place distant from the primary care site. It is important to schedule and track the results of these consultations with recommendations so that the patient can receive the best care. Software that can track these referrals and permit recording of recommendations and follow-up can improve patient care. This software feature may be included in the "Referrals" section (above) or the software may include a separate section for medical consultations.

2.1.23.Encounters (LMR)

Patient encounters (longitudinal medical record) including dates, reason for visit, exam, diagnosis, treatment, services, medications, lab, x-ray, follow-up.

Patient encounter tracking is the essential core of the longitudinal medical record (LMR) and is essential for chronic disease management. Each patient encounter must be recorded and include information on activities that were conducted at the time. Encounters can include brief "nurse visits" where basic information is collected and medications refilled as well as more intensive clinician visits where more information is collected and analyzed. Together, these individual encounter records constitute an important clinical record of the patients care and treatment and response to treatment over time. Software that collects this information and presents it for review can greatly improve patient care by giving clinicians a good picture of the patient's history and response to treatment. The encounter information is also an important source of information to monitor the performance of the health system and can be used to improve patient care protocols.

Software should be flexible and customizable to include the information that is important for protocol and program. It should be easy to use so that data entry does not become a burden. It should use coded values wherever possible to improve the reliability of the data.

Many systems are using the WHO Patient Monitoring Guidelines as an essential data set for ART. Often this essential data set is modified to accommodate regional protocols and programs.

2.1.24.Vital signs

Vital signs such as temperature, pulse, BP, and weight for each encounter.

The basic set of vital signs are important information for monitoring patient status. The software should include the capability to easily record this information at each encounter. Vitals such as weight can be presented as a graph which some people find useful in monitoring initial response to ART.

2.1.25.Lab

Lab ordered and results, provision for review of results over time and comparison of multiple lab values. Does the system produce graphs of results?

There are several laboratory tests which can be important for monitoring patients with HIV infection. It is important that these be recorded in the record and be available for easy review. Often laboratory results lend themselves to graphical presentation which improves their readability by making it easier to see trends. Software should record the date of each laboratory test and the results. Advanced software can also flag abnormal values or trends for the attention of the clinician.

2.1.26.Lab results interpretation

Automatic checking of lab results for abnormal values or combinations of values.

Information that is entered into computer software has the advantage of being available for analysis and decision making. Simple computer analysis can include checking laboratory values for abnormal results. More advanced analysis can include trend analysis or checking multiple values for coherence or abnormalities. In advanced systems, software analysis can look at related clinical information and use this to improve the analysis.

When selecting software, check how easy it is to configure this analysis software. Changes will need to be made to the parameters and rules for analysis, depending on regional or local protocols and lab methods therefore the software should be easy to configure to meet changing requirements.

2.1.27.Lab electronic linkage

Provision to load external lab results to EMR automatically.

Often laboratory results are available in electronic form. If these results can be set up to be automatically transferred into the clinical software, it will improve the speed, accuracy, and completeness of laboratory information. This linkage must be carefully designed and implemented with proper protocols to ensure unique patient identification and reliable communication of information. Software that offers this feature can improve information system.

2.1.28.Radiology

Radiology ordering and results; text results; image results.

Radiology results are a special case of information that should be integrated into the patient record. Software to record this should record the date, type radiology exam, and results in text format, this is minimum requirement. More sophisticated software can store radiology images for retrieval. However, this requires a large amount of storage space and can place a significant technical hurdle to implementation. For most purposes, the basic radiology information (without images) is sufficient.

2.1.29.Medication

Record of medication; dates start and end or duration; graphing of medications. Drug interactions; drug dosing; standard treatment regimens.

Medication tracking is crucial to managing ARV patients. The treatment of AIDS requires the prescribing and monitoring of a complex combination of drugs according to a specific protocol. Current therapy decisions depend on prior medication history, response, adverse reactions, and resistance.

Software should be able to track medication by individual drug and drug regimen (combination) including dates and dosage. Particular attention should be made for ease of entry of this complex information as well as viewing of medication history. Graphs of medication regimens, individual drugs, and gaps in treatment can be very useful.

Drug regimes are an important concept in ART. A drug regimen is defined as a specific combination of drugs (typically three drugs). Patients will typically start with a certain regimen and this may change over time as the patient develops adverse reactions or resistance.

Software must include the ability to define standard drug regimens and to allow execptions and substitutions. As protocols will change over time, these regimens must also be easily changeable. In order to view individual patient drugs, the regimen information should include the names and dosages of each of the individual drugs in the regimen. This information is important as it also permits later analysis of individual drug response.

When selecting software, particular attention should be paid to the flexibility of the drug regimen features to ensure that they can be easily customized to regional or program needs. In addition, the ease drug data entry is important since this is a function that will be performed frequently. If this data entry procedure is complex, confusing, or difficult, the data quality will suffer.

2.1.30.Other

Any other clinical features of note.

Software designers are endlessly inventive and many have developed useful features to improve the capture and display of information. Evaluate these features carefully in your use environment to gauge their usefulness.

2.1.31.Individual Patient Reporting

The type of individual patient reporting is available for use in the clinic. This may include a patient summary sheet or more complete report. Is it user configurable?

In the above discussion, reference has been made repeatedly to a “patient summary sheet” or to display individual patient information. It is important to remember that the primary purpose of collecting this information is to use it and is of most use is at the point of care of individual patient. The point of care is the place where the clinician makes decisions on patient care and is crucial that the clinician have complete and accurate information.

Software that can present individual patient information in a clear, convenient and useful format will improve patient care. This may be on a computer screen (if these are available for patient interaction) or more likely in printed paper format that can be carried to the patient at the time of the encounter. It is extremely important that this individual patient reporting be evaluated carefully to ensure that it meets regional and program needs and protocols for good patient care. One should also evaluate the ease of modifying the content and format of this information to accommodate changes as clinical care evolves.

2.2.Guidelines for care

2.2.1.Standard guidelines

Provision to include standard care and treatment guidelines in software and to have the system prompt this information.

Access to standard guidelines in the context of individual patient care can improve compliance with the guidelines and patient outcome. This is a simple advice system but it can bring about improved patient care.

When evaluating a guideline advice system, one should consider when and how the information is presented. If the system produces a printed patient summary sheet that is used at the point of care, the guideline information should also be printed on the the patient summary sheet.

One should also consider the ease of modifying and updating this information as treatment guidelines change.

2.2.2.Customized patient advice

Software makes an assessment of patient condition and current treatment and makes recommendations.

One of the advantages in having an electronic medical record is that the information in the record can be analyzed automatically and used to give treatment advice. This is a form of "expert system". The treatment advice can be something as simple as showing the standard treatment protocol or it can be more a more complex system where the computer evaluates individual patient information and tailors advice to the specific condition.

This is an advanced software function and may not be available in most software. If it is present, it must be capable of being modified to meet the specifics of regional treatment protocols. The system should be investigated to determine the ease of adding and modifying "rules" for treatment advice. This function can improve the quality of care but it will require an effort to set up and maintain.

2.2.3.Patient take home information

Information for patients to take home regarding their diagnosis, care and treatment.

Another advantage of the electronic medical record is that it can analyze an individual patient record and produce patient advice that is tailored to the specific status of that patient. This "individualized patient advice" can improve patient knowledge and cooperation and improve their care. This information should be reviewed with the patient at the clinical encounter and then given to the patient to take home as a reminder of the advice.

The amount of customization and flexibility of this patient advice system will vary with the software and should be evaluated carefully for applicability within the regional context. An important factor to consider is how easily the information can be customized to meet local requirements and how easy it is to customize the criteria for distributing custom advice.

2.2.4.Decision support / expert systems

Describe any decision support or expert system features in the software including the degree of user configuration possible.

A decision support system is an advance expert system that can draw information from the patient record, apply rules that have been designed by experts, and give specific patient advice. This is a step more advanced than the customized patient advice system. In evaluating this feature, it is important to gauge the ease of customizing the analysis to meet local requirements.

2.3.Clinic Management

2.3.1.Appointment scheduling

Group scheduling of rooms, practitioners, and patients.

In order to improve the efficiency of service delivery, it is helpful to distribute the workload to match the human and physical resources. A computer scheduling system can help with this task. It makes it easy to gauge workload per clinician and allocate facilities. It can monitor overbooking as well as underutilization.

When evaluating this feature, carefully assess the ease of making an appointment as well as the tools that monitor resource utilization.

2.3.2.Service statistics

Clinic service statistics for management.

Basic clinic service statistics are an essential management tool to improve the efficiency of delivery of services, gauge productivity, and advocate for appropriate resources. Electronic medical record software should be able to produce this information from the encounters that are entered. This is an example of "free" information that comes as a side benefit of maintaining an electronic system. Evaluate this feature by the flexibility and customization available in the reports.

2.4.Clinic attendance and services patient register

2.4.1.Patient register

Patient register and the information that is recorded.

Electronic medical record software should be able to produce a "register" of patients. The register is a master patient list. The patient register should be able to be filtered by any of the information (such as patient status, demographics, clinical status, clinician, etc.) to produce sub-registers which are useful for administrative and program monitoring and evaluation. Evaluate the flexibility of the filtering and sorting capability of the patient register function to ensure it meets requirements for local use.

2.4.2.Summary register

Any summary registers that are produced.

Summary registers are useful for program monitoring and reporting on activities. They should be able to be configured and produced automatically from the information that is collected by the EMR. Assess the flexibility and customization features of the summary register.

2.4.3.PMTCT information

PMTCT indicator for specific information.

Prevention of mother to child transmission (PMTCT) programs are important and significant in reducing HIV infection. These programs have unique indicators for monitoring their effectiveness. Evaluate the software to ensure that it tracks and can report on the necessary PMTCT indicators that are necessary for local program management as well as project reporting.

2.4.4.Referrals & counseling

Patient referrals for testing, extended care, and counseling.

Because of the complexity of HIV care, it requires a broad, multi-provider approach. However, it is difficult to monitor these complex interactions. The EMR software should capture referrals for testing, counseling, and extended care. Evaluate the software on how well it tracks and reports on these referrals since this is an important function in the success of the program. The software should include features to categorize, sort and filter the reports to show all aspects of interventions.

2.4.5.Notifiable conditions monitoring

Notifiable diseases reporting and tracking.

An important function is reporting of notifiable diseases for epidemiological tracking. The advantage of the EMR is that it can produce these reports from information that is routinely captured during patient encounters. Evaluate the software to ensure that it captures the information that is needed and can produce the appropriate reports. Electronic reporting of this information can improve accuracy and timeliness.

2.5.Logistics (drug and equipment inventory and supply)

Drug, supplies, and equipment logistics are complex functions that require their own systems. However, the EMR can provide valuable information to inform the logistics system. Some EMR systems have even been extended to include elaborate logistics functions.

2.5.1.Capabilities

Order, inventory, distribution, expiration dates.

The EMR can provide information on actual use of drugs and supplies as a side benefit of routine drug and test ordering. Simple input to the logistics system of "demand" can improve logistics. More sophisticated software can track actual ordering, inventory, and distribution of drugs and supplies including expiration dates and lot numbers. These software functions should be evaluated in the context of the overall logistics system.

2.5.2.Integration with patient care

Describe how logistics is integrated with patient care.

In more sophisticated systems, logistics information can be generated from patient care module. Demand for drugs and supplies can be tracked by patient care activities. It is important to evaluate the information produced by the EMR system in the context of the logistics system.

2.5.3.Drug Regimen support

A drug regimen is a group of drugs given together. Drug regimen support includes the ability to define and prescribe regimens and also retain the detail of the drugs in the regimen.

The concept of a "drug regimen" is central to ART. Patients are started on a particular drug regimen (typically three drugs) and there may be substitutions and changes of regimen during the course of treatment. In the clinical context, it is important the the software be able to maintain drug regimens and track their use with individual patients. This information also has important uses in logistics as it can produce reports on which regimens (and drugs) are being used. The information can also be used to forecast demand by analyzing factors such as how long new patients are maintained on each regimen before requiring a substitution or change of regimen.

2.6.Reporting

Reporting is essential to patient, facility, and program management. Reporting in the EMR takes advantage of the routine collection of electronic information. Since information is in electronic format, it can be easily filtered, sorted, and aggregated to produce reports that give insight to the operation of the system. Reporting is an important software function and the software should be carefully evaluated for the type and variety of reports that are produced as well as the flexibility and ease of customization.

2.6.1.Summary reports (time, place status)

Type and flexibility of summary reports for facility.

Summary reports are the basic level of reporting for a facility. They produce information on counts of activities and actors that are very useful for management and program monitoring. The software should be evaluated for the type and variety as well as flexibility and customization of summary reports.

2.6.2.Aggregate Reporting (facility, district, national)

Ability to aggregate data from multiple sites into district and higher level reporting (with sub-totals) .

While individual summary reports are useful for evaluating and managing facilities and individual activities, they are typically aggregated at higher management levels to produce information on broader areas of activity. Electronic system can be designed to improve the speed and accuracy of this reporting by automatically aggregating lower level information. However, this function must be properly designed to transmit and receive electronic information and produce aggregate reports. The software should be carefully evaluated to gauge the ease of producing information for aggregation, transmitting this information to a location for aggregation, and finally for aggregating the information and producing the reports. If all of these steps are done electronically with appropriate protocols, aggregate reporting can be fast and accurate. The software should also be evaluated for its aggregate reporting capabilities. This especially includes the filtering and sorting capabilities as well as the ability to "drill down" into data to explore it in detail.

2.6.3.Monitoring and Evaluation functions (facility, district, national)

Ability to calculate indicators for specific programs. Indicate which programs are supported.

There are many projects that support HIV care and each of these has unique requirements for reporting.

Preparing these reports can place a significant burden on personnel. The information in the EMR can be used to produce these reports with much less effort. The software should be carefully evaluated to ensure that it tracks the necessary information required by programs and that proper reports can be produced automatically.

2.6.4.Epi investigation

Specific reports to look at communicable disease patterns and outbreaks.

Epidemiological investigation is an important function that can be improved through the use of electronic information. The side benefit of collecting routine patient information is that this information is available for analysis of the spread and characteristics of the epidemic. This is important for monitoring and planning activities.

The software should be evaluated to ensure that it records and tracks important epidemiological indicators and that there is a flexible reporting system for analysis of this information.

2.6.5.Research capabilities

Ability to access the database for ad hoc queries. Indicate method of query such as SQL.

The EMR information is a valuable database on the characteristics of the epidemic, the response to treatment, and health system activities. As such is can be an important research tool to provide information on best practices for treatment and management of care activities. Researchers tend to want to perform “ad hoc” queries on data to analyze it as questions come to mind. If the underlying database EMR uses standard query method such as SQL, researchers can make valuable use of the data. Some software also provides useful flexible query tools to improve visualization of the data. Evaluate the software for these features. At a minimum, the database should provide SQL capability.

2.6.6.Reports to funders (including PEPFAR indicators)

Ability to produce specific reports to support funded programs.

Many programs are funded by PEPFAR which has specific reporting requirements. The EMR should be able to produce these reports automatically. Evaluate the software for the ability to collect and aggregate data and produce these essential reports.

Reports to other specific funding programs may also be necessary and the software should be evaluated for the ability to produce any required reports.

2.6.7.Custom reporting

Describe extent to which customization of reports is possible.

Program management and funding reporting often have changing requirements. Evaluate the software on its ability to produce custom reports to meet new requirements. A typical “report generator” allows selection of data fields, aggregation, and indicator calculation through a simple user interface. The new report should then be able to be saved and run again in the future, possibly changing some of the parameters such as dates.

2.6.8.HL7ART data transfer protocol

Support for the WHO HL7ART data exchange protocol.

The WHO is supporting development of a standard for individual patient record transfer. This transfer is useful for moving patient records from one location to another when a patient moves and is important to maintain continuity of care. The other use of this protocol is to send (anonymized) patient records for monitoring of care and research. The protocol uses the HL7 health data and XML data interchange standards.

Evaluate the software to determine whether or not it supports (or can be upgraded to support) this HL7ART data interchange protocol.

2.7.Security and Data

2.7.1.Backup tools

Data backup tools that are included other than standard computer backup utilities.

Electronic data is robust because it can be easily copied and transported and can be accessed from multiple locations. However, it is vulnerable to fire, flood, theft, and computer malfunction if it is not backedup and stored at multiple locations. Most computer systems have standard tools to make backups of important data that can then be moved to a safe location. Some EMR systems also include tools to make this process easier and more specific to their data. Evaluate the software backup tools to ensure that you can easily make backups of your valuable data and restore it when necessary.

2.7.2.Security

Security of access to the data in the computer and during transport. Indicate if data encryption is supported.

The EMR system will contain valuable information that is personal and may be subject to privacy regulations. The software should have a method of restricting access to data only to authorized persons. There are many protection schemes and degrees of access. Evaluate the software carefully to ensure that it meets legal and ethical requirements for access control.

It is important to ensuring that only authorized users have access to highly sensitive patient information. Systems have been categorized at the three levels of security viz:

Level 1: No security or security is a simple system level password.

Level 2: Security is a simple password system.

Level 3: Includes user level access and may also include advance features such as encryption and user identification of modifications to the system.

Data can be encrypted for storage and transport where is may be vulnerable to unauthorized access. Software should be evaluated to assess its encryption capabilities.

2.7.3.Data transfer

Ability to transport data to another system for analysis. Describe data and process.

The WHO HL7ART data format is one component of data transfer. Software may use this or it's own proprietary format for data transfer. The process of data transfer goes beyond the specific format of the data during the transfer and includes procedures for extracting the data from one system, sending it to another, and loading the data into the second system. It should also include the protocol to resolve replication issues where you have multiple copies of the data being used in different locations.

2.7.4.Data validation/quality checking

Type and degree of data validation and quality checking built into the software. Is this done as batch or real time?

Electronic systems have an advantage over paper in that the data can be checked for validity. There are a number of types of checking that can take place to improve the quality of the data. These include range checking, entry against a table of valid entries, completeness of data, and consistency checking against multiple factors (i.e. a pregnant patient must be a female). Data can be checked at the time it is entered which in most cases is preferable to a batch checking process that is performed at some later time when it may be more difficult to correct the errors.

Software data validation is an important function and the software should be evaluated carefully to assess the type and flexibility of the data checking that is performed. Data checking must be user configurable to accommodate local requirements.

Software products have built in checks to ensure data quality. Some checks occur at the time of data entry and are based on logic rules (e.g. males can be listed as pregnant) or legal values (e.g. ages of patients must be below 100). Well-designed systems minimize free text fields and provide drop down lists that can be modified by specific data managers. Some systems also provide checks to detect duplicate patient records (e.g. based on algorithms of name and personal identifiers.)

Three categories of data validation are described below:

Level 1: Minimal or no data checking provided.

Level 2: Some field level checking provided such as acceptable numeric or date ranges.

Level 3: More sophisticated checking provide that can involve a combination of fields and values using a programmable rule set.

2.7.5.Tools for paper input (i.e. Forms generator)

Are there paper forms that correspond to the data in the computer system for use when a computer is not available. Do forms correspond to data screens? Are paper forms supported for input?

It will be rare to have a complete electronic medical record. The resources required to install computers and associated support infrastructure at every point of care are too great. The software should include the ability to produce paper forms suitable for patient interaction at the point of care. These forms are typically used to provide patient information (summary sheet of history and current status) as well as to collect information for later entry into the computer.

The forms should support a data flow process where the paper form is first printed and contains the necessary information to enable clinician decision making at the point of care. Then the form is used to collect information from the current encounter and finally it is used for data entry into the computer. This form must be capable of being customized easily to ensure that it meets changing requirements. The data input part of the form should be designed to facilitate rapid and clear data entry. The best way to do this is to have the data input screens correspond closely to the organization of the data on the form.

This is an important function for the software and it should be thoroughly evaluated in the specific clinical context to ensure that it works smoothly and facilitates the patient encounter.

2.8.System Information

2.8.1.Software architecture, platform (language, database, OS, server)

Software operating system, program development language, server (if used).

The specific software architecture including operating system, program development language and server are important considerations since they affect cost, deployment, maintenance, and support. This ART Inventory document has a complete discussion of the various architectures to assist in evaluating this information.

2.8.2.Data communication architecture (LAN, WAN/Internet, data import and export)

Types of data communication supported including protocols and modes (wired, wireless, IR, etc).

One of the great advantages of electronic information is that it can be easily transmitted. Software can support various methods of data transmission. Individual implementations will take advantage of various data communication methods. The software should be evaluated to ensure that it supports the intended communication methods. This ART Inventory contains a detailed discussion of data transmission methods to assist in evaluating this information.

2.8.3.Scalable

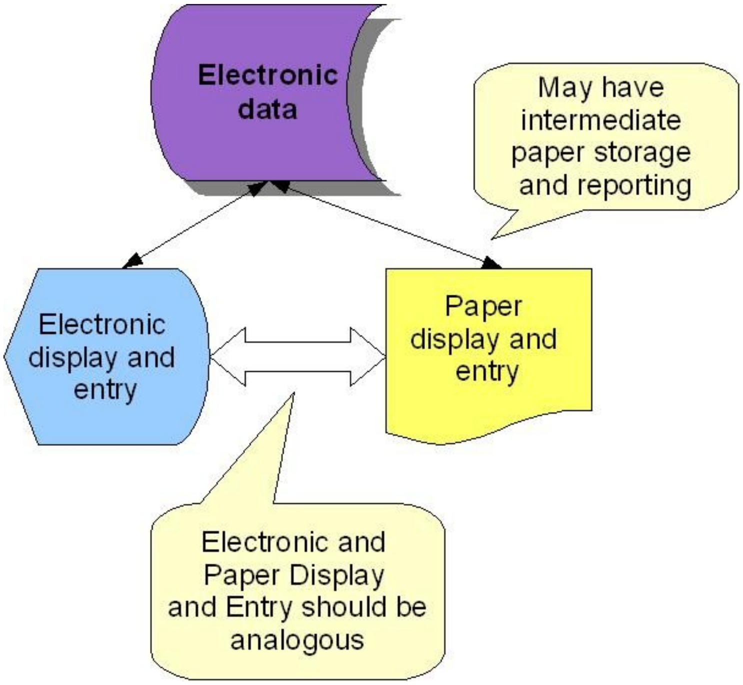
Can the system scale to large sites, multiple sites, district and national level? How is wide area network communication maintained?

Scaling refers to the ability of software to work on different sizes and configurations of computer systems. A small clinic may have a single computer. It may grow to a local area network (LAN) with several compters. It may then add another clinic in a nearby location which needs to be connected through a wide area network (WAN). Groups of clinics may become nationwide and may handle thousands of patients. Software that scales well will be able to grow to handle all of these situations.

Software design, architecture and platform determines how well it scales. The issue of scaling is discussed in this ART Inventory in the section on platforms and architecture. This information will help you understand these issues and evaluate software.

2.8.4.Data flow map (paper – recording – transmission – M&E – action)

Describe data flow from collection to entry to communication.


Electronic data
May have intermediate paper storage and reporting
Electronic display and entry
Paper display and entry
Electronic and Paper Display and Entry should be analogous

As described above under “Tools for Paper Input”, it is important that the software support an efficient work flow. Most electronic systems will interface with paper at some point, most commonly at the point of care. It is important that the software be designed to support paper point of care forms which are flexible and include information such as patient clinical and demographic information summary and which include a method for recording new information. This new information should then be entered into the computer using screens that are designed for ease of input. Good work flow also includes the generation of alerts and messages based on expert advice systems. Information should be automatically sent to where it can be acted upon. In addition, the information should be used by individual clinicians for patient care and for program monitoring.

Electronic information systems offer many advantages in automated analysis and communication and the computer systems should be designed to take advantage of electronic information. Evaluate the software carefully to assess how well the software automates the process of putting information into action.

2.8.5.User modification at site

Is it possible to customize the software at the user site (non-programmer). Describe fields or lists that are customizable.

Software is by it's nature modifiable. Good software allows users to modify some of the options for local use. One attractive feature is the ability to have user definable fields which allow the software to be easily modified to capture information customized to local needs. The reality of information systems is that they must adapt to meet local information needs. ART clinics are not a homogenous environment and individual programs and sites have unique needs.

Evaluate the software to assess its ability to capture local information - the ease to customize data fields for local use. One should also look for the ability to structure this locally defined data by designating the data type (text, numeric, date), specify ranges, or pick values from a user defined list. The software should also include the ability to analyze this information and include it in custom reports.

2.8.6.Ease of modification (local or international, and skills required)

What level of effort and skills are required to modify the software.

As noted above, individual countries, regions, and programs will have unique information requirements. Any individual software package cannot meet the needs of all programs. In addition, all software requires maintenance to correct errors in programming and to update it to meet changing requirements. Fortunately, software can be modified. However, the ease and methods of modification will vary quite a bit from one software package to another. The factors which determine how easily the software can be modified include the language, architecture, data design, software design, platform, availability of source code, local programming resources, and international programming resources.

One of the most important factors is the availability of the source code. If you have access to the source code and have permission to modify it, you can develop local resources to customize and maintain the software. If there are good local resources for software maintenance, it will generally provide better and more efficient service. This gives better control of the software system and data.

Evaluate the software carefully to understand the terms of the license agreement. All software has a license which specifies the terms and conditions for use and access to the source code.

Software distributors offer different licenses and these should be evaluated carefully since it will affect the ability to use and maintain the software. Some distributors may let you view the source code but not make changes, others may let you make some changes but place restrictions on the use of the changed software. If you must rely on an outside software distributor to make changes to the software, they should be carefully evaluated to ensure that they will be responsive to your needs. The license agreement should also specify what will happen if the distributor goes out of business or cannot support the software. In this case, there should be a provision for you to obtain the source code (such as an escrow agreement) so that you can continue to maintain and use the software.

The best access to source code and use license is what is known as “free open source software (FOSS)”. This type of license places the fewest restrictions on your use and modification of the software. FOSS licenses such as the GNU General Public License (GPL) allow this use, to make copies and modify the software without restriction. The GPL permits the freedom to run the program, for any purpose; the freedom to study how the program works, and modify it; the freedom to redistribute copies; and the freedom to improve the program, and release the improvements to the public.

The important issue here is that since you will be making a large investment in software, equipment, training, and data entry, there must be an assurance that you will be entitled to the use of the software to access data. This requires that you have a reliable software support system and have the option to develop your own support in the event of failure of any external support.

2.8.7.Source code available

Is the source code available for modification? What is the policy on compiling and distributing enhancements?

Source code refers to the high level programming language instructions that are compiled into the software program that you run on your computer. In order to make any changes to the program, you must have access to the source code. Access to the source code is the ultimate guarantee that you will be able to fix problems and modify the software to meet your needs.

As discussed above, software is distributed under different licenses, many of which do not give access to the source code. If you do not have access to the source code, you must carefully evaluate the software developer and distributor to ensure that they will be able to continue to support your software installation without interruption.

2.8.8.System documentation available

Is there system documentation available for use by programmers?

Development of software systems is complex and typically includes specifications and software documentation that describe the functional requirements, data design, and specific software design. These documents can be very useful to programmers when modifying the software.

The system documentation should be evaluated carefully to ensure that it provides good information for programmers to modify the software.

2.8.9.User interface evaluation

Type of user interface; assess ease of use.

The user interface refers to the visual display and methods of interaction with the software. It is a complex subject and there are accepted principles of good user interface design. A good user interface will be clear, easy to understand, flow logically and be intuitive. The investment in software, training, and data entry will be significant and a good user interface can make it easier to enter and access information. It can also improve the ease of training individuals. Most modern computer systems use a "graphical user interface" (GUI) which improves data layout and presentation and permits navigation with a mouse. This is generally considered a superior user interface.

The software user interface should be evaluated carefully to gauge the ease of use and training. Screens should be clear and understandable, navigation from one screen to the next logical screen should be easy. The usability of the computer system should be tested by "naive" users as a gauge ease of learning and intuitive factors of the interface.

2.8.10.Web interface

Web interface available for use of software on a WAN

The Internet is a powerful communications tool for several reasons. If offers both a standard user interface and standard communication method. These are universally understood. The web interface offers information access over a wide area network (WAN). Because of universal support, a lot of software uses a web interface even for local installation. ART systems by their nature will need to be installed in local and wide area networks. Software should be evaluated to assess if it includes a web interface for wide area network access. This may be the same interface as the one used for local access. If they are different, the software should be evaluated carefully to assess any differences in the user interfaces and ease of use. Differences could affect training and daily use of the software.

2.8.11.Documentation available (language)

End user documentation availablity and language(s)

If the system comes with good end user documentation and training materials, it will be easier to train people and have them using the software efficiently. The software end user documentation should be assessed for completeness and usability. End user documentation should cover all of the functions of the software organized in a logical reference book format. It should be easy to quickly locate clear, complete information on any aspect of the software that an end user might need explained. End user documentation evaluation should also include an assessment of any language needs and translations required.

2.8.12.Training material available

Training material availablity and language(s)

All individuals using the software will need to be trained. If the system comes with good end user documentation and training materials, it will be easier to train people and have them using the software efficiently. Training materials are specific guides used to train individuals to use the software. There may be more than one training guide developed for various users of the system such as data entry, reporting, and system administration. A good training guide presents information clearly in logical sequence organized around specific tasks. Training guides for the software should be evaluated for suitability in your specific environment and for your anticipated classes of software users. Language support and translations should be evaluated for suitability.

2.8.13.Installed base (# users and duration)

The installed base of users, numbers of patients, and the duration of time

Information on the number of installations and patients registered in the system can help in evaluating the maturity of the software as well as its acceptance. While not a guarantee, software that is widely installed and used with large numbers of patients has generally matured to the point where it is reliable, scalable, and addressed data, user interface and training issues. It also indicates a potential source of support and help with modifications.

2.8.14.Languages supported

Languages supported

Language support will be an important consideration. Clearly, the software must support the language used by the majority of people who will be interacting with the software. Often a country may have several languages that must be supported.

There is a software architecture factor that can make it easy for software to support multiple languages. If the software is designed with all of the display text stored in a data table rather than embedded in the source code, it is much easier for the software to support multiple languages. If all of the display text is in a separate table, it is relatively easy to add a new column to the table and translate the text messages into the new language. The software can then support multiple languages just by specifying which language column of the table will be used. If the language requires a special character set, the software should be capable of supporting the "Unicode" standard.

2.8.15.Formal support (type of organization)

Organization supporting the software and their level of funding and stability

The software will require continuous maintenance and support to fix problems and stay up to date with required changes. Before investing in software, the supporting organization should be carefully evaluated. The organization that will support the software should be prepared to assure that it will be financially and organizationally stable over long term. The availability of source code is an important factor to consider in the event that the supporting organization is not able to continue. The supporting organization may be non-profit NGO, for-profit business or government agency. Each of these types of organizations has strengths and weaknesses but none is immune from failure to support the software. Each should be evaluated carefully.

2.8.16.User community

A community of users available to provide additional support

A community of existing users can be an important resource for support with installation, training and maintenance issues. This is an additional source of information that can supplement support from the developer or distributor of the software. Communities of users (particularly for open source software) can be very active and helpful.

2.8.17.Sponsoring organization

The parent organization that developed the software

Often the software has been developed under the direction of a sponsoring organization which will have an interest in ensuring continuous support for the software. This is another factor to consider when evaluating systems.

2.8.18.Funding source

Funding source for the software

The source of funding for the software is an important factor to consider when evaluating the stability and longevity of the software. An organization that is well funded is more likely to continue support in the face of adversity. Large organizations can be more stable in funding. Governments can have access to stable funding but are also subject to political influences.

Open source software is a special case. Open source software can be developed by large stable organizations but it is often supported by a community of developers and users. Individually, each of these may not be large but the community as a whole can provide much more stability and continuity than any single organization. Of course, with open source, there is always assurance that as a last resort, you will have access to the source code and can support yourselves.

2.8.19.Development team size

Development team size and availability for customization

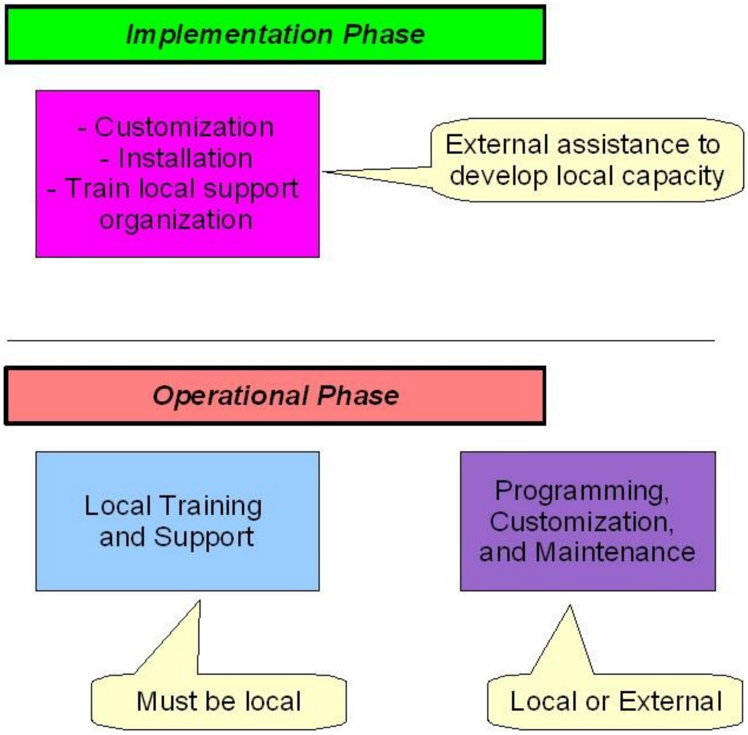
The number of programmers developing the software is an indication of the support available for customization and maintenance. This should be considered in relation to the number of sites that these developers are supporting. Open source software will often have a large number of developers, each devoted to a particular installation but also contributing to the overall project.

2.8.20.Support team size

Support team size and availability

Local development, support, and maintenance of software application is always preferable. This has implications for software selection, as well as staff and training for routine management and maintenance of a data system. This implies that software be open source, programmed in a language commonly used, a product used by other facilities in the region or country, and allows exported data files to be transformed into formats familiar to local analysts.

At least during the initial installation and training, outside support may be required. Often, continuing support can be contracted. The number of people available for this external support should be evaluated in light of the number of installations that the external organization is supporting.


Implementation Phase
- Customization
- Installation
- Train local support organization
External assistance to develop local capacity
Operational Phase
Local Training and Support
Programming, Customization, and Maintenance
Must be local
Local or External

2.9.Hardware Requirements

This information is intended to help understand the technical requirements and costs to install a typical system. The categories of system are generalizations and will change as technology advances. The important factor here is that if the software can run on older or less technically demanding hardware, there is the possibility of reusing existing hardware or purchasing less expensive hardware.

Use this category system: A=current state of the art (2+ GHz processor, 512 MB memory, 20+ GB HD); B=one generation old (1 GHz processor, 128 MB memory, 10+ GB HD); C=two generations old (200 MHz processor, 64 MBmemory, 1 GB HD)

2.9.1.Client hardware requirements (speed, memory)

Minimum hardware requirements for client (workstation) computer

Client workstations will be the most numerous in the system. If the software is capable of running in a web browser or uses a thin client software, then the hardware will cost less since older or less expensive computers can be purchased.

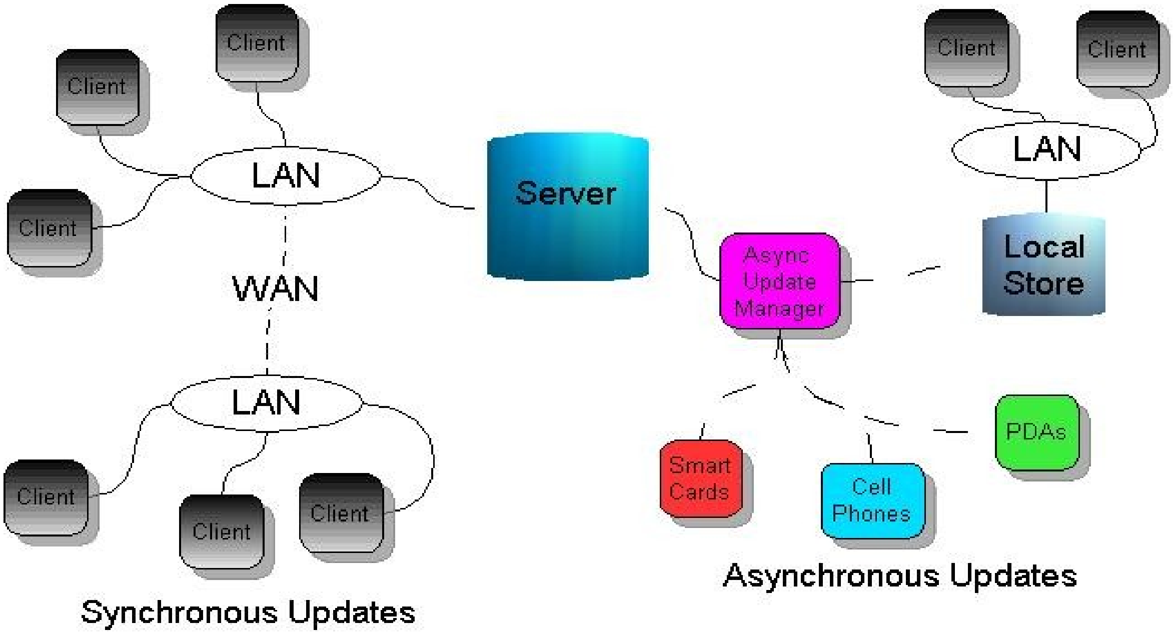
2.9.2.Server hardware requirements

Type of hardware is required for the server

Servers typically will have more demanding hardware requirements because they must store all of the data plus perform whatever operations the client demands. It is usually best not to try to save money by purchasing lower performance server hardware. There will typically only be one server per site.

2.9.3.Communications requirements (local area and wide area)

LAN (ethernet), WAN (indicate protocol), wireless (WiFi or other).



Most local area networks run using the Internet TCP/IP protocol over Ethernet and this hardware is commonly available and inexpensive. Wide area networks (WAN) also usually run the Internet TCP/IP protocol. However, there are several options for communications including WiFi, WiMax, fixed-line (dial-up modem, DSL, ISDN, T1 in increasing order of speed), and wireless SMS or GPRS. Make sure that the bandwidth offered by the communications method of choice is sufficient for the software requirements. There are significant cost differences among the different communication media and should be assessed for long term cost.

2.10.Software Requirements

The ART application software will typically run on top of an operating system and a database and may also require additional software such as language interpreters of separate packages for reporting. Each of these software packages has a cost to purchase, install and maintain. Evaluate the cost of of the software as a whole.

2.10.1.Operating systems for client and server

The most common operating system is Microsoft Windows. The current version is XP which comes in a “Home” and a “Professional” version. The Professional version is significantly more expensive and it may be required by the ART application you choose.

Linux is a popular alternative operating system and is noted to be very reliable. It is commonly used with servers and can also be used as a client operating system. Linux is free open source software.

2.10.2.Database

Most applications run on top of database software. Database software can be very expensive. Oracle and Microsoft SQL Server are two databases that are commonly used and both have significant costs for purchase and maintenance. Some applications can use a “runtime” version of Microsoft's SQL Server which is available at low cost.

There are two popular free open source databases, MySQL and PostgreSQL. Many people feel that the performance and reliability of these databases are better than Microsoft's SQL Server.

You may not have a choice of database software. Some ART applications only run with one particular database so be sure to include the cost of the database in your calculations.

A software platform consists of a computer operating system as the foundation and application software (computer “programs”). All health information systems require some type of “database” which is specialized software optimized to store and retrieve data.

Most of the applications evaluated run on either Microsoft Windows (proprietary) and/or the LINUX operating system (FOSS.) Application software can be written in various languages and must be compiled to run on different operating systems. The exceptions are “web based” applications which use a standard web browser and do not require any software to be installed on the client computer. These programs will run on any operating system.The actual language is not important except when the option of receiving the “source code” to modify the program is available. The applications evaluated are built using either a primarily client-server database software (SQL Server, MySQL, and Oracle) or a primarily desktop database application (such as Microsoft Access). These platforms each have strengths and weaknesses as detailed below.

Structured Query Language (SQL) is a database standard query language that allows a common set of commands to manipulate data in a database. Most Data Base Management Systems are so called “relational” databases that follow the SQL standard. The main reason is that relational databases fit the commercial and/or corporate environment well, and it is particularly well suited for large sets of linked data. Some common SQL databases are “SQL Server”, “Oracle”, “MySQL” (FOSS), “PostgreSQL” (FOSS), “MS Access”, and “DB2”.

Some health applications also use either hierarchical databases (for instance “Fileman” from WorldVista) or object-oriented databases (like Cache). These database types can be better suited for data related to complex objects like a human being and/or they might retrieve medical data faster than relational databases. However, the sheer commercial “weight” of relational databases gives them prominence in the marketplace. This dominance means that it is considerably easier to find local technical expertise for the development and maintenance of relational systems.

Microsoft Access is a user-friendly desktop Data Base Management System that has the added advantage of being an integrated part of Microsoft Office. A significant advantage of MS Office/Access is that much existing computer literacy in developing countries is linked to knowledge about MS Office, in particular Word, Excel, and PowerPoint. The Health Information Systems Program (HISP), which had developed and implemented the District Health Information Software (DHIS) in many different developing countries, unequivocally points to this fact as their primary reason for initially choosing MS Office (Access) as their main development platform – existing computer skills were regarded as more important than technical excellence. HISP is, on the other hand, today moving towards more client-server type software.

1.1.1.1 Microsoft Access

Access runs only on the Windows operating system. Each computer has to run its own Access application using a copy of Access that is installed on the machine. Access-based applications can have the database on the user's computer, or on a server (connected either through a LAN or a WAN). Access application can be written for multi-user environment where several users access a single database. However, multi-user Access requires special skills and extra programming effort, so not all Access applications work in a multi-user environment.

Advantages:

- Many people understand and use Access (and Excel, Word, and other aspects of Office).
- Single user freestanding applications are in many cases easier to create than multi-user client-server applications

Disadvantages:

- MS Access, or to be more specifically its Jet Database Engine, is designed for use in a smaller environment, with a maximum database file size of 2 Gigabytes (Access 2000/XP/2003). Performance degradation of MS Access as database size increase is often exaggerated, but HISP recommends archiving older data when the number of records goes beyond 600,000 to 800,000. Assuming that an EPR initially comprises 20 data records with an additional 10 "visit" records being added per year, it makes little sense to use Access if you deal with more than 10,000 to 20,000 patients.
- MS Access applications cannot be easily scaled as long as the data is stored in a Jet database file. However, it is relatively easy to export Access Data from the Jet database engine to the "Micro Soft Database Engine" (MSDE). MSDE is the database engine in SQL Server and supports database files of unlimited size. In other words, the application's "front end" (what the user sees) can be in Access but the data is stored and "served" from an MSDE database.
- Access applications with a Jet database are not as secure as client-server based database using a more advanced DBMS. Jet databases also are not as fault-tolerant as more advanced database engines. For example, if there is a power-cut in the middle of an update, a Jet database file might become corrupted whereas more advanced DBMSs will "roll back" to the last stable version of the database.
- Access requires Windows as an operating system and also requires all of the software to be installed on each computer, even when used in a networking environment. If the Access application is compiled, licensed copy of Access is not required on user machines but will be required for development, custom query, and reporting.
- It is more difficult to update Access as each machine must be updated. Further, since all of the software runs on each computer, the memory and processing power of the computer must be able to handle the application. This generally requires more expensive computers than are used to run thin client web browser applications
- Incompatibilities exist between various versions of MS Access and this can make it difficult to maintain. Software applications that depend on older versions may have difficulty with newer versions of Access. A homogeneous MS Access version environment is important.

1.1.1.2 OpenOffice.org Base

The new version of OpenOffice (www.openoffice.org) contains a database and a front end forms, reporting, and query designer similar in functionality to Microsoft Access. It installs a SQL database by default (HSQL) which has good performance and is easily scalable to large data sets. The front end of Base can be linked to other databases including MySQL and Postgre for improved performance and large data sets.

One advantage of this is that it is all free open source software so there are no licensing fees. Another is that all of the data is stored in open published formats so there is no chance of getting locked out of access to your data. The OpenOffice suite also contains word processing, spreadsheet, presentation, and graphics software that provides a free alternative to Microsoft Office.

This type of database with front end forms, reporting, and query is good for simple standalone and networked database projects and can also provide a good "ad hoc" query capability to large data sets stored in standard SQL databases.

2.10.3.Other software required

Some ART application packages may require the purchase of additional software packages such as reporting software for data analysis. Make sure that it is included in the calculations.

2.10.4.FOSS

The free open source software (FOSS) alternatives that are available for operating systems and databases have surpassed the performance and reliability of the proprietary alternatives so you should strongly consider using these. In addition to low cost (free), they are in most cases more reliable and offer better performance.

The best model for providing customization and building capacity in country is the open source software (OSS) model. OSS is published under a copyright that allows free copying and distribution, encourages modification and enhancement of the software, and requires that changes are also published. Besides providing free software, OSS allows each country to develop the capacity to maintain and modify the software to meet their needs. This is a "development friendly" software environment. OSS gives countries access the tools they need to use to develop their indigenous information technology assets. OSS also ensures that countries can always have access to their valuable data unencumbered by proprietary file formats and software licensing.

OSS also benefits from "network effects". The OSS model encourages individuals to modify the software and to publish their modifications for others to use. What this means is that even though an individual country may have weak capacity, they can draw on the efforts of the other countries who are using the same RIHIS and they can contribute their changes to others. There is the potential to develop a powerful collaborative workgroup of countries using the RIHIS. International organizations can provide seed money, technical expertise, organize the network. Everyone can contribute to the network. The result is a "virtuous cycle" that can help bootstrap the technical capacity of developing countries and help them become free from dependence on first world assistance. An additional advantage of OSS is that it is "peer-reviewed" in that the software is published for others to review and check for accuracy. This leads to much higher quality software since errors are more likely to be discovered and corrected.

The Open Source model also avoids the problem of abandoned software. Commercial software and even software that is distributed freely (without source code) and controlled by a single entity is tied to the survival of the sponsoring organization. If that organization fails or decides to discontinue the software, the software can be orphaned and left without support, maintenance or upgrades.

2.11.Installation/Configuration

2.11.1.Ease of installation

2.11.2.Set-up and configuration

2.12.Financial Costs

2.12.1.Cost of Software itself (initial and recurring fees)

Cost of software for initial purchase and any recurring fees.

Often proprietary software will have an initial purchase cost plus an annual maintenance fee. Ensure to include both in calculations of costs.

2.12.2.Costs of underlying hardware (including back-up systems, maintenance and repair)

These costs can be estimated from the information in the fields above on the categories of systems required for clients and servers. Also include cost of battery power backup systems, network equipment, wiring, maintenance and repair.

2.12.3.Costs of underlying software (including ongoing license/support fees)

If the choice of not using free open source software has been made, make sure to for investigate continuing license fees. Many proprietary databases have annual fees for maintenance.

2.12.4.Cost of modification of software (external contract costs)

The initial cost estimate for the software should include a cost for customizing the software to meet your specific requirements. Most software will require modification to meet local requirements. The initial purchase agreement should specify these modifications and cost.

2.13.HR Costs (FTE by skills)

When planning a system, the cost should include many factors beyond the purchase of the software and hardware. Often these are significantly more than the initial “software cost”. In preparing a budget, consider each of these costs for your planned installation.

2.13.1.Customization to local requirements

2.13.2.Documentation (local)

2.13.3.Training (data entry, analysis, supervision, tech support)

2.13.4.Start-up implementation

2.13.5.Ongoing operations (data entry, supervision, analysis, reporting)

2.13.6.Technical support

2.13.7.Maintenance

2.14.CDC Requirements

The US CDC has established recommendations for software that promote ease of use, operation, and data communication. These may not be an absolute requirement in your situation but they are important factors to consider when making a decision on software.

2.14.1.Uses generic web-browsers for data entry and analysis,

The program should use current standard HTML for access over Internet protocols.

The universal web browser offers many advantages since it has a standard graphical user interface and can run on low performance (inexpensive) hardware using standard Internet TCP/IP protocols. Preference should be given to systems that use this interface.

2.14.2.Underlying database is an industry-standard SQL solution with ODBC support,

There is an industry standard for database design which is called SQL. There are many tools to support this standard which ensure that you can have full functional access to your valuable data. Open Data Base Connectivity (ODBC) is a standard method of connecting different databases on different operating systems. Preference should be given to software systems that use an SQL database with ODBC connectivity.

2.14.3.Authenticates through an LDAP directory,

As part of the security and access setup of your software, it will require a directory of authorized users and their access permissions. A standard method of setting up this directory is the Lightweight Directory Access Protocol (LDAP). The advantage of using LDAP is that it can facilitate common access privileges across multiple applications providing they all use LDAP. This greatly simplifies the setup and maintenance of directories. Preference should be given to software systems that use a LDAP directory.

2.14.4.Security is handled through SSL connections based on X.500 digital certificates,

The X.500 directory services form the core of the LDAP and issues the authentication certificate. This should be done using the web Secure Socket Level (SSL) feature to ensure that there is no compromise of the authentication directory or certificate information. Preference should be given to systems that use this secure method.

2.14.5.Uses HL7 for data import and export,

Health Level 7 (HL7) is a standard method of communicating medical information which includes standard tags defining information along with the content. It has been widely adopted in the health field to transfer information between computer systems. There is also an evolving standard for ART using HL7 (HL7ART) which is being organized by the World Health Organization. ART software systems must be able to send and receive information with other systems. HL7 is the method of preference and the ART software you select should offer HL7 support.

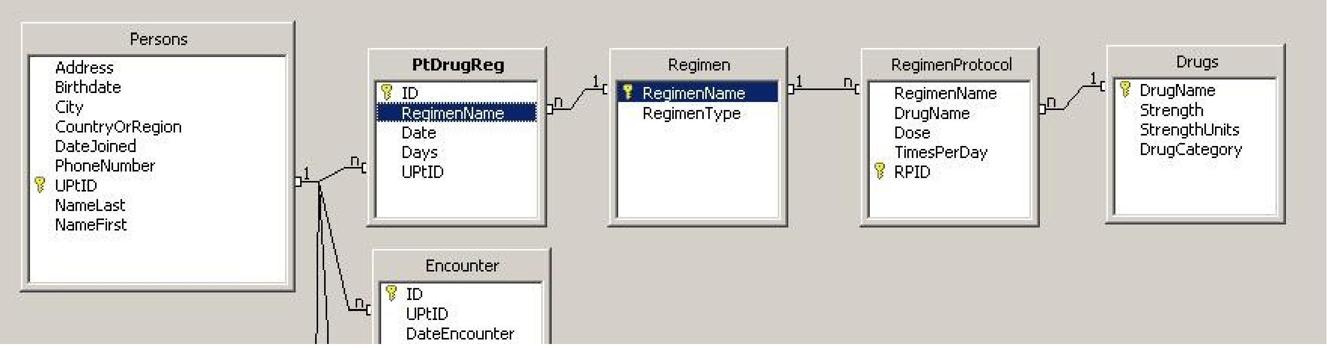
2.14.6.Uses ebXML for Internet-based data brokering.

Electronic Business using eXtensible Markup Language (ebXML) is a family of XML based standards sponsored by OASIS and UN/CEFACT whose mission is to provide an open, XML-based infrastructure that enables the global use of electronic business information in an interoperable, secure and consistent manner by all trading partners. This level of communication goes beyond HL7 to include generalized automatic messages. Typical uses in the ART context may be laboratory test ordering, billing and results reporting. It supports advanced communications among information systems.

1.2 Drug Regimen Management

Successful ART requires management of complex drug regimens usually consisting of multiple drugs in combinations according to accepted protocols. In addition, patients will typically be on additional drugs to control or prevent opportunistic infections. It is crucial to understand not only the patient's current medications but also the patient's drug history of prior regimens and any treatment failures and reasons for changing regimens.

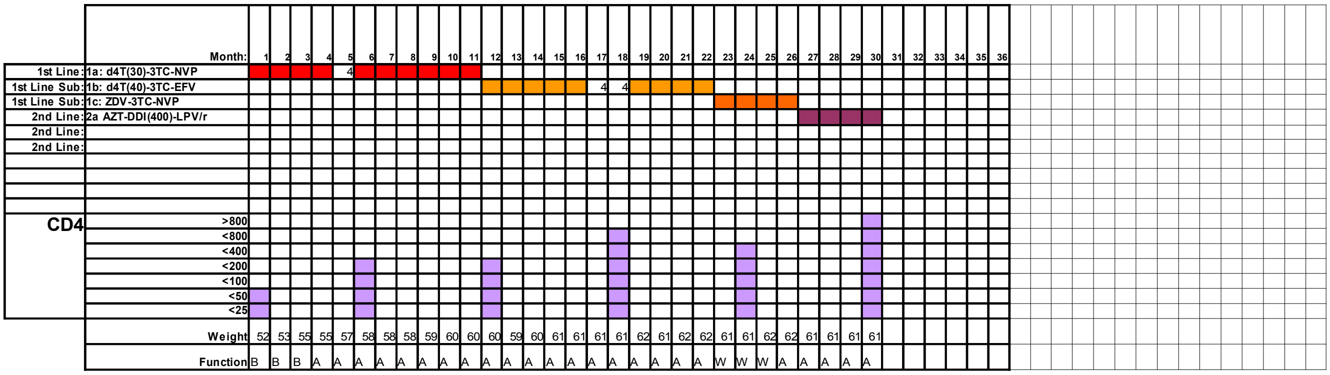
In order to do this, the ARTIS system must have rich features to record and display drugs. These features are described in detail in the "Functional Requirements" section under "Drug Regimen" and "Logistics" which refer to the capabilities of specific systems.



This diagram shows a typical data relationship to track patients, drugs, and regimens. Note that the patient is linked to a specific drug regimen with a date and duration. The regimen contains a number of drugs, each of which has a dose and protocol. This information is sufficient to capture all of the necessary information about a patient's drug history including (individual drug name, dose, frequency, duration, drug category. The regimen approach bundles multiple drugs together which makes it easy to record standard drug regimen protocols for individual patients.

Different ARTIS may have variations in implementation of drug regimen features. It is generally accepted that the individual drugs in each regimen should be specifically listed and stored in a patient linked table with each drug regimen order rather than relying on the system to maintain the links from drug regimen to specific drugs. Drug regimens will change over time and the specific drug components will change and when it does change, it is important to retain the original drugs that a specific patient received as per their record.

The presentation of complex drug history information at the point of care is important to facilitate decision making. Formats have been developed for paper based systems and these can be automated for computer presentation. The "ART card" format shown below makes it easy to see a patient's drug regimen history along with other pertinent information such as laboratory, weight, and functional status.



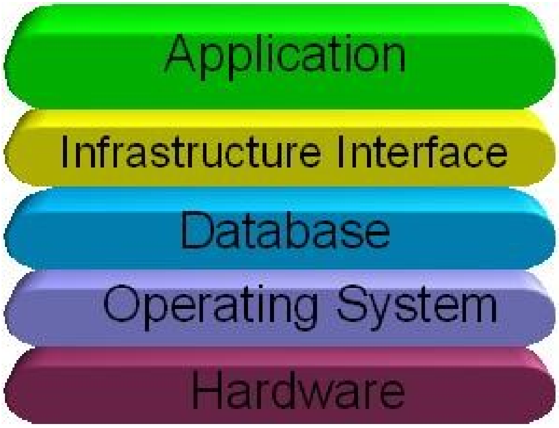
	Month:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1st Line:	1a: d4T(30)-3TC-NVP	🟥	🟥	🟥	🟥	4	🟥	🟥	🟥	🟥	🟥	🟥																									
1st Line Sub:	1b: d4T(40)-3TC-EFV												🟧	🟧	🟧	🟧	🟧	4	4	🟧	🟧	🟧	🟧														
1st Line Sub:	1c: ZDV-3TC-NVP																							🟧	🟧	🟧	🟧										
2nd Line:	2a AZT-DDI(400)-LPV/r																											🟪	🟪	🟪	🟪						
2nd Line:																																					
2nd Line:																																					
CD4	>800																														🟪						
	<800																		🟪												🟪						
	<400																		🟪						🟪						🟪						
	<200						🟪						🟪						🟪						🟪						🟪						
	<100						🟪						🟪						🟪						🟪						🟪						
	<50	🟪					🟪						🟪						🟪						🟪						🟪						
	<25	🟪					🟪						🟪						🟪						🟪						🟪						
	Weight	52	53	55	55	57	58	58	58	59	60	60	60	59	60	61	61	61	61	62	61	62	62	61	61	62	62	61	61	61	61						
	Function	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	W	W	W	A	A	A	A	A						

Another important issue is drug logistics. The task of making sure that clinics are stocked with the necessary drugs for complex drug regimens is difficult. The ARTIS can provide valuable information to inform the drug logistics system. A good ARTIS can provide reports on actual dispensing of drugs. In addition, it is possible to forecast future drug demand. The ARTIS can use the information it contains on the number of patients, their current drug regimen, the typical durations of the various regimens, patient status to model future demand for each drug regimen. For instance, a good model will be able to detect a rise in drug resistance or adverse reactions and change its forecast accordingly.

Monitoring and evaluation of patient compliance and staff performance are also important information that can be derived from the ARTIS. Patient compliance is dependent on a number of factors, some of which are under the control of the clinic staff and health system. As an example, stock outs are an important factor in compliance and the ARTIS can report on these. A good ARTIS will track compliance and reasons for compliance to allow managers to improve the performance of the health delivery system.

3. Information System Requirements

A computer information system consists of the actual application software that provides a user interface and performs the desired specific functionality. The prior section on "Functional Requirements" covered application software parameters. Underlying the application, however, are several layers of software, and ultimately, the computer hardware. The sum total of all of these layers is the "computer system". This section will address computer system selection issues below the level of the application.



This diagram is a generalized diagram of the various tiers that make up a computer system. The Functional Requirements section discussed the Application and Database tiers and also made some comments on operating system and hardware requirements. Sometimes these functions are separated on different hardware, as discussed below.

3.1. System Architecture

Architecture refers to the organization of various software and hardware bits that make up a system. For purposes of ARTIS, we need to be aware of network architecture and multi-tier architecture.

3.1.1. Network Architecture

A network permits individual computers to communicate and share information. This is of fundamental importance to enable the effective sharing and use of information. Stand-alone computers are of limited usefulness. However, when computers and information is shared, the information becomes much more valuable.

Computer network architecture is designated as "local area" or "wide area". The distinction between local and wide area is that a LAN is usually confined to a single building or group of buildings and wide area networks cover large distances (the Internet is a WAN). Current LANs are most likely to be based on switched Ethernet or WiFi technology running at from 10 to 1000Mb/s (megabits per second). The defining characteristics of LANs in contrast to WANs are: (a) much higher data rates, (b) smaller geographic range and (c) they do not involve leased telecommunication lines.

3.1.2.Multi-Tier Architecture

Multi-tier architecture is a method of dividing up the work of a computer application and distributing functions to various computer systems each of which can be optimized to perform that function. The multi-tier architecture functions often run on machines that are connected over local or wide area networks. There are many variations in the arrangement of hardware and software configuration. The most common form of a multi-tier architecture is referred to as client-server and this architecture is particularly well suited to ARTIS since it is optimized for database access.

Client-server architecture is based on running a database server and clients that are either located locally using a LAN or over a WAN. The Internet is an implementation of a WAN.

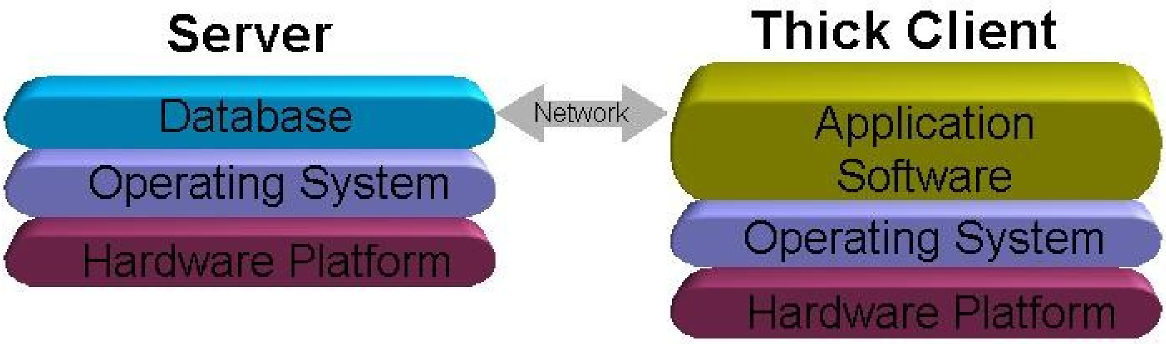
The client-server architecture is very popular because it is scalable. It can run on a single computer (where the client and server are both installed on the same computer), a LAN with a separate server and multiple clients (using LAN or Internet protocols and a web browser or thick client), or a WAN (using Internet protocols). It is excellent architecture for deployment at a national or regional level.

The “client” side of client-server computers can be either “thick” or “thin.” Thick clients install application software on the client and this is capable of extensive data processing and interface manipulation. Thick clients usually use the server only for the purpose of storing and retrieving data. Thin clients, on the other hand, use only a small software program on the client (usually a standard web browser) which has limited data processing ability. Most of the data processing is done on the server. As discussed below, there are variations of pure thin or thick clients. Some thin clients are getting thicker (and more capable) with the addition of Java, JavaScript, and a combination of techniques known as "AJAX".

Client-server configurations offer in principle a higher level of security since the server (and the data) can be located in a physically secure area that does not require routine access by personnel. This secure site can be protected from environmental damage and unauthorized physical tampering. Client-server configurations are somewhat more difficult to design and install and require local or wide area communication links in order to function. These links can be problematic in areas where electrical power and communications infrastructure are not reliable.

The following is an overview of how a client-server network can be connected:

3.1.3.Thick Client - Server



An example of a Thick Client – Server configuration is:

Server: Microsoft Windows operating system, Microsoft Access or SQL Server database

Client: Microsoft Windows operating system, Microsoft Access or Visual Basic application

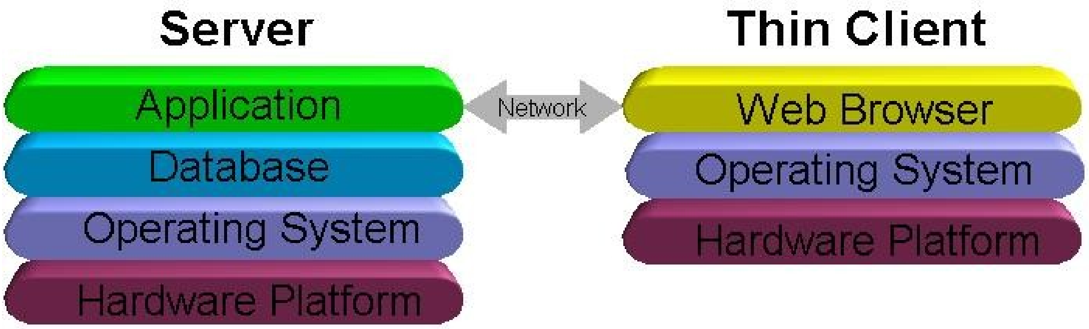
The Microsoft .NET platform is also an example of a Thick Client – Server but it also includes a “thick server” component with an additional processing layer installed on the server.

To be able to handle increased number of users with minimum disruption and reprogramming, the server only stores data so there is no high demand on it and it’s possible to add a large number of clients without stressing it.

Thick clients have a rather large application software package installed. This requires more computing resources on the client. In addition, when the software is changed, the new software must be installed on every client. This can be automated, either using scripting or using Net Controller software, as long as the client machine has a reliable network connection.

3.1.4.Thin Client – Server

An example of a Thin Client – Server configuration is:



Server: Linux operating system, MySQL or Postgre database, PHP or Java application

Client: Any current operating system running any current web browser.

Hardware requirement for client computer is low, since the client machine does not have to store any data or carry out any data processing, older machines are sufficient for the task.

Since there is only a simple web browser installed on the client, the client portion is very easy to set up and maintain. When the application software is updated, it only requires changing on the server. The clients do not need to be updated. (These machines though low on maintenance are not completely free since they may have to have local operating system, browser, or anti-virus software updated.)

Client machines need a reliable connection to the server, and, where the transaction volume is high, sufficient bandwidth is necessary. In many settings neither reliable phone line nor network links are available

3.2.Platform

A computer hardware and software "platform" is the underlying foundation which runs the application software. Some software platforms are closely tied to hardware (such as Microsoft Windows is tied to the Intel hardware). Other software platforms such as Linux can be transported (by recompiling) to different hardware. Still other software platforms (such as Internet web and Java) are designed to run across multiple hardware and software operating system platforms.

Platform decisions become important for two reasons. First, there usually exists a substantial investment in "legacy systems" hardware and software. Second, in most cases, the new ARTIS will need to be compatible with this legacy hardware and software.

3.2.1.Operating Systems

The operating system is the first software level just above the hardware. It provides services that allow software access to hardware resources such as processing instructions, memory, data input and output (display, keyboard, network, printers, etc.)

There are many different operating systems. However, for purposes of the desktop software that has been evaluated, there are only two that need to be considered - Windows and Linux.

1.2.1.1 Windows

The Microsoft Windows operating system is the most prevalent OS found on desktop computers. Most of the ART software will run on Windows. There have been various versions of Windows over the years and most of the ART software requires the more recent versions such as Windows XP or Windows 2000.

The Windows operating system is somewhat controversial because in spite of its popularity, it does not have a good reputation for reliability and it is frequently subject to attacks by computer viruses and other malicious software. Windows was originally designed as a single user isolated computer operating system and has had difficulty adapting to a connected world. However, the popularity of Windows means that it is easy to find people who can maintain the software. In addition, there is a very wide range of softwares available for Windows.

1.2.1.2 Linux

Linux is a free open source software system. Most of the Internet runs on Linux computers and it is gaining popularity as desktop operating system.

Linux was designed from the start as a multi-user connected computer and because of this has been free of viruses and malicious software. It has also been proven to be more reliable.

However, Linux is less well known and it can be difficult to find people to set up and maintain Linux computer systems. There are fewer software packages written for Linux although all of the common software functions are covered. For instance, there is an excellent word processing, spreadsheet, presentation, and database suite called OpenOffice.org that is compatible with Windows.

3.2.2.Infrastructure Interface

The infrastructure interface is a level of software between the operating system and the user application that can provide many useful user interface and communications facilities. This level of software can also provide developers with tools to make it easier to program software for applications. There are several specific infrastructure interface level platforms that are of relevance to ART software. These are the Internet-Web, Microsoft .NET, Java, and AJAX.

3.2.3.The Internet - Web Platform

The combination of a web browser and an internet connection to a server provides a user interface and access to services such as databases. There are many web development tools and languages to help in creating user friendly applications. The most popular configuration is a combination which goes by the acronym of LAMP. This stands for Linux, Apache, MySQL, and PHP. Linux is the operating system, Apache for the web server, MySQL database, and PHP programming language. This combination is used to run many popular high volume web applications. It is easy to set up and maintain.

The Internet-web platform consists of two parts. The server as described runs LAMP and hosts the application and all data. User (client) access is through a standard web browser which can be run on any platform. The web browser has very low resource requirement and runs on just about any existing desktop computer equipment including many older computer systems. All of the processing is done on the server and very little required of the client.

The Internet-web platform is a good solution for networked applications.

3.2.4.AJAX

A specific extension of the Internet-web platform is called AJAX. Asynchronous JavaScript and XML, or Ajax, is a web development technique for creating interactive web applications using a combination of: XHTML (or HTML) and CSS for presenting information; the Document Object Model manipulated through JavaScript to dynamically display and interact with the information presented; and the XMLHttpRequest object to exchange data asynchronously with the web server.

Like LAMP, Ajax is not a technology in itself, but a term that refers to the use of a group of technologies together.

Ajax applications look almost as if they reside on the user's machine, rather than across the Internet on a server. The reason: pages get updated, not entirely refreshed. Traditional web applications submit pages, to a web server. The web server responds by sending a new web page back. Because the server must submit a new page each time, applications run more slowly and awkwardly than their native counterparts. Ajax applications, on the other hand, can send requests to the web server to retrieve only the data that is needed. On the client, JavaScript processes the web server response. The result is a more responsive interface, since the amount of data interchanged between the web browser and web server is vastly reduced.

3.2.5.Microsoft .NET

Microsoft has been developing and promoting its ".NET" platform for several years and some applications have been moved to this platform or are in the process of being developed in this platform. The .NET platform is a combination of development tools and operating environment that permits the development of highly interactive software that can run on LAN and WAN configurations.

The .NET platform is a type of "Thick Client – Server" which requires Microsoft Windows for both the server and client machines. It is designed to scale to large installations and can use several different high capacity databases.

The disadvantages of .NET include its requirement of a large investment in hardware and software on both the client and the server. In addition, it is complex to install and maintain, and is still evolving which means frequent upgrades and instability.

3.2.6.Java

The Java platform is the name for a computing environment which can run applications developed using Java programming language and a set of development tools. In this case, the platform is not a specific hardware or operating system, but rather an execution engine called a virtual machine, and a set of standard libraries which provide common functionality.

Java is a computer processing language that runs on browsers and can be used to provide more data processing capability on a thin client. Java is software that is loaded from a server and runs on the client to perform data and user interface manipulation on the client. One important advantage of Java is that it runs on all current computer platform operating systems. A single Java program can run on many different computers. All that is necessary for a client computer to run Java is for that computer to have a Java runtime environment (JRE) a set of programs for all the client computer to run Java applications. Java is known as a "write once, run anywhere" computer language and is very popular application.

The Java language has a high level of security built into its structure to prevent the language from corrupting the client computer. Java security has been designed for a networked, multi-user environment and offers a high level of protection.

3.3.Open Standards Compliance

Computer hardware and software is capable of a great range of variety. However, it must all work together. The organizational structure that keeps hardware and software from different sources working together is standards. There are hardware standards for plugs and signals over various wires. There are software standards for communication and interaction. High level software standards allow designers to take advantage of unique hardware and software capabilities and permits sharing of information. Standards in effect permit information to transcend physical hardware and virtual software worlds.

Data without communication is useless. Communication makes data valuable by allowing it to be shared and analyzed. In order to do this, there must be open standards for storing, retrieving and communicating data. An open standard is one that is published so that anyone can implement it and gain access to the information. In the past, closed proprietary standards were developed by many companies as a way to "lock in" their customers to their software. The closed proprietary standards prevented customers from accessing their information without purchasing access rights from a commercial organization. This increased costs and reduced access to information. Currently, most organizations realize the value of open standards and demand that any software they purchase support open standards to ensure access to valuable data.

We have discussed the high level HL7ART data standard in the "Functional Requirements" section above. This standard is what will permit communication and use of patient information on different hardware and software.

There are other industry standards for network hardware and software such as Ethernet and WiFi. There are standards for data communication such as the foundation of the Internet which is called TCP/IP. The Internet also supports standards such as HTTP, HTML, and XML.

The important factor to keep in mind is that hardware and software must adhere to standards to be effective. As far as possible avoid proprietary standards that lock you into a particular vendor or prevent you from accessing your information without proprietary software from that vendor. Information is valuable because it can be shared, aggregated, and analyzed. It is worthless if it is locked up.

4. Operational Requirements

Selection of an ARTIS is only the start. Successful operation of the system is crucial to continued collection of data and access to information. Software systems should be designed with tools to facilitate operation of the system and there are many hardware and software decisions that can be made to improve operation.

These factors should be addressed during ARTIS selection and implementation. Usually, operation is a joint responsibility of project staff and vendor staff. The division of responsibility and support mechanisms should be carefully defined to ensure that implementation and operation proceed smoothly.

Since the majority of the cost of an ARTIS takes place over the operating life of the system, it is crucial that operating procedures are clearly defined.

4.1.Operations Monitoring

Comprehensive monitoring of the entire production ARTIS environment (hardware, network, application, OS, security, etc) is critical. An effective monitoring solution can often predict and fix problems before they adversely affect the end users of the application. An effective monitoring system should be implemented and operated by support personnel. The monitoring system is separate from the ARTIS system itself and consists of a collection of tools and procedures. The following items should be considered when implementing a monitoring system:

- Event reporting
- Application events reporting
- System Monitoring:
- CPU utilization
- Disk Space utilization
- Memory utilization
- Operating system monitoring
- SNMP capability to monitor external devices (routers, etc)
- Web server monitoring
- Network activity (traffic in/out).

4.2.Browser Support

The ARTIS should support standard Internet browsers that are available on common platforms. Currently, the browser that is most in compliance with Internet standards is the Mozilla foundation browser engine. This is implemented and distributed free in various configurations for all current computer operating systems (including Windows, Macintosh, and Linux) at www.mozilla.org. This browser is distributed under the names Mozilla, FireFox, and Netscape.

Microsoft also distributes their Internet Explorer browser on the Windows platform only. This comes installed on most Windows systems. Unfortunately, it is not as compatible with the open standards for web browsers as the Mozilla browser and this has forced many web server designers to program special code for handling Microsoft's unusual implementation of the web. Also, the Microsoft web browser has unfortunately had a high incidence of malicious software attacks. One large security vulnerability is the "ActiveX" function which opens up the client's entire operating system to outside attacks. If Microsoft Internet Explorer is used, ActiveX should be disabled.

In order to ensure compatibility with the current and future web, the ARTIS system should be compatible with current web standards and the Mozilla browser.

4.3.Security

Network security is an important function to ensure privacy and integrity of system data. The entire field of security is complex and is beyond the scope of this document. However, this section will outline the significant factors that need to be considered when configuring an ARTIS.

4.4.Server Configuration

In most systems, the server stores all information and must be protected carefully from unauthorized access and tampering. Server user login accounts should be limited to few trusted personnel involved in the setup and/or maintenance of the ARTIS application.

4.5.Virus Protection

An anti-virus application should be installed, running and kept up to date with virus signatures on all servers, and should be configured to perform weekly full scans of the system. The anti-virus application should also be configured to automatically trigger alerts to support personnel in the event of virus detection.

The virus signature files should be kept up to date automatically by querying the antivirus vendor’s site daily for signature updates. When an update is available, should be downloaded and installed automatically.

This automated process ensures that updates are as timely as possible and it requires no manual involvement from the System Administration staff.

4.6.Network Configuration

The ARTIS production server should be hosted in a protected zone (DMZ) behind an Internet firewall that enforces and monitors all Internet traffic to the application.

The ARTIS servers should be configured to open only the TCP/IP ports required by the application.

4.7.Encrypted User Communication

The production system will require the installation of Secure Socket Layer (SSL) certificate using strong 128-bit Key encryption on the application server. SSL provides encrypted transmission of any data being exchanged between the server running the ARTIS and the client system (Internet browser). Without SSL, sensitive data such as passwords and patient information would be transmitted as clear text and would thus not be protected from hack attacks.

4.8.User Authentication

The system needs to authenticate, enforce and maintain unique system wide user id and password credentials, for every user that has been granted access to the system, the application should only allow authenticated user access. The issuance of user ids and passwords to users for online access should be limited to a select group of authorized users.

The system automatically logs off users after a specified time of inactivity.

All user passwords in the database should be encrypted so they can be deciphered only by backend server modules. All passwords should be transmitted and stored in encrypted fashion.

There are various methods of organizing access control. The "Lightweight Directory Access Protocol (LDAP)" is an open standard for controlling and monitoring user access and can be extended to multiple applications.

4.9.User Authorization

User authorization is commonly organized around "roles". Roles are defined for different types of users of the system. For instance, a person who has the job of registering patients needs access to a certain set of information and functions but not others, different from the access required by the person running reports.

Every user that has been granted access to the ARTIS must be assigned a specific Role in the system. The user role defines a set of privileges for a particular user. These privileges in turn indicate to the application whether to grant or deny particular user access to specific functionality in the application. The ARTIS must support the assignment of different roles to different users.

4.10.Audit trail

An important function is the audit trail which tracks and logs all critical interactions with the user including capturing the identity of the user, the user's action, and the timestamp of the action. Sample user interactions include:

- User successful and failed login attempts
- Updates to user profile such as name, credentials, address and contact information
- Creation and updates to patient profiles

- Viewing patient medical history and/or test results
- Creation of orders by user
- Order updates by the user
- Creation of test reports by user
- Creation, review and updates of billing information and statements.

4.11.Deployment

Critical factors for a successful deployment include the following:

- Clearly defined project teams and roles assumed by appropriate individuals within each organization.
- Availability of necessary staff participation to assist in the deployment and training needs.
- Availability of necessary staff participation in this effort and providing additional staff members as per requirment.
- Timely and effective communication.
- Effective management of the deployment and training plan.

When ready for system installation, a formal deployment plan should be developed that addresses the critical issues related to a successful deployment. A recommended approach for a deployment would include the following items:

- Define the deployment teams
- Establish a release to production criteria that will allow the final software to migrate to the production environment. Criteria should include the following:
- There are no Open bugs.
- Test cases performed during the Integration and System Test phases have passed.
- There are no discrepancies between the version released to the staging and production environments and the version used during the final regression testing.
- The User Acceptance Test (UAT) was successfully completed.
- UAT issues were addressed.

- Setup the production hardware, software and network environments

Once the final software is deployed in the production environment, the following high level tests should be run to validate the system setup and ensure all the system's functioning. It will not be necessary to run the entire test suite in the production environment from a functionality viewpoint, but only a subset.

- Network – Test that all servers are networked correctly and are able to communicate via the preset protocols and ports.
- System Accessibility – Test the accessibility of the ARTIS application via the preset URL for both internal and remote users.
- Application Communications – Ensures all applications running in the various environments can successfully communicate via the preset protocols and ports.
- Systems Integration – Ensures the ARTIS and related applications function well in the same environment.
- Security – Test SSL encryption between a client (user) browser and the server hosting the ARTIS web server.
- Application Security – Test various layers of security within the ARTIS application including:
- Passwords are fully encrypted within the database
- User authentication
- Operating System Server Security – Analyze system and security logs to ensure that the integrity of the system and network are maintained:
- Operating system event logs
- Database audit logs
- Virus protection logs (if available)
- Application monitoring logs
- Application server logs
- Internal Users – Ensures a typical internal user can successfully access the ARTIS application, save data, and print from the application
- Remote Users – Ensures a typical remote user can successfully access the ARTIS application, save data, and view the printable screens/reports
- Printing – Includes sending the printable test reports and billing statements to a printer already installed on the user's computer (local or network).
- File Upload/Download (if required) -Test upload/download of files from a source to the ARTIS application.
- Application Tuning – The applications may require some tuning to maximize the performance based on the traffic and load experienced during the testing phase in the production environment.
- Restoration – Once testing is completed, the database and all initial settings will be restored.

Plan data migration

Handle legacy orders

Restore production data in the production environment, (create users, roles, etc.)

Go live with a pilot deployment that includes a limited number of users to minimize the support effort required in the initial phases of deployment,

Generate release notes and documentation changes,

Go live

Production support must be in place to ensure the system is operational and monitored during business hours, and can be attended to when required. In addition to the high level deployment plan, a detailed deployment checklist should also be developed. This checklist should include all necessary tasks related to the installation, configuration, testing, etc of the ARTIS system.

4.12.Support

On-going support for the ARTIS deployment falls under both the vendor (or other entity that created the system also referred to as ‘vendor' here) who will have to continue to provide support, maintenance and upgrades for the ARTIS and under the project's IS group who will be responsible for the day to day running of the system.

In selecting a vendor specially consider the vendor's ability to deploy support personnel at a reasonable cost and in relative proximity to the country. This means that the country should not be reliant on support personnel that are based only in North America or Europe. The vendor should be expected to provide local resources.

A regional support center can provide second line support to groups of countries in close proximity.

On-going first line support will be provided by IT staff that is directly controlled by the Public Health or other related departments within the countries. Deploying and providing ongoing operational support for the ARTIS by these groups will require skills that span across the following disciplines:

- Operations Personnel
- System Administration
- Network Administration
- Database Administration
- Quality Assurance personnel.
- Hosting

The ARTIS server infrastructure should be hosted in a controlled environment with adequate power, physical security, cooling, and UPS battery backup in the event of temporary power interruption.

4.13.Incident Processing and Tracking

The selected vendor shall define a technical support process to enable the lab to submit support requests in the event of problems or issues with the ARTIS. Incidents should be able to be submitted either by phone, email, or using a web form. The Support system should have the ability to track support incidents that are submitted via either of these methods. Once the customer submits a technical inquiry to the support organization, an "Incident" is generated. An Incident is the processing of a technical inquiry or the attempt to solve a technical problem, regardless of the number of required phone calls or e-mails. Opened Incidents remain open until a solution is achieved or the incidents closed upon mutual agreement with the Customer.

The lab shall determine the urgency of the support inquiry in coordination with the support organization. The Support engineer may use his or her reasonable discretion to change the processing order of inquiries in case of identical urgency and priority, or for reasons of efficiency, provided the postponed Customer does not suffer any significant disadvantages.

An example of such a schedule is listed below and should be defined in a service level agreement.

Urgency Levels Specification Response Time

Level 1 – Critical Database server or failure of ARTIS Software components or major system problem causing entire application to be unavailable. < 1 business hour

Level 2 – High Error/issue causing an interruption of business, or possibly causing server failure. Portion of application may be unavailable. Temporary workaround may be possible. < 4 business hours

Level 3 – Low - A problem affecting the production in the long term, but not causing immediate failure. Workaround is possible. < 8 business hours

4.14.Application Software Maintenance

Procedures for change control need to be put in place to address the following types of application software upgrades:

Critical bugs related to the ARTIS application should be resolved in a timely fashion. Such bugs may result in erroneous data and don't usually have a workaround. The implementation and deployment timelines will be mutually agreed upon.

Implementation of any minor enhancements or updates related to the application. Such enhancements do not include major database or other infrastructure and design changes. The implementation and deployment timelines will be mutually agreed upon.

Procedures for maintenance should be clearly specified. All software requires maintenance and requirements will change during the life of the application. Local software maintenance has the advantage of good communication and can provide better service. However, technical skills may not be available locally and external assistance may be necessary. Open source software has the advantage in that local capacity for maintenance and can be developed over time. This also contributes to the technology infrastructure development of the country and is an important development objective.

4.15.Training

Adequate training must be performed before authorized users are allowed to use the system in a production environment. The following approach is recommended to achieve the training goals:

- Create training material based on the delivered user documentation,
- Train the trainers. How to use the software, including system setup and configuration, system maintenance and other routine operations. The training will be directed towards system users as well as the system administration and technical support staff personnel.
- Training sessions should be performed on the staging environment,
- Use simulated/test data (patient demographics, results, etc.) that will not be maintained after the training is completed,
- Track and address issues raised during the training phase.

4.16.Backups and Disaster Recovery Plan

The ARTIS should have the capability to perform routine backups of the data in the system. The ARTIS must have the ability to do both manual and scheduled data backups. The data backups must be done to a storage media that can be stored off-site and should also provide for the ability to do backups to remote connected devices as well, for example a Storage Area Network (SAN) architecture in the future. The system must provide for data restore and recovery capabilities as well.

4.17.Data-Entry Procedures While System Is Down

Most ARTIS will rely on a combination of paper and electronic data access and entry procedures. Provisions should be made to ensure continued operation of the system in the event the electronic system is not functioning. Most often, the paper procedures will be adequate.

5. Documentation

Documentation will include sufficient and detailed documentation for each user "Role". The specific documentation guides shall be provided to cover the needs of the following users.

Installation and Administration

The installation guide shall be detailed enough to provide system's administrators with all of the information necessary to install and bring the application live in a production environment. The documentation shall list all utilities and tools necessary for the proper administration of the ARTIS. These tools shall cover management and administration of the ARTIS database, the user interface, and any auxiliary programs integrated into the delivered ARTIS.

User guide

Laboratory User documentation shall be sufficient and detailed enough to allow different levels of lab users to perform their functions with out having to rely overly on external support.

Online help

The system shall provide context sensitive online help at the data field level.

Training

The system vendor shall submit a training plan which shall include the cost (if any), course titles, a summary of content, and state the position/qualifications of the target audience. Training shall be provided on two levels; ARTIS system maintenance and system operation designed for IT staff and user training designed for personnel that will be using the ARTIS on a day-to-day basis.

Operations – trouble shooting, escalation procedures, knowledge database.

Systems Administration documentation shall include information on trouble shooting and support escalation procedures. The system must also provide a centrally managed, web based knowledge database that users can access to get the latest information on trouble shooting, support, bugs, updates etc.

1. Global Standards

In order to provide assistance with the complex task of ART patient monitoring, international organizations have developed standards and guidelines. These provide standard data definitions, data sets, and data interchange standards. Together, these ensure accurate, complete information that can be transferred at the patient level and analyzed at higher levels for program monitoring and evaluation.

5.1. WHO ART Patient Monitoring Guidelines

The World Health Organization has developed patient monitoring guidelines for ART. These guidelines are a solid foundation and form the basis of many ART programs. This extract of the Guidelines is provided as an important reference. The complete text is available from the WHO.

Patient monitoring guidelines for HIV care and antiretroviral therapy (ART), WHO, Geneva

Based on the World Health Organization (WHO) HIV patient ART monitoring meeting held at WHO/HQ Geneva, Switzerland from 29-31 March, 2004 http://www.who.int/3by5/publications/documents/artmonitoring/en/

5.1.1.Objectives of the Patient monitoring guidelines for HIV care and ART

These guidelines have been provided by the World Health Organization (WHO) and other international partners in order to aid in the development of an effective national HIV care and antiretroviral therapy (ART) patient monitoring system. Specific objectives include to:

- provide and facilitate national stakeholder consensus on a standardized minimum set of data elements to be included in patient monitoring tools;
- help to establish a functioning patient monitoring system so as to enable the rapid scale-up of effective chronic HIV care, ART and prevention;
- provide considerations for HIV care and ART information systems design;
- introduce the practice of a simple cohort analysis for HIV patients on ART;
- map the standardized minimum set of data elements to the core ART program indicators and other internationally agreed upon indicators; and
- contribute to successful program monitoring and global reporting and planning through the measurement of indicators at the district, national and international levels.

5.1.2.Intended audience

These guidelines are intended for those involved at various levels of the development or revision of patient monitoring tools such as HIV care and ART patient and facility records, registers and reports, or electronic systems, including:

- national AIDS program managers
- ministries of health
- monitoring and evaluation officers
- other providers of HIV care and ART who may be interested in the technical framework underlying the HIV care / ART patient monitoring system.

While the system described in these guidelines will be used by the clinical team providing chronic HIV care and ART, this document is aimed primarily at those involved in HIV/AIDS programs at the district and national levels. There are training materials that are specifically targeted at people working at the facility level.

The “Three Ones” agreement – one national HIV/AIDS action framework; one national HIV/AIDS coordinating authority; and one agreed country-level monitoring and evaluation system – should facilitate the cooperation of stakeholders in using standardized data elements and compatible patient monitoring systems in each country. At the global level, the harmonization of key elements of patient monitoring is one component of a global, coordinated HIV/AIDS monitoring and evaluation strategy.

5.1.3.Patient monitoring within efforts to scale-up HIV care, ART and prevention

The global emergency of HIV/AIDS has led to unprecedented attention and commitment from the international community to improve access to HIV care, ART and prevention. Many developing countries are currently designing and scaling-up large HIV care and ART programs in order to save and improve the lives of those infected and affected by the disease and to reduce HIV transmission. In this context, the ability of countries to provide and sustain effective long term HIV care with ART and prevention is critical. This requires an effective patient monitoring system integrated with care and treatment (and prevention) at the health facility. Success depends on monitoring the program by measuring key indicators and immediately feeding these back so as to improve program activities

This large commitment by governments and international, bilateral and non-governmental agencies to providing access to ART requires the formation of clinical teams at multiple HIV care / ART sites and a system to support this care, both managerially and with training, supervision, clinical mentoring and other quality assurance inputs after training. A patient monitoring system forms the backbone of clinical care, treatment and prevention.

In many health facilities, most HIV care is currently episodic acute care with the exception of TB treatment. Establishing good chronic HIV care including ART requires forming and preparing a clinical team to provide continuity of HIV care. A key element of continuity of care is keeping a record which summarizes this care and allows each health worker or counselor to understand what has happened before the patient's HIV clinical stage, weight, and functional status; what prophylaxis, other medications, education, and psychosocial support have been provided on earlier visits; the patient's family, pregnancy, contraception and TB status (checked at each visit); and a summary of the patient's ART over time. The core of these guidelines is an agreed list of essential minimum standard HIV care and ART patient monitoring data elements and their definitions (Chapter 2 and Annexes A and B). These can be collected in a variety of ways with different formats of patient cards or records.

In addition to tracking important data for individual patient management, clinical teams need to summarize patient data from the group of patients they are responsible for in order to manage their patients better, to plan, order drugs and file this data. The growing number of patients in chronic HIV care and progressively on ART is a management challenge for clinical teams. A patient monitoring system based on chronic care registers helps clinical teams organize the care of groups of patients. Early in ART program implementation, nurses in some facilities without formal chronic care registers or before registers were printed, created their own by drawing columns to collect the necessary data elements on blank sheets of paper or exercise books. This demonstrates the inherent need for clinical teams providing chronic care to collect data on groups of patients in a timely manner despite the existing burden of record-keeping at many facilities. A small proportion of this aggregated data goes "up" and is also used for program monitoring.

In a public health approach to making ART widely available in low-resource settings, patients are started on one of several first-line regimens, based on clinical staging and sometimes a CD4 count. The second-line regimens are limited and more expensive. The success of individual patient management (including survival) and of the ART program depends on the ability to keep patients on a first-line regimen as long as possible.

There must therefore be a serious commitment by the patient, treatment supporter, clinical team and the community to almost perfect adherence and to remain on a first-line regimen as long as possible. It is very important for clinical teams and the managers at district and national level to monitor the proportion of patients who either remain on the original first-line regimen or have substituted to an alternative first-line regimen and the proportion who survive and remain on ART. This is recorded on the cohort analysis report form.

5.1.4.Simplified ART cohort analysis

Simplified cohort analysis is a key component of ART patient monitoring. It should not be confused with cohort studies which are a demanding research activity. In patient monitoring of ART, a cohort is an ART start-up group which in these guidelines (and the WHO illustrative system presented in Chapter 4) consists of all patients starting ART in the same month. Cohort analysis compares baseline characteristics of patients who started on ART, with their status at 6 and 12 months, then yearly. It allows comparison of the proportion of patients surviving on ART, remaining on the original first-line regimen (or substituting to an alternative first-line regimen), and returning to the functional status of working (or playing, for children). Where CD4 counts can be determined regularly, cohort analysis can show the improvement in the median CD4 count over time. The median CD4 count for a group of patient is a good measure of immunosuppression and predictor of mortality and serious opportunistic infections (OIs).

TB programs have demonstrated the importance and feasibility of simplified cohort analyses based on data transferred from TB treatment cards to a register. Cohort analysis is a key organizational principle of TB monitoring. It is carried out routinely and successfully in all national TB programs and is considered necessary to track trends in program progress and determine treatment outcomes for patients. This is often based on a paper register maintained by the district TB coordinator; some countries are now entering the register data electronically in order to generate reports.

The simplified ART cohort analysis form can be filled out by most clinical teams and can provide important immediate feedback on success in keeping patients on first-line regimens. The district ART team, during on-site visits, needs to fully verify the data by going back to the register data for each monthly cohort.

5.1.5.Cross-sectional data on numbers of patients in HIV care and on ART

Efficient management of large numbers of patients and steady work towards national targets for the numbers of patients in HIV care and on ART requires the ability to accurately keep track of these numbers and to avoid double-counting. The patient monitoring system allows clinical teams to tabulate and report on a monthly or quarterly basis on the numbers newly enrolled in HIV care, the numbers waiting for ART, and three ART numbers:

- new on ART (in the last month or quarter)
- cumulative ever started on ART at the facility
- currently on ART at the facility.

Success in reaching ART targets will be based both on cumulative ever started on ART and those currently on ART (subtracting those who have died, stopped ART or been lost to follow-up). These numbers are disaggregated by sex and age because of the importance of monitoring gender equity in access and assuring adequate attention to providing ART to children.

An effective patient monitoring system should be standardized and allow continuity, referral and communication between all levels of care – from records kept by the patient, family or community treatment supporter; to the first-level facility, to the district hospital; to further referral to specialist physicians or for laboratory examinations. The system should be appropriate for adults, children and pregnant women.

5.1.6.Both patient and programme monitoring

Patient monitoring serves two main functions: first, it enables effective clinical management of patients; and second, patient monitoring data are used for program monitoring and management, contributing to standardized indicators at the district, national and international levels for in-country and global reporting and planning.

At the national or international level, countries are developing ways to report on the set of internationally standardized indicators for monitoring national AIDS programs milestones1 and the targets they have set working with large scale-up initiatives such as the “3 by 5” campaign2 or the "2-7-10" targets defined by the United States President's Emergency Plan.3

Indicators are used at various levels and for different purposes as shown in Fig. 1. For example, as described above, the clinical team may use individual patient data for individual clinical management of a patient, while data on groups of patients may be collected and aggregated at the facility level as performance measures (for quality improvement) for the clinical team. Among the key information that may be used to calculate such indicators is: what regimens patients are on; whether or not they are dead or lost to follow-up while on ART (survival); weight, functional status and clinical stage (quality of life and productivity); and adherence to ART, among others (see Chapter 2). Community-level monitoring systems, while not covered in these guidelines, play an important part in patient monitoring, and are currently being developed.

Patient monitoring is the routine collection, compilation and analysis of data on patients over time and across service delivery points, using information either directly from paper forms or entered into a computer.

These data are best collected and stored at the health facility, and include basic patient demographic characteristics and contact information; information related to patient HIV care and ART history; and patient encounter information collected at each visit. Patient monitoring is often referred to as "patient tracking”. Patient monitoring provides important information for patient management, both of individuals and groups of patients.

Patient management is the relationship between providers on a clinical team and the individual patient over time, assisted by written records. Patient management may also be referred to as "clinical management" and "clinical monitoring".

Program monitoring is the routine tracking of priority information about a program and its intended outcome. Monitoring at the facility, district and national level requires many types of information, including aggregated patient data.

5.2.ART Minimum Data Set

As part of the WHO ART Patient Monitoring Guidelines, a data set has been defined as the minimum required for effective treatment and patient monitoring. This minimum data set (MDS) is a good reference point in evaluating any ARTIS and is included in the functional requirements evaluation of ARTIS. Individual ARTIS may implement this MDS with slightly varying structures to accommodate their system architecture and to make the translation from a paper system (where the MDS was originally defined) to electronic relational database structure.

This section is a summary of the minimum data set description. The full specification is included in the WHO ART Patient Monitoring Guidelines Document referenced above.

Patient monitoring guidelines for HIV care and antiretroviral therapy (ART), WHO, Geneva

Based on the World Health Organization (WHO) HIV patient ART monitoring meeting held at WHO/HQ Geneva, Switzerland from 29-31 March, 2004 http://www.who.int/3by5/publications/documents/artmonitoring/en/

5.2.1.Minimum Data Set

Essential minimum standard HIV care and ART patient monitoring data

This chapter presents the recommended essential minimum standard HIV care and ART patient monitoring data listed in more detail in Annex A. These data are broken down into four categories:

- Demographic information
- HIV care and family status
- ART summary
- Patient encounter information.

The categorized list of data variables provided is for patients that are registered in HIV care and may or may not be on ART, and should be considered when designing records, registers and reports that monitor patients in HIV chronic care settings. Regardless of how the data are collected, it is essential to standardize variable definitions and codes in order to facilitate the accurate analysis of data across facilities, districts and countries.

It is important to identify, early on, which patient-level data are needed in order to manage individual facilities and to monitor and report on HIV service delivery activities. As noted in Chapter 1, patient monitoring data may also be relevant to managers for drug orders and supply forecasting, other planning, quality improvement, and reporting to the district and national level for program monitoring and management.

A more complete description of the variables is provided in Annex A: Standard HIV care and ART data variables and their coding. In addition to the name of each variable, it includes a coding scheme, frequency of collection, and provides guidance on whether or not it is recommended that the variable be aggregated and used for program monitoring at the facility level.

A summary of the list of essential minimum standardized data elements by category:

5.2.2.Demographic information

- Name, sex, date of birth, age at registration, marital status
- Unique ID number, patient clinic ID number
- Address, phone, contact information

5.2.3.HIV care and family status

- Date and location confirmed HIV-positive, HIV subtype
- Entry point into HIV care
- Current health facility, district, district clinician/team
- Treatment supporter(s) name/address/contact information
- If family members/partners: name, HIV status, unique ID number, date of birth/age at registration
- Drug allergies

5.2.4.ART summary

- ART history prior to entry
- ART START date/ treatment cohort:
- Date medically eligible to start ART
- Why medically eligible; baseline CD4, clinical stage
- Date medically eligible AND ready to start ART
- Date medically eligible, ready AND selected to start ART
- Functional status and weight at ART start
- First line regimen
- Original first line regimen (list drugs)
- If SUBSTITUTE within first-line regimen: dates, reasons, and new regimens
- If SWITCH to or SUBSTITUTE within second-line regimen or more: dates, reasons, and new regimens
- ART interruptions: dates, reasons
- STOP ART: dates, reasons
- RESTART: dates
- Transfer In, Transfer Out: date, facility transferred from or to
- LOST (temporary): dates
- DROP: dates
- DEAD: date

5.2.5.Patient encounter information

- Encounter date, whether scheduled or not, next scheduled follow-up visit date
- Months on current regimen
- Current functional status, clinical stage, weight, height (for children)
- TB status, TB treatment start/stop dates
- Pregnancy status, family planning method(s), PMTCT referral/provision
- Possible side-effects (including drug allergies), severity
- New symptoms/diagnoses/OIs
- Laboratory test dates and results
- Prophylaxis: medication, dose dispensed, start/stop dates
- ART dispensed: Regimen code, dose dispensed, (start/stop dates)
- Adherence assessment (pill count, self-report, other) and reasons for both ART and prophylaxis non-adherence

- Referral or link to other clinical or supportive care
- Hospital days since last outpatient visit

5.2.6.Calculating indicators from the patient monitoring data

Keeping careful track of patient data is not only critical for monitoring individual patients, but is also essential in forming indicators for monitoring the progress of the ART program at the facility, district, and national levels. An indicator is a measurable number, proportion, percentage, ratio or rate that suggests the extent of achievement in delivering HIV care and ART of a program, or summarizes the level of some condition in a district or facility's patient population. Six types of indicators can be calculated from the recommended patient monitoring data in these guidelines:

- Indicators related to patients accessing HIV care and ART
- Indicators related to success of ART
- HIV drug resistance early warning indicators
- Other indicators for program monitoring at the facility level
- TB/HIV indicators
- Prevention indicators.

Many of these indicators are crucial for managing and adjusting ART programs at facility and district level. A sub-set of core internationally agreed indicators provides a national picture of the progress of scaling up ART and allows comparisons with other countries, contributing to the global understanding of ART scale-up. These indicators are summarized in Table C.1 Core 8 and 9, and the numerator of Core 7, originate directly from patient monitoring data, while Core 1 to 6 originate from other data sources. A more detailed description of Core indicators 7, 8 and 9 are provided in Annex C.

5.3. WHO HL7ART Data Interchange Standard

The WHO has also organized an effort to define data interchange standards for transferring patient information. Patient information will need to be transferred from one electronic system to another when a patient transfers from one location or program to another. In addition, the standard defines the transmission of data from patient care facilities to management organizations that will use the information to track performance, improve protocols, and report to funding agencies.

This standards effort is known as HL7ART. It uses the HL7 health information data standard and XML representation to define a standard representation of ART data. It is important that an ART system support this data standard to facilitate transfer of patient information for patient care and management.

The following is an extract and summary of the report of the initial meeting of the HL7ART working group.

5.3.1.Introduction

1. In the context of this discussion of data flows, the group focused their discussions on methods of facilitating the transfer of patient records between different EMR systems. It was recognized that this meeting would not necessarily deliver a fully formed solution to this issue, but it could perhaps provide a context to explore how such methods of data interchange might be developed. The original context of the discussion was to understand the minimum data set required for patient care. The minimum data set was defined as the data included in a message packet that is transferred between two EMR systems. From this starting point in the discussion, it became clear that it is important to understand what data are required when a patient moves from clinic to clinic. It was stated that this scenario is common in sub-Saharan countries where patients are very mobile.

Ideally, the discussion suggested that the data message sent between EMR systems should be larger than a minimum data set and should: • include the entire patient record, and • support the mobile patient. If a patient transfers from one clinic to another clinic, what information is required from the source EMR system to handle this patient? The data message exported from the source EMR should include the basic information required to treat these mobile patients. Although, it was recognized by many meeting participants that even a limited amount of data would be useful to any physician. The transfer of patient records should allow patients to move easily from one EMR system to another, and the data structure should reflect the minimum data set required to implement this requirement. Whatever might be defined as the minimum data set should be extensible. An extensible data set would more effectively meet user needs, as it is used, and would also let users provide feedback on the utility of the data.

5.3.2.Logical Data Model

This discussion session examined a proposed logical data model for ART data. The data required to manage ART therapy must be organized in a logical structure to support reporting and analytic needs. The proposed logical data model (or relational structure) developed for ART data, was leveraged from the WHO Interim Patient Monitoring Guidelines (IPMG) for HIV Care and ART. Questions to be addressed included issues related to the inclusion of various ART-specific details in the logical data model. For example, should the data model include the relationship between a patient visit and ART background, should the model indicate the relationship between a visit and a clinic, and is it necessary to support multiple physical clinics in the model? Discussion The ART Logical Data Model is a logical structure that is used to define the data required to manage ART therapy. It is a model for the exchange of minimum patient data and defines how data is stored in the system. The data schema used by the logical data model must reflect the data that is considered important to collect. In the real world, countries are already taking existing data models and extending them to meet their specific needs. The main areas addressed by the logical data model should be:

Collection of information about patient (personal/family info/demographics)

- ART background (for example, when the patient started treatment or changed regimen)
- Place of treatment (for example, the clinic or institution that manages patient treatment)
- Visit information (for example, when the visit was made, what type of visit, and any clinical observations). A visit could be defined as a clinical visit with an encounter with a clinician. However, the definition of a visit is generally moving towards the notion of an encounter to describe any point of contact between a patient and a healthcare worker. Concerning tests, it was stated that a considerable amount of information could be collected about a test, but it is only necessary to know the test result and when the test was performed. Additionally, test results or referrals are not necessarily linked to visits.

5.3.3.Definition of Encounter Data

It is expected that basic encounter data would include date, time, place, and the responsible entity. Not all systems, however, are capable of recording data related to an encounter. For ART monitoring, however, a requirement exists to capture various types of patient encounters, including clinical, hospital, and home visits. An encounter that groups observations will limit the observations to a particular time, date, and place. Additionally, it is valuable to record face-to-face encounters as a separate encounter type.

5.3.4.Drugs in the Specification

In recognition of the complexity of drug orders, a separate discussion was reserved for the handling of drugs in the specification. Drug orders may be composed of standard regimens, substitutions, and individual combinations. Additionally, clinicians may need to track other drugs prescribed for opportunistic infections. Recording drug regimens in a schema can be problematic if they are slightly modified for different patients. With each modification, the schema must be updated to reflect each new combination of medications. The questions addressed by this discussion included issues related to how a drug regime should or could be recorded, and whether drug brand name or generic drug information has clinical relevance.

5.3.5.Version Control and Vocabulary Management

This discussion session addressed the areas of vocabulary management and version control for the HL7/ART message specification.

Vocabulary Management

Vocabulary management is required to define the valid values for the coded data elements used in a message specification or in a logical data model. A common data vocabulary is essential for ART messaging because it facilitates the task of moving data between different systems. For example, two EMR systems cannot share transaction data, even if they can agree upon the format for that transaction, if they do not use the same vocabulary to describe the values exchanged in the coded fields of a transaction message. However, vocabularies used for clinical data are dynamic and can change faster than published specifications.

Version Control

Version control for the HL7/ART specification is necessary to manage modifications as items are added to the specification or deprecated. The HL7/ART specification is currently considered a superset of the WHO Interim Patient Monitoring Guidelines (IPMG), because the minimum data set used to represent ART tracking information is leveraged from the latest version of the IPMG. Consequently, IPMG specification versions could be linked to HL7/ART specification versions by a version map. Before the workshop, it was proposed that the IPMG would have a single sequential version code and that the HL7/ART specification would have a version code consisting of major and minor revisions.

5.3.6.Transport and Security

The security, integrity, and privacy of ART patient information must not be compromised when ART data is transferred between two EMR systems. Because it is feasible that many types of attacks could be targeted at the data transport layer, a discussion session was devoted to this crucial topic. It is important to balance the security of data against the need to maintain an open system that could be implemented across many platforms. If security is too rigorous, it may be problematic to implement it easily. This discussion addressed various issues, including how a message that was passed between two systems could be authenticated and how these messages could be encrypted.

5.3.7.Indicators

The data set defined by the HL7/ART specification should support a number of indicators to facilitate ART program monitoring and evaluation. In this session, Eddy Beck discussed the relationship between patient management and monitoring and program monitoring and evaluation. The questions addressed by this discussion included how to develop indicators and where to develop indicators. For example, indicators could be developed from incoming data and then validated by clinicians. Concerning where indicators could be developed, data collected at local, regional, and national level could be used to develop pertinent indicators.

2 Glossary

Compiler: A computer program that takes source code (written by programmers) and creates machine instructions for computers. These machine instructions cannot be easily deciphered or read by human programmers.

Cost: Information systems have many costs, the cost of the hardware and software is often only a small part of the total cost of the system. Training, implementation, and maintenance costs are usually high. Often the most expensive item is the ongoing operation of the system. The time spent entering, cleaning, moving and analyzing data is your investment in data. Efficiencies in these areas can lead to big savings. The most valuable part of your information system is not the hardware and software but the information itself.

Database: A computer file that stores information. It is organized with columns (fields) and rows (records) for each entry. This is a flat file database. A relational database can be thought of as multiple linked spreadsheets that use an identifier field (such as a patient number) to link multiple records together

Disease Management (Registry): Software that tracks individual patient information in a longitudinal patient record but focuses on a single (or several) diseases rather than being a general purpose EMR. It is designed for DM of specific medical problems.

Electronic Health (or Medical) Record EHR (or EMR): A computer based medical record.

FOSS (or OSS) Free Open Source Software: Software that is distributed with the source code. It is distributed under a license (often the GPL) that explicitly permits you to modify the source and to re-distribute the source code. The source code is the human readable programming language that you can modify to change the behavior of the software. The software can be modified to meet specific requirements. The software is distributed without cost.

The “free” in FOSS has two meanings. First, the software is distributed without cost. Second, you are free to modify the software. The second meaning of free provides the opportunity to customize the software and build local IT capacity and skills. An added advantage is that you won’t be locked out of your software or data by a proprietary format or required upgrades with other software.

General Public License (GPL): A specific copyright license that gives permission to copy a software program and to modify the software, and to publish the modified software as long as it continues to be published under the GPL. It is the most common type of software license used for OSS.

Hardware: Actual physical computer equipment including CPU, memory, hard disks, communications equipment.

HIS (Health Information System): The sum total of all information collection, communication, analysis, and use.

HL7 (Health Level 7): A standard protocol for communicating health information.

LMR (Longitudinal Medical Record): A medical record that tracks individual patients over time through multiple encounters with the health system. It can contain history, exam, diagnosis, treatment, laboratory, and referral information.

Metadata : Information about data. It is a document that gives clear and consistent definitions of the terms used in a data set.

Register: Patient information that is recorded for a single encounter at one point in time. Multiple patients are recorded in a single register.

RHI (Routine Health Information): Information that is derived at regular intervals of a year or less through mechanisms designed to meet predictable information needs.

Software: A general term for computer instructions.

Source Code: Actual computer instructions written by programmers. Source code is compiled into machine instructions that the computer uses. If you do not have the source code, you cannot modify the software.

SQL (Structured Query Language): An industry standard language for creating and maintaining databases.

XML (Extensible Markup Language): An industry standard for representing information transmitted on the Internet that allows the type of the information to be specified.