

AUTOMATION OF INFORMATION MANAGEMENT A HEALTH CARE PROVIDER'S PERSPECTIVE

Michael Montello, PharmD
Cancer Therapy Evaluation Program
Protocol Information
National Cancer Institute



Abstract: *This article provides a step-by-step description of the information technology (IT) development process used by the National Cancer Institute's Cancer Therapy Evaluation Program. The steps covered are: assessing current operations, setting goals and priorities, establishing business rules, knowledge acquisition, modeling, joint application development/rapid application development, beta-testing, independent validation and verification, production, and post-production. Lessons CTEP learned from the experience are highlighted.*

The Cancer Therapy Evaluation Program (CTEP) is part of the National Cancer Institute's (NCI's) Division of Cancer Treatment and Diagnosis. CTEP's mission is to improve the lives of cancer patients by finding better ways to treat, control and cure cancer. CTEP accomplishes its mission by:

- Establishment of the clinical development plans for novel and promising investigational agents
- Preparation and submission investigational New Drug applications (INDs) for these agents
- Review and assessment of treatment protocols
- Assurance of regulatory compliance of clinical trials sponsored by the Division of Cancer Treatment and Diagnosis
- Assess patient safety
- Distribution of investigational agents
- Funding of multi-center programs (i.e., Cooperative Groups).

CTEP's collaborators include clinical sites, agent resources, and regulatory agencies. Clinical sites include cooperative groups, cancer centers,

university hospitals, oncology practitioners, and international organizations. Agent resources are the pharmaceutical industry, biotech firms, and NCI's Drug Development Program. CTEP interacts with its sister federal agencies such as the U.S. Food and Drug Administration (FDA) and the Office for Human Research Protections.

CTEP has more than 160 INDs, which rivals most large pharmaceutical companies. On an annual basis, CTEP reviews approximately 1,000 protocols and protocol-related documents, registers and tracks 10,000 investigators, processes about 75,000 drug orders, and registers about 32,000 patients.

A STEP-BY-STEP APPROACH TO AUTOMATING INFORMATION MANAGEMENT: THE CTEP MODEL

Five years ago, CTEP completed an internal assessment of its operations, and the conclusion was that there were significant inefficiencies. After careful consideration, CTEP management decided to seek technological solutions to improve its operations.

The first in developing a technological solution was to identify the role of each member of the team. As a healthcare provider, my role was to provide subject matter expertise to our technical experts.

Table 1 shows the steps in the process. In brief, the first three steps: 1. assessing current operations, 2. setting goals and priorities for where you need to be in the future, and 3. establishing business rules—can probably be

TABLE 1
Steps in Automating Information Management

- Assess current operations
- Set goals and priorities
- Establish business rules
- Knowledge acquisition
- Modeling
- Joint application development/rapid application development
- Beta-testing
- Independent validation and verification
- Production
- Post-production

accomplished in one or two meetings. Knowledge acquisition (KA) is often the most time consuming step in the entire software development life cycle. During KA, the software developers start to learn about the processes from the subject matter experts. KA is followed by joint application development/rapid application development (JAD/RAD). The system is then validated through beta-testing to assure that it works correctly. Once thoroughly tested, the system can be placed in production. The process continues as additional modifications are identified during the post-production phase.

Assessing Current Operations

In 1996, CTEP conducted an internal assessment of where we were. Most of CTEP's branches did not have any IT system, and those that did were not interconnected. Problems included: multiple non-integrated databases, redundant data collection, inefficient use of resources, inconsistent CTEP nomenclature and standards, and vague and inconsistent policies and instructions. Paper was the major method of communication. CTEP experienced frequent problems with data access and retrieval. Most importantly, the information collected often did not provide the data necessary to address CTEP's mission.

Once the assessment of current operations was complete, CTEP staff identified what information and systems we required to improve its operations (e.g., to identify a minimum dataset to address the science, safety, and regulatory issues).

Setting Goals and Priorities

Step 2 was to establish our needs. In parallel to CTEP's internal review, NCI operations overall were assessed by an external committee (Armitage Committee) to identify potential areas of improvement. One of the committee's top priorities for NCI was to establish "A powerful,

modern, user friendly and secure informatics infrastructure. . . so investigators and researchers can focus on scientific opportunity and reducing the burden of cancer on the American public . . ."

CTEP's internal goals for information management were to:

- Facilitate a flexible and rapid response to scientific opportunity
- Assure patient safety
- Efficiently use resources
- Develop a cost-effective approach to addressing administration and regulatory concerns
- Identify problems early

CTEP's priorities were to:

- Develop an enterprise approach to data collection and storage, with one database for all of CTEP
- Standardize policies, procedures, and nomenclature
- Streamline standard operating procedures
- Reduce and prioritize data (identifying questions that need to be answered; managing and using data, not just collecting it; and using the "dashboard approach" of collecting only really important information)
- Leverage emerging technologies

Establishing Business Rules

Step 3 in the software development cycle is the "how." Ground rules for the team are established. In establishing business rules, identify the core working group with representation from all concerned parties (end users and developers). The core team must be available throughout the entire process. Identify a team leader who is empowered to make decisions for the group. Establish roles and a "no blame" culture. Staff are encouraged to dream big and identify short and long term needs.

Knowledge Acquisition

During knowledge acquisition (KA), the subject matter experts tell the development team what their needs are and how they do business. They transfer their functional knowledge to the development team. Users gain a better understanding of what their needs are, and developers gain a better understanding of the project. Extensive project planning is done including scope, finance, resources, risks, and quality management. These plans are essentially rough estimates because the specific project requirements have yet to be established. Plans will be modified and adjusted once KA is complete and staff have a more realistic sense of the necessary resources. Metrics are also identified to assess if the system requirements are met.

Change is a necessary part of the process. It's critical that changes be tracked because they have a ripple effect on the entire project. Requirements and change management tracking are separate but related activities. Both activities continue throughout the development process.

Developers track requirements and change requests and conduct the impact assessment. The subject matter expert must review tracking documents for completeness, prioritize needs, and identify which requirements and changes to implement. A simple methodology is to prioritize requirements and changes by measuring their necessity (high or low) and difficulty in terms of dollars (high or low). Low cost/high impact projects move forward while high cost/low impact do not. The difficult decisions relate to those tasks that of intermediate cost and impact.

Step five is modeling. Process modeling is a pictorial representation of the business

processes and functions of the project. "As is," (i.e. how you currently do business) processes are compared against "to be," (how you want to be doing business) models. The data model is a pictorial representation of the data elements and how they relate to one another.

Step six is JAD/RAD. During this stage, developers provide a quick and dirty mock-up of what a particular part of the application will look like. It is the developer's best guess at this point in time, based on the information they've gathered from knowledge acquisition. The subject matter experts review and test the mock-up to see if it meets their needs. Through an iterative process, the system requirements are incrementally refined. Each iteration should enhance the capability of the system by building on the success of the last iteration. The process is repeated in parallel with other sub-applications. System requirements are either adjusted or validated through this iterative process.

Step seven is beta-testing, combining the final JAD/RAD iteration of sub-applications into one complete system. The developers test the total application. Then the end users evaluate the system and work with real data to, literally, try to break the system. Requirements and change management continue throughout beta-testing.

Step eight is independent validation and verification. This is an optional step. The key word is independent. Subject matter experts and the developers have vested interest in the project so they may not be able to be as objective as necessary. A third-party reviews the system to see if the requirements have been met. This is a costly step and to date has only been utilized for applications that have a high impact on operations, are very complex, or pose a high risk of harm to people (e.g., patient safety) or property.

The next step in the software development life cycle is production. Program staff use the system with real data for real functions. End-users receive extensive training and documentation. Legacy systems (if applicable) are turned off. Programmers and trainers remain on-site during the transition period to walk people through the system. Tracking requirements and change management continue. Metrics are gathered to see if the system really does meet the needs and requirements that were identified earlier.

In post-production (step ten), requirements and change management continue. It's critical to identify whether changes are niceties or real requirements, and whether they should be made now or wait for Version 2. That is a business decision based on resources and the importance of each change. It's important to note that a good system will require modification. If there are no change requests, this may be an indicator that the system is not being utilized.

Realized Benefits

To date, CTEP has realized many benefits from its IT efforts. These can be measured in terms of a more effective program, enhanced trial safety, and improved overall efficiency. The effectiveness of the program has been improved by empowering staff to make more informed decisions because data that they used to have to get from multiple files in multiple offices are now available at their desktop. Administrative tasks have been simplified to keep the focus on science. Reporting requirements have been streamlined by identifying a minimal informative data set for CTEP-sponsored studies, eliminating redundant user-hostile reports, and minimizing follow-up for clinical and safety data reporting. Other NCI programs are now using CTEP systems.

Trial safety has been enhanced by standardizing the format and database, which promotes adverse event (AE) assessment and comparison across trials. Protocol violations are identified and corrected early before they become extensive. CTEP investigators are kept abreast of all AEs filed with the FDA and assure real time compliance with AE reporting guidelines.

The Program is able to do more with less. For example, over the last five years, the number of full-time staff in our protocol office has remained stable, but the workload has almost doubled, and the number of full-time staff in our Pharmaceutical Management branch has remained stable while orders have increased 2.5 times.

Lessons Learned

Table 2 highlights lessons learned.

- Beware of the iceberg. What you think will be a simple task is not. There is a lot underneath the water that you do not know about.
- Watch for project creep. New requirements and changes will be identified; make sure you reign them in.
- The subject matter experts must be involved in the process or it will not work. Staff and their supervisors must be willing to commit time and resources.
- Know your role.
- Your goal is to manage and use data, not to capture it.
- Consider the 80/20 rule; you will spend about 20% of your time to get 80% of the work done, and then spend 80% of your time to get the last 20% of the work done. You must identify where to cut your losses.

- Focus on what you do, not how you do it. What you do is the key factor in developing applications. How you do things might not be the smart way to do them once you leverage technology.
- Encourage staff to dream big and have a thick skin.
- Be prepared for “what if” questions. During knowledge acquisition, the developers will ask you many questions, including about exceptions to every rule.
- In the management and change process, identify and correct problems early. The cost of change increases logarithmically the further in the development process you are.
- Formalize the requirements tracking and change management process. Track this process in real time. Have a process for decision-making in change management (e.g., these are things we will approve to be changed, these are things we will never do, and these are things we will do later).
- Beta-test the system thoroughly. Have multiple testers who have different backgrounds. Leave ample time between beta-testing and production. You will identify bugs and need to address them. Beta-test with real (not necessarily live) data.
- During production and post-production, do not underestimate legacy migration issues. If you currently have an application and are switching to a new application, there will be many issues. You need to map the data to the new system, consider how to turn the old system off, and so forth. This is a very labor-intensive process. This is a big-ticket item that people do not recognize.
- Establish clear standards for data entry. Minimize the risk of “garbage in, garbage out.”
- Post-production change requests are good; no change requests may indicate that the system is not being used.
- Use metrics to document your successes and failures.

TABLE 2
Lessons Learned

- Beware of the iceberg
- Watch for project creep
- Involve subject matter experts in the process
- Know your role
- Your goal is to manage and use data, not to capture it
- 80/20 rule
- Focus on what you do, not how you do it
- Encourage staff to dream big and have a thick skin
- Be prepared for “what if” questions
- Identify and correct problems early
- Formalize the requirements tracking and change management process
- Beta-test the system thoroughly
- In production and post-production, do not underestimate legacy migration issues
- Establish clear standards for data entry
- Consider post-production change requests good
- Document your successes and failures