WHAT SHOULD AN ERMS PILOT ACCOMPLISH?

In the main

- Clarify users' understanding of the system.
- Verify the adequacy of specifications for the Electronic Records Management System (ERMS)
- Validate the usefulness of the retention assignments within the file structure.
- Verify system response time using a production data base.
- Obtain user acceptance of procedures.
- Validate initial productivity rates and cost/benefit projections.
- Recompute resource requirements.
- Evaluate interfaces with related business functions and information systems.
- Determine effectiveness of training programs.
- Verify that system is technically stable.
- Verify the data base design.
- Verify the functionality of software.
- Identify and address obstacles for the full-scale implementation.
- Produce samples of all outputs.

Prerequisites for a successful pilot

- Define and document all key applications.
- Establish clear cut performance objectives and evaluation criteria.
- Perform detailed records analysis.
- Review existing retention schedules and amend, as appropriate.
- Develop data requirements from required outputs and from source materials.
- Consider security concerns and access permissions/restrictions.
- Perform prototype work sessions with package before customising.
- Finalise system design.
- Test the efficacy of the enterprise-wide taxonomy.
- Complete conversion specifications and procedures
- Complete preliminary user documentation
- Develop quality acceptance methodology for conversion data.
- Develop work package for critical implementation tasks.
- Perform risk analysis.
- Build sufficient time into schedule to fully test the system.
- Test everything

Criteria for selecting a pilot area.

- Mission critical functions with strong leadership
- Range of job types, differing degrees/level of comfort using computers, and both active and infrequent users of records (of the type included in the project) in routine daily tasks.
- Level of files and types of records, representative of all formats that will require that all applications be run/thoroughly evaluated.
- Reasonably large data base to adequately test the ability of the system to manage the records management functions required by the business entity.
- Moderate activity, so as not to overload the system during this testing phase.

- Headquarter and other office locations (with differing mechanisms/approaches for RM and technical support) introduced to the pilot in a phased approach.

Key outputs of the pilot

- Specifications for business system and software and recommendations for further customisation for use in your business entity
- Recommendations for areas to be dealt with before full-scale implementation, including options for use of outside (contractor) staff.
- Updated feasibility study
- System management procedures
- Additional ideas for system development
- Updated resource requirements
- Problem resolution system
- Marketing plan
- Metrics that validate/refute benefits realised from implementing processes.

Lessons learned by those involved with pilot projects in the past can be grouped into three topic areas: Users and usage, Implementation, and Technology.

Users and usage

1. Users want to be involved (e.g., in the solution selection process, file plan development, strategy for business entity-wide deployment of processes) but may not have the time to devote to the project. Designating a Point of Contact (POC) within each group selected to participate in the pilot can keep the pilot project team aware of what is going on with the users and the users involved in the decisions made by the pilot project team.
2. While users want to be involved in policy decisions, they do not want to have to constantly think about business processes. Simplify file plans, simplify and automate organisational forms, use templates, and consider rule-based auto-categorisation to minimise daily decision-making.
3. Staff reacts differently to change. Long-time staff may resist change, but those whose work requires extensive information handling may be more accepting of change than others.
4. Allow users to shape the software and associated procedures to the business processes and accommodate user-generated innovations into the system. Users are afraid of losing control (e.g., the ability to add/remove files from the file plan). For those projects experiencing significant reluctance to adapt to new processes, participant acceptance can be improved by being responsive to requests for change, where possible. Examples from system pilots include making retention codes visible alongside the title of each file; relaxing the file rules to include non-records that users might need for business purposes; and adding transitory files for storing electronic documents participants want to keep for 60 days.
5. Individuals need to see the difference an ERMS makes in their daily routine tasks, but this takes time. Usage of the software grows through the peer pressure associated with business process improvements. Individuals should see advantages in their own work if the implementation effort is to succeed.
6. Users won't use the system until they see benefit; won't see benefit until they use the system. Pre-sell the system by relating benefits to everyday tasks/routine work of staff. Find incentives for use; disincentives for avoidance. As time progresses and team-based work

groups become more prevalent, reluctant staff must use the ERMS to retrieve records generated by others and pertinent to their work.

7. Management support for the project influences the degree to which staff will utilise the system: While there may be strong support from senior management for an ERMS, there should be specific "continuing and visible support from the top for this particular pilot project during the trial period."
8. It takes time to adjust to using a search engine as a retrieval tool instead of navigating file plans.
9. Use the results of the pilot to respond to unrealistic expectations of users.

Implementation

1. The best way to learn is to keep the ERMS pilot projects simple. Don't try to test more variables than a pilot project can manage well.
2. Pilot project teams need to have a solid understanding about what the software can deliver and how it works, making certain that it functions properly during pre-pilot testing before involving users.
3. Make use of the pre-pilot period to prepare groups selected for participation in the pilot. Review the importance of RM to the business entity and the differences between paper and electronic records management. Use this time to develop your POCs or Super Users by including them in training and testing of the system before the formal pilot launch.
4. Determine the information that is important to capture and automate as much of the process for documenting this at the outset.
5. Work the plan but restructure the pilot if the situation warrants.
6. Allot sufficient time for the pilot so that the system is fully tested and the team can evaluate user receptivity prior to business entity-wide deployment.
7. Incremental, phased rollout of the pilot will allow the project team to manage the process more effectively.
8. Make multiple avenues to learning and help available to pilot project participants, noting that:

    - Training is a substantial cost item.
    - ERMS software requires technical training that needs to be reinforced throughout the pilot and beyond.
    - Introductory training sessions followed up by individualised coaching at participant workstations is most effective. Additional learning opportunities focusing on RM concepts and methods are desirable.
    - Providing both "pull" and "push" options for support to pilot project participants allows users the option to learn in the manner most suited to them. Examples of "pull" items include loading user manuals and maintaining FAQs on an Intranet or Web site. Pilot project teams can be proactive in providing help through unsolicited calls to see if participants need further assistance, for example.
    - Having a robust help system in place is essential to successful implementation.

Technology

1. Use of thin client architecture will allow IT staff to deploy the software quickly and easily, with no need to customise the desktop.
2. Avoid macros and integrations with software that requires extensive modification of individual desktops. The macros are unreliable and, in addition to changing frequently, software on the desktop varies with the department and location. Each new version will threaten the connectivity of the macro or integration.
3. Integrate the product at the operating system level. Operating systems upgrade to new versions slower than desktop applications, and there are fewer to integrate.
4. Develop a robust Web-based product that works the same way a client server version of the product would work. Client-server architecture is difficult to deploy.
5. Make the ERMS software appear invisible to the user. Allow the ERMS server and file plan to look like another local drive and directory that the user accesses when saving and opening documents. Let the user perform the "save as" or "open" function, see the ERMS drive, and navigate through their file plan to the desired file. This will boost user acceptance, and it will reduce the amount of training required.
6. When selecting any new software product, do not ask a vendor if their product can perform a particular task, because the answer is almost always "yes." Ask the vendor to demonstrate exactly how the product performs the task and analyse the demonstration from the perspective of the typical user. Remember the bottom line, the user is the most important person affected by a new product.
7. When selecting new ERMS software, be diligent in researching the viability of ERMs products. The corporate acquisition process can cause high volatility in the vendor market, with some products losing support for continued innovation post-acquisition. Integration of product lines may delay promised product modification and improvement.

Recommendations for conducting a successful pilot project.

- Develop a complete project definition.
- Document and obtain approval for all user requirements.
- Complete software design before starting conversion.
- Evaluate applications through prototyping before finalising design.
- Prepare a full set of specifications.
- Work closely with system developer and conversion groups.
- Define all acceptance procedures.
- Address details.
- Allow plenty of time.
- Allow for Murphy's Law
- Study similar projects.