realtimepublishers.com®

The Definitive Guide^{tm} To

Enterprise Change Management

MERANT

Dan Sullivan

Copyright Statement

Chapter 6: Practicing ECM

ECM is more than the sum of silo-based change-management practices. ECM crosses organizational and functional boundaries to integrate assets, coordinate workflows, establish policies, and provide the mechanism to identify and control dependencies. Software development, systems infrastructure, and content management are essential elements of organizations' day-to-day operations. Prior to ECM, these assets were treated as distinct and separate products with their own life cycles and management regimens. Confluences in workflows were more coincidental than managed. Policies were narrowly focused with little consideration for dependencies across functional domains. ECM has changed the framework for change management.

Within ECM, the focus is not on distinct types of assets—such as software, network devices, and documents—but on their common attributes. Chapter 2 developed a model, or framework, for understanding change management at the enterprise level. The model includes:

- Assets
- Dependencies
- Policies
- Roles
- Workflows

Each of these elements is relevant to silo-based change management but is even more useful when combined to span an enterprise-wide ECM framework.

Varieties of Asset Change

Assets are objects that change over time. These include enterprise applications, such as CRM systems and data warehouses; systems infrastructure, such as servers and firewalls; and content ranging from project plans to design documents.

Simple Change-Management Patterns

Assets often have multiple versions. The evolution of those versions varies with the complexity of the asset and the number of participants in the process. Assets can change in a linear fashion; for example, as Figure 6.1 illustrates, an article is written by an author, revised by an editor, and final revisions are made by the author in a linear process.

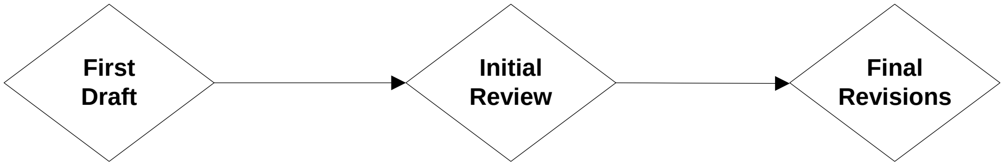

Figure 6.1: Linear changes are the simplest to manage.

Software development for small, single-developer applications also have a simple change-management cycle. Developers often work in a code/compile/test cycle as depicted in Figure 6.2.

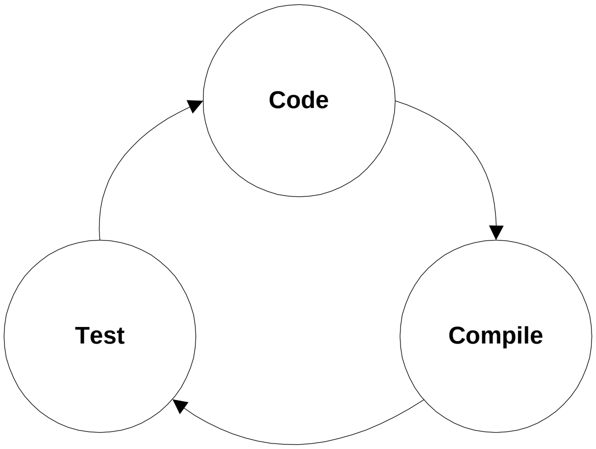

Figure 6.2: Cyclical change patterns occur in small software development efforts.

Complex Change-Management Patterns

Changes to most enterprise assets are more complex than these examples show. Consider a new policy on email privacy from a Human Resources (HR) department. The need for a policy is initiated by some event in the organization, such as the disclosure of a private communication, or as the result of a regulatory requirement, such as the HIPAA legislation governing healthcare records. Staff members from across the organization will contribute to the policy. HR, Legal, and IT provide initial input including objectives, legal requirements, technical constraints, and cost estimates. Figure 6.3 shows the development cycle for this type of process.

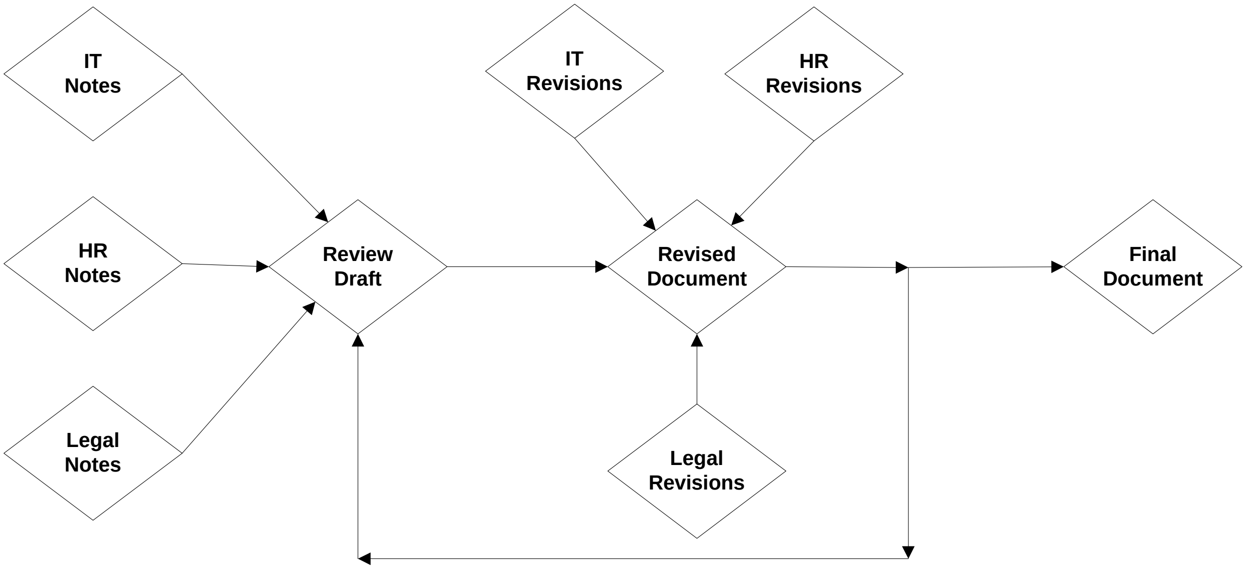

Figure 6.3: When multiple participants and multiple iterations are involved in asset revision, the workflow becomes more complex.

In this example, multiple participants are simultaneously revising a single asset. If each participant revises a version of the document, the changes need to be manually consolidated. Some document-management systems support tracked changes from multiple participants against a single document. In addition to eliminating the consolidation step, these tools provide the basis for effectively auditing changes.

Software development and systems configuration management have similar patterns of complex change. In the case of software, versions vary by function and platform. New versions of applications typically add new features, correct bugs, and improve performance. These changes occur throughout the application. Platform-specific versions typically concentrate changes on low-level services closely tied to an OS, such as file services, process management, and access controls. Coordinating multiple changes across two dimensions—function and platform—further increases the complexity of the management process.

One of the most significant challenges in systems configuration management is the difficulty in duplicating production environments in a development and test environment. Many organizations create network labs to test and configure products, but the cost of implementing and maintaining identical production and test systems is prohibitive. This situation places a premium on understanding the dependencies between systems and their configurations. Configurations often span multiple platforms, organizational units, and functional boundaries. Effective ECM requires an understanding of these configurations in order to foresee the impact of changes to other assets and processes.

Many tools are available to manage each of these change life cycles individually. The limits of these silo-based management tools are motivating the development of cross-domain ECM applications. As with many complex, organizational problems, there are multiple approaches to a solution; however, they are not all equally effective.

Selecting ECM Tools

Selecting an ECM tool entails several steps:

- Assessing an organization’s functional requirements
- Evaluating technical features of ECM tools
- Estimating the ROI

Assessing the Functional Requirements of an Organization

Functional requirements span a range of needs. This section describes those requirements that should be considered for all ECM applications.

Determining Asset Types

Some assets are obvious—such as source code for applications, project documentation, and network hardware; others stretch our notion of assets—such as strategic plans, intellectual property, and best practices. Some assets are managed directly in an ECM system—documents and software, for example. In other cases, we manage information about assets, such as the type and configuration of a network router. The first step in determining functional requirements is identifying all the asset types that need to be managed. For each asset type, understand its life cycle:

- How is the asset acquired or created?
- How does it function with other assets of the same and different types?
- How does the asset change over time?
- What priority is placed on effectively managing the life cycle of this asset?

Next, assess the dependencies that exist between assets and configurations.

Assessing Dependencies

Assets depend upon other assets and stages of workflows. ECM tools should support common dependencies, such as software releases that depend upon successful builds, which, in turn, depend upon versions and branches of source code development. Other dependencies will vary by the specific requirements of a project.

Consider a strategic marketing plan. It depends upon product release schedules, which, in turn, depend on the outcome of production schedules, quality assessments, and preliminary marketing research. These schedules and this research are stored as documents that are produced, in some cases, by staff and consultants, and in other cases, by an application, such as an ERP tool. By clearly identifying these dependencies, managers can pinpoint bottlenecks and develop risk mitigation strategies to adopt a strategic plan for unexpected events.

An ECM tool should effectively model a variety of dependencies. In particular, it should:

- Report on assets and workflows affected by a change to an asset
- Allow for multiple dependencies between assets
- Include information about timing dependencies
- Rank priorities of dependencies

With an understanding of both common dependencies and dependencies specific to an organization, the next step is to understand workflow processes.

Understanding Workflows Within an Enterprise

Assessing workflows within an organization is similar to assessing dependencies. There will likely be common workflows, such as in software development, testing, and releasing, as well as organization-specific workflows. When considering idiosyncratic processes, do not limit the discussions to the traditional change-management domains of software, systems, and documents. Areas such as strategic planning, budgeting, product development, and other high-level business processes lend themselves to the benefits of change management.

The discipline of ECM is appropriate to a broad range of operations. Well-designed ECM tools will lend themselves to processes outside the traditional areas of change management.

Evaluating Access Control and Security Requirements

All enterprise applications must address security. Some ECM requirements are similar to those found in other applications. ECM tools should integrate with single sign-on (SSO) services and utilize directory services such as Microsoft AD or other LDAP services. The tool should also support ECM-specific requirements such as controlling users' ability to:

- Create, delete, and modify assets
- Define and edit dependencies between assets
- Create and modify workflows
- Define triggers, such as notifications that an event has occurred
- Change security and access control settings

ECM tools should also enable administrators to assign users to groups, which, in turn, are assigned privileges. In addition to their use in controlling access, roles are also used to define which users can execute steps in workflows, as Figure 6.4 shows.

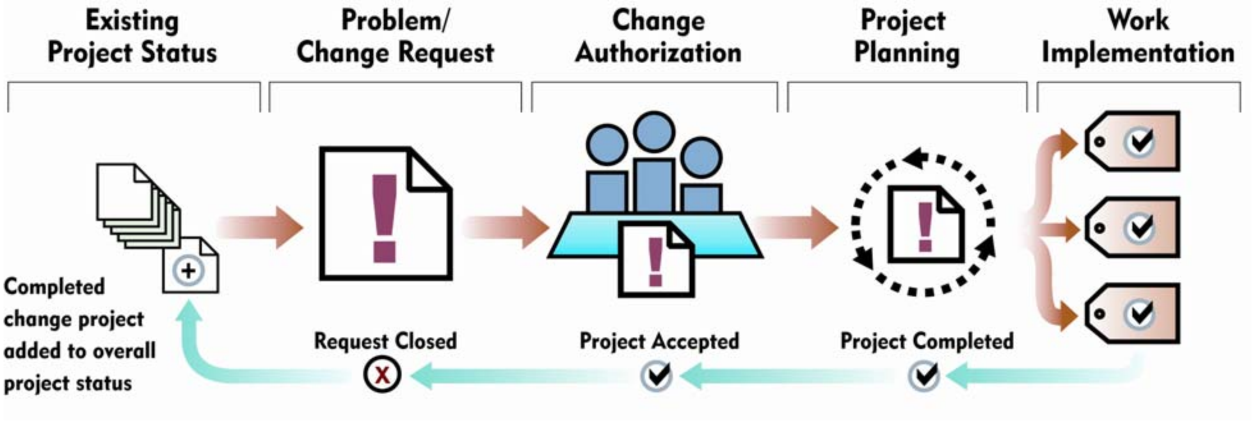

Figure 6.4: Roles serve multiple purposes in ECM applications, including controlling the execution of workflows.

The combination of assets, dependencies, workflows, and their access controls are the elements of ECM, but functional requirements should consider two additional aspects as well: information delivery and legacy application integration.

Delivering Information

ECM systems maintain a wide range of in-depth information. Assets come from across the organization, dependencies span traditional boundaries, and workflows can entail large numbers of users. Information about these assets, commonly called *metadata*, is also found across the organization and in varying forms. Add to this the multiple versions and branches associated with assets, and you can see how an ECM repository can quickly grow with detailed information. This information growth is a doubled-edge sword. On one hand, having ECM details in a single source provides one point of access for information. It also provides the means to integrate information across domains, projects, and operations.

On the other hand, the ECM application can suffer a similar problem to the World Wide Web—with so much information available, it's difficult to find exactly what you need. The key to successful information delivery is identifying specific items rapidly and keeping related items easily accessible. This goal is accomplished in several of ways, including:

- Search
- Navigation
- Visualization
- Structured reports

Search should include basic lookup functionality based on named entities, such as "Finance Department Data Mart—Invoice Extraction Script" and "Boston WAN Segment—Router 3." It should also include metadata searching—searching by metadata attributes, for example "all routers in the Boston WAN Segment" and "all data extraction scripts modified in the last 30 days." This type of targeted information retrieval is best suited when you need information about a particular asset, process, or dependency. However, a broader view is sometimes needed.

Navigation allows users to move from general categories to specific instances and vice versa. For example, navigation enables users to start at a high level, such as project documentation, and move down a hierarchical scheme to IT projects, through business intelligence projects, down to a Finance department data mart documentation (see Figure 6.5).

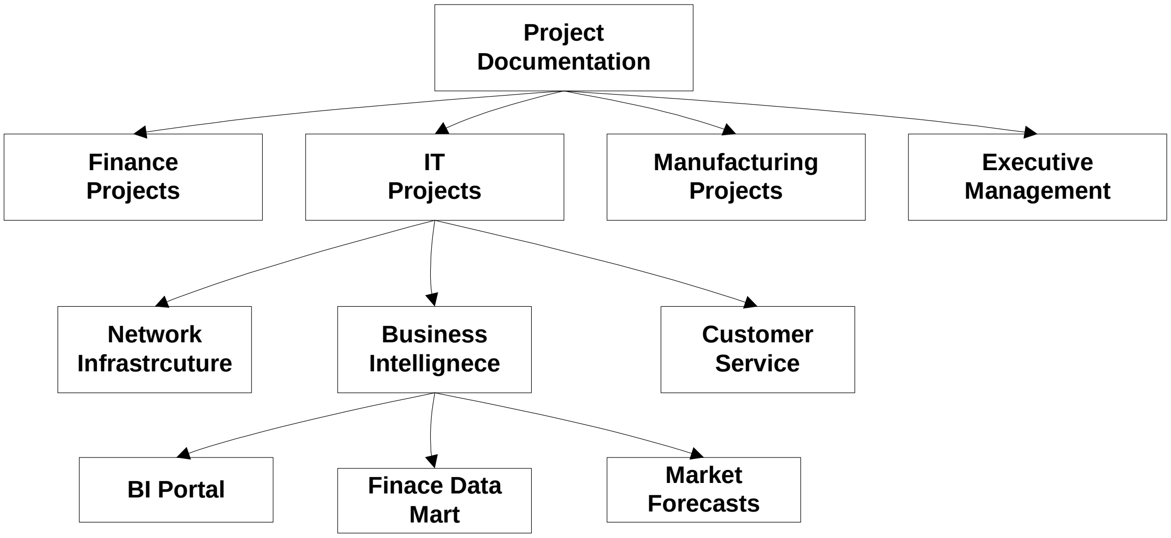

Figure 6.5: Hierarchical navigation allows for information retrieval based on topics rather than search terms.

We know from enterprise content management that no single hierarchy, or taxonomy, will meet all navigation needs. Generally, multiple taxonomies are required. In ECM, hierarchies could be designed around projects, assets, departments, or time periods. Specific taxonomies do not need to be designed prior to selecting an ECM tool. The goal should be to use a tool that supports multiple modes of access to repository data.

Search and navigation are based on terminology. Navigation provides more contexts than search, and multiple taxonomies provide greater depth to context than a single taxonomy can. In some cases, even this information delivery method is insufficient to quickly grasp the context of an asset or workflow. In those cases, visualization tools are called for.

Visualization is a powerful tool for understanding the overall structure and breadth of a topic in an ECM repository. Ideally, a visualization tool provides a picture of the relationship between elements of an ECM repository, such as select assets and the dependencies between those assets. Visualization is also essential for quickly understanding the nature of a workflow (see Figure 6.6).

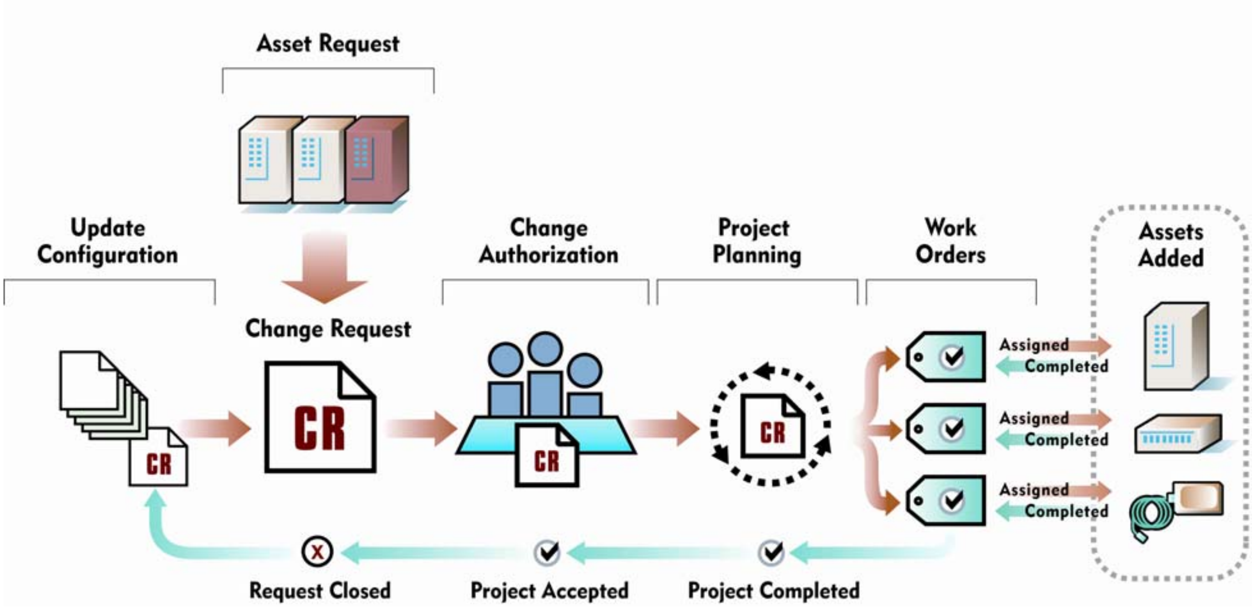

Figure 6.6: Visualizing complex workflows aids in user understanding.

The last information-retrieval technique is structured reporting. Traditionally, these reports support operational management. Reports serve several functions, including:

- Describing in detail the status of particular assets or configurations, including bill-of-materials.
- Identifying exceptions to a plan
- Summarizing the overall status of a set of assets
- Listing dependencies
- Identifying workflows that manipulate an asset

No one information retrieval and reporting tool can meet all enterprise needs. ECM tools should be flexible enough to allow the integrated use of multiple tools. Visualization could be used to quickly identify the main elements of an enterprise process, search and navigation can isolate particular elements within that process, and structured reports produce detailed descriptions of individual assets.

Functional requirements tend to focus on the intersection of end users and tools. Technical requirements shift the focus to the underlying implementation issues.

Understanding Technical Requirements and Options

ECM tools will function in the IT infrastructure of an organization. For most enterprises, that means a distributed, heterogeneous environment. To meet the demands of functional requirements while fitting into an existing environment, ECM tools should be judged on several technical requirements, particularly:

- Platforms supported
- Scalability
- Integration
- Systems administration

Work with Existing Platforms

IT infrastructures have evolved over the past three to four decades. It is not unusual to find latest-generation ERP systems hosted on state-of-the-art UNIX servers residing on the same network as 20 year old IDMS database applications running on mainframes designed a generation ago. ECM applications should work with the technology that currently exists within an organization. To do so, the tools must support multiple platforms, including, at least:

- Web
- Client/server
- Mainframe

Of course, each of these is a broad category representing multiple variations. Web platforms include HTTP, HTML and related scripting languages, and other protocols, such as Web Services protocols and WebDAV, which extends HTTP to allow users to manage files and collaborate over the Web.

> 🕮 For more information about WebDAV, see http://www.webdav.org. You can also find technical information provided by the Internet Engineering Task Force (IETF) at http://www.ietf.org/html.charters/webdav-charter.html.

Web platforms must also include portals, which are frameworks for integrating applications and services in a Web environment. Applications are deployed in portals through components called *portlets*. Portlets vary from vendor to vendor, but several standards are under development that should eventually provide a common portal base for ECM developers and others.

📖 META Group's "The Emergence of Portlets" offers a high-level perspective on the future of portlet adoption and is available at http://itmanagement.earthweb.com/it_res/article.php/1146261. Technical details about Web Services for Remote Portlets (WSRP) are available at http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf.

Client/server is a two-tiered model based on client machines running visual interfaces and business logic and servers providing database services. Windows, Macintosh, and UNIX-based systems can all serve as clients. Each of these requires different versions of client software. Web-based deployments offer more flexibility for both developers and users, but some functionality is still easier to deploy on client machines. Client/server is no longer the dominant architecture for distributed applications, but ECM vendors will need to support it along with other deployment schemes.

Mainframes continue to meet the high-performance, high-transaction needs of large enterprises. Although there was discussion about the demise of the mainframe during the heyday of the client/server architecture and the early days of the Web, the mainframe has continued as the backbone of large data centers. Today, companies are leveraging the high performance, security, and scalability of mainframes to deliver e-business services as well as traditional batch and basic transaction processing. Mainframes are vital to enterprises and ECM applications need to accommodate this platform along with others.

Scaling to Perform

ECM applications must be designed for high performance. These systems track low-level details about changes to code baselines, the results of builds, editorial changes to documents, routing logic in workflows, and other asset, dependency and workflow issues. Several design elements provide the needed scalability:

- High-performance relational database
- Designs optimized for the network environment
- Data model optimizations
- Pooling distributed resources

ECM applications generate large amounts of raw data that must be efficiently managed. The best tool for that job is a relational database management system (RDBMS). Relational database application designers and RDBMS vendors have developed many techniques to deliver rapid responses to large numbers of concurrent users. These databases also provide security and access controls for ECM applications. Major database vendors, such as Oracle, IBM, and Microsoft, have deployed distributed versions of their databases, which are useful from an ECM perspective as well.

ECM applications generate a lot of network traffic. Even simple changes to a document will create traffic to save the document and record the metadata about the change in a version-control system such as a document-management system. If the change triggers an event in a workflow, additional traffic is generated to send notification messages and route the document to the next party. These events have secondary effects, such as checks against access control databases, generation of audit information, and replication of changes to failover servers. Databases are tuned to optimize performance within the data storage realm. ECM core applications should be designed to minimize unnecessary network traffic. For example, they can cache data locally and to minimize redundant data retrieval.

Data storage in an ECM tool should be designed to read and write for high volumes of data. For example, to minimize the time required to search document metadata, the metadata should be stored separately from the documents. Metadata is generally much shorter than its corresponding document, so it can be stored in data structures optimized for small amounts of data. This scheme would not work for documents, however.

Documents can be stored either outside the database in a file system, in which case they do not incur the additional overhead found in RDBMS or in the database in a storage scheme designed for retrieving large blocks of data. Retrieving data from disk is relatively slow compared with other operations. Retrieving fewer but bigger chunks of data is one way to optimize large data access operations. The same optimization would not work for metadata retrieval because of the smaller size of the data. Additional unnecessary time would be spent accessing data.

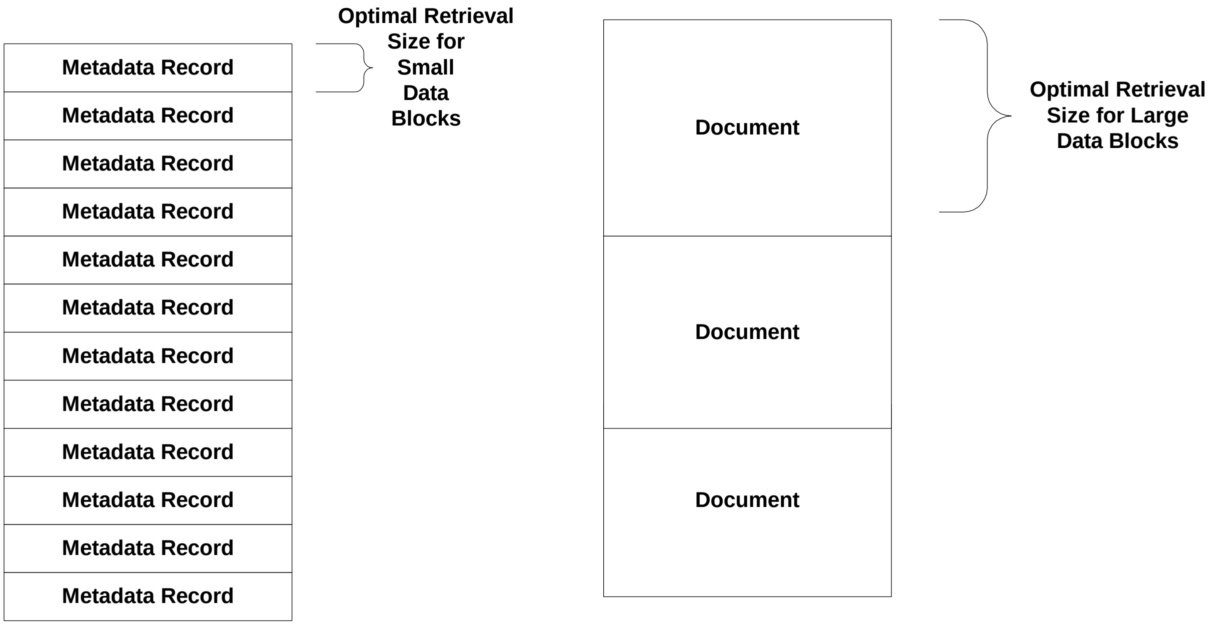

Figure 6.7: Data storage is allocated based on several factors, including average size of data records. Small records are stored with small allocations, or block sizes, while large records require larger data blocks.

When users interact with an application, the application often creates a process dedicated to that user. This design is simple and works well when the number of users is limited. As the number of users grows, the overhead involved in creating, managing, and deleting processes diminishes overall system performance.

One method for addressing this problem is to pool processes. Rather than create separate processes for each user, an application manages a set, or pool, of processes that are shared among users. This configuration reduces the demand on the server and improves overall performance. Additional performance gains are realized when the ECM application supports pooling.

Integrating ECM Components

One of the key advantages of an ECM system compared with silo-based change-management systems is the ability to analyze dependencies and workflows that cross traditional change-management boundaries. For example, a change in a policy for creating baselines in a software development project should trigger a change in both the source code version control system and the document-management system used by developers.

Reporting should also cross boundaries. If a server is scheduled for an upgrade, which applications are affected? Are any of those applications involved in critical workflows? Obvious workflows are easy to identify. An accounting system generates invoices and processes payments, but it might also provide reference data to an enterprise data warehouse that is updated nightly. To effectively meet ECM requirements, metadata about assets, dependencies, workflows, and policies require tight integration. Optimally, this integration is at the database level; less effective is interface-level integration.

Metadata integrated at the database level share a common data model in an integrated repository. ECM components are organized according to their type, such as assets and dependencies, rather than by their functional domain, such as software development and systems configuration management. The advantages of this approach are twofold. First, it provides an environment in which new domains can be added without fundamentally changing the ECM application. ECM might initially focus on software development, systems configuration management, and document management but other domains (for example, strategic planning and marketing campaigns) can also benefit from ECM practices.

Second, a consistent model allows for a single set of reports and analysis tools to work across functional domains, minimizing user training. More important, it allows for cross-domain analysis not available in silo-based approaches (see Figure 6.8).

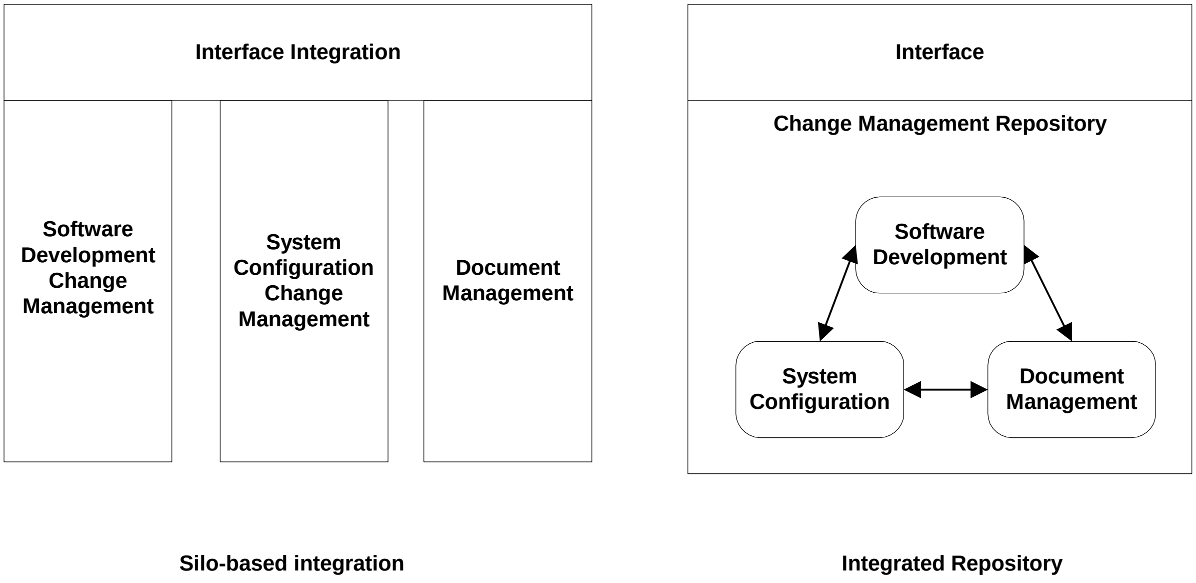

Figure 6.8: Integration at the data-model level allows for cross-domain analysis. Silo-based designs are challenged to perform similar analysis.

ECM applications should integrate with tools that support domain-specific functions. Software developers often use testing tools to generate test scripts, run suites of tests, and analyze the results. Content producers use desktop publishing, image, and video editing programs and other tools that need to function with document-management systems. When available, ECM tools should use open standards, such as Web Services protocols (for example, SOAP), to ensure the widest possible level of integration.

The capability to integrate data within an application and share it with other systems is a fundamental product of system design. ECM applications that will adapt to the diverse and changing needs of an organization are built on integrated repositories and open standards.

Administering ECM Applications

Another functional requirement to consider with ECM applications is systems administration. ECM administration factors include:

- Support for SSO
- Client integration with portals
- Integration with LDAP
- Backup and recovery procedures
- Server failover

- Support for clustered servers
- Support for an organization's standard RDBMS
- Routine maintenance procedures
- Enterprise application integration
- System integrity and security

Of course, these and other administrative issues underscore that fact that ECM applications are subject to the same systems configuration management issues as other enterprise applications. Platform support, scalability, integration, and systems administration issues are central to the technical requirements of ECM applications. The next section will address some of the functional requirements of these tools.

Estimating ROI of ECM Systems

The benefits of using an ECM system range from the obvious, such as saving software developers' time through version-control practices, to the obscure, such as preventing a fine from a government agency for not producing audit records that certify correct procedures were followed in a regulated process. Estimating the ROI of an ECM system requires an understanding of:

- Hard savings
- Soft savings
- TCO
- Regulatory requirements

The first three items are used in ROI and related calculations. The fourth, regulatory requirements, trumps the others. Regardless of the return or payback period, some industries need ECM to remain in compliance with regulations.

> 🕮 For more information about the details of ROI and related calculations, such as Net Present Value, Internal Rate of Return, and Payback Periods, see "An ROI Primer" at http://www.zdnet.com.au/newstech/enterprise/story/0,2000048640,20267881,00.htm and "ROI Guide: Internal Rate of Return" at http://www.computerworld.com/managementtopics/roi/story/0,10801,78524,00.html.

Hard savings are definite reduced costs accountable to the use of an application. With regard to ECM applications, hard savings include:

- Eliminating software license and maintenance costs for silo-based systems, such as separate version-control and document-management systems
- Eliminating the overhead of maintaining ad hoc data interchange mechanisms for poorly integrated point solutions
- Consolidating servers and other hardware that had supported silo-based systems
- Reassigning support and administrative staff responsible for eliminated systems

Soft savings are harder to quantify. They do not entail tangible savings, such as a license cost, but typically tend to describe productivity gains, such as:

- Improved programmer productivity because of better access to requirements and dependency information
- Better project management through enforcement of policies, especially with regard to version control and baseline management
- Reduced time to create test suites, automate the execution of those suites, and distribute the results to appropriate developers
- Better identification of dependencies between network devices and fewer conflicts between components
- Less duplication of documents, fewer email attachments, and better control over official versions of documents

Cost is the third factor in the ROI equation. The term TCO has gained popularity to describe both the initial costs of acquiring an application, such as software licensing and server costs, and ongoing costs, such as maintenance and support. Again, ECM applications are like other enterprise-scale systems and incur costs beyond the initial expenditure, for example:

- Staff support including database administrators, network administrators, and Help desk personnel
- Staff training
- Maintenance contracts
- Hosting and infrastructure costs

ECM applications can extend the benefits of traditional change-management systems. Success is dependent on matching the needs of an organization with the appropriate application, deploying the application in a cost effective manner, and introducing change-management practices in ways that maximize user adoption. The first step in that processes is to understand the options available to you.

Assessing the ECM Market

ECM products are evolving from multiple points in the IT market. Some have a software-development orientation, others are more focused on document management, and still others approach the problem from a systems configuration perspective. In some cases, vendors offer near-ECM systems such as Product Lifecycle Management (PLM) applications.

The diversity of approaches underscores the complexity and breadth of ECM. It also demonstrates that there is no “domain free” perspective on ECM. Products in this space evolve from earlier change-management products. One way to assess these products is to examine:

- The domain or domains that orient the product
- The degree to which they can represent assets, dependencies, and workflows outside those original domains

These two issues are the subject of the next sections.

Approaching ECM from a Software Perspective

Consider, for example, Merant Dimensions. This ECM application has strong support for software configuration management (SCM), including baseline management, build management, and release management. Dimensions integrates with third-party SCM tools as well as other specialized tools such as Merant Mover for deploying applications across the enterprise and Mercury Test Detector/Defect Tracker for supporting quality assurance procedures. Although the roots of Dimensions are clearly in the SCM arena, the architecture is flexible enough to support other areas of ECM.

Dimensions tracks assets, dependencies, and workflows. Policies and roles ensure that assets are managed properly according to an organization's needs. Equally important, information about change-management objects are stored in a single repository ensuring the broad integration of assets that is demanded in ECM environments.

📖 For more information about Merant Dimensions, see http://www.merant.com/Products/ECM/dimensions/home.asp.

Products will vary in their support for non-software assets, such as documents, and their ties to particular methodologies. Support for methodologies can enhance software development. However, they should not be so fundamental to a tool that they hinder workflows or asset models outside software development.

Approaching ECM from a Document-Management Perspective

ECM applications can also have a document-management foundation. These systems expand beyond conventional document management to support the concept of enterprise document of record. This is more of a logical concept than a single physical document. For example, the enterprise document of record for a new product might consist of content stored in an ERP system, a content-management system, a PLM system as well as a CRM system. Supporting enterprise document of records in manufacturing environments improves collaborative engineering, product development, resource scheduling, and production operations.

Another example is the concept of a patient record. Patient information is distributed across a range of systems in contemporary healthcare systems, including clinical records systems, pharmaceutical systems, and image management systems for X-ray, CT scans, and MRIs. Under HIPAA legislation in the United States, much of this information is considered protected health information and is subject to strict regulations. Effectively managing this type of distributed "record" requires an enterprise-management perspective like that found in ECM.

Approaching ECM from a Systems-Configuration Perspective

Vendors can also tackle ECM from a base in systems configuration management. This type of change management starts with a focus on hardware, software, and network assets as well as on systems configurations and monitoring and dependencies between systems. These tools have strong support for automatic discovery of server state, storage usage, processor load, and other operational measures.

Some vendors offer specialized change-management tools that point to new development areas in ECM. These products do not fall into the ECM category because they are silo-based approaches, but they do provide functionality that ECM vendors will have to eventually support. For example, database change-management tools identify dependencies of proposed changes and evaluate the impact of changes on tables, views, stored procedures, and other database objects. They might also include support for migrating and converting data as necessary to implement changes.

Understanding the Current State of the Market

ECM is a new and evolving technology. Its roots are in specialized change-management domains such as software development, document management, and systems configuration management. Tools in this market have evolved from strong positions in as least one of these domains. In some cases, such as with Merant Dimensions, there are strong roots in both software development and configuration management.

As ECM continues to develop, we can expect vendors to improve support for the change-management areas outside their original focus. A key differentiator will be their approach. If vendors choose a unified, single repository approach with data models designed for high-level, cross-functional entities (such as assets and dependencies), the tools will maximize the benefit to users. If silo-based applications are collected in a single interface without the deep integration provided by a single repository and common data model, users will not realize the full benefits of ECM.

Introducing ECM to an Organization

Introducing ECM to an organization requires preparation. The first step is assessing the organization’s readiness for ECM. This step is followed by a series of steps to introduce the concepts of ECM and determine the best approaches to integrating ECM with ongoing operations.

Assessing the Readiness for ECM

Realizing the benefits of ECM depends on users adopting ECM practices as part of their day-to-day activities. Many staff, contractors, and consultants are already under pressure to produce more with less, meet tight deadlines, and take on new responsibilities. To understand better how ECM will be accepted, consider several questions.

Are Change Management Practices Already in Place?

Many software developers already use version control, testing, and quality control systems. Extending these practices to include better document management and systems configuration considerations should not be difficult. If basic version-control practices are not used, even if the systems are available, adoption will be more difficult.

If basic change-management systems have become “shelfware,” determine what caused the poor acceptance. Does the tool not support effective workflow? Is the interface difficult to use? Does the tool only solve part of a problem? If these are the cause of the poor adoption, ECM could provide the solution.

Look for similar situations with document management and systems configuration management. Are the repositories up to date with the latest documents and components? Is the metadata associated with assets complete and useful? Users might choose not to add metadata when saving and modifying documents. This shortcoming indicates that users are not aware of how useful the metadata actually is and probably do not know how to use the additional information for information retrieval. The repositories have become archives rather than operational support tools. Again ECM may address these issues. If a tool fits the way staff works and developers, content providers, and network administrators know that information they provide is actually used to improve operations, the system is more likely to be adopted.

Is There Executive Sponsorship for ECM?

Executive sponsorship is a key factor in organizational readiness. Sponsorship cannot stop at approving a budget; it must follow through into the deployment and adoption of ECM. Executive sponsors should:

- Define the role and benefits of ECM in the organization
- Outline a long-term plan for adopting ECM in an incremental fashion
- Understand the risk factors, such as potential migration and integration problems
- Define management practices based on ECM
- Identify ways ECM tools are used to ensure compliance

The greatest benefits will be realized with ECM if it is used across departments, projects, and subject areas. Executive sponsorship is essential to meeting that objective.

What Problems Are Solved by ECM?

It is important to understand the drivers behind ECM use in an organization. Some may be clear (for example, companies need ECM to remain in compliance with government regulations). In other cases, users of change-management systems already understand the benefits of the discipline and want to extend its reach. ECM technology can address a variety of issues. One source of information about such issues is post-project assessments. Review these assessments with a particular eye to understanding:

- Communication problems within the team
- Code-management problems
- Unusually long time to resolve bugs and requirements issues
- Failure to implement required functions
- Late discovery of integration problems
- Poor communication on overall progress of the project
- Executive sponsorship
- Introduction of ECM in response to poor performance, compliance

The benefits of ECM are not limited to projects.

Are Daily Operations Constrained as the Result of a Lack of ECM?

Infrastructure management can be especially difficult without centralized change management. Tell-tale signs of the need for ECM from a systems configuration management perspective include:

- Projects that are hindered because of a "don't change the infrastructure, something might break" attitude
- Network administrators using multiple tools for monitoring system performance, configuration, and status but without the ability to integrate the information gathered from these systems
- Lack of any formal change-approval process result in authorized production downtime
- Inability to report on the status and configuration of servers and other hardware used by a particular application

Assessing an organization's readiness for ECM entails understanding past experience with change management, operational issues that will benefit from ECM, and the level of executive leaderships prompting the project. When it is clear that there is a need and support for ECM, it is time to develop a deployment strategy.

Deploying ECM

There is no magic formula for deploying ECM, but you can follow some general guidelines:

- Start with small groups and deploy incrementally. Ideally, a rollout starts with teams or departments already familiar with change management. The initial focus with these groups would be to extend the scope of their change-management practices. For example, having software developers use document management and system configuration management along with code-management features.
- Start with midsized projects that need ECM but are not overly constrained by tight deadlines or budgets. When the pressure is on, teams have a tendency to abandon new practices and resort to familiar habits.
- Find cross-departmental projects that require coordination. ECM can be especially useful with distributed teams.
- Set expectations that project managers will manage from ECM content. Make the system the official record of source.

Organizations will adopt ECM at different rates and for different benefits. Those that succeed will start with a clear understanding of the business objectives and a realistic deployment plan based on an incremental deployment.

Benefits of ECM

ECM introduces several types of benefits to organizations, including:

- Better alignment of IT operations and business needs
- Improved IT quality control
- Improved ROI on software development and management
- Improved infrastructure management and security
- More predictable project life cycles

Aligning IT with Business

Managers can better understand business needs and IT constraints when information about both is readily available. Single repository change-management systems provide reporting on technical aspects of a project, such as the code development life cycle, along with information about the sometimes dynamic business requirements. ECM provides the basis for a single, holistic view of a project, not just domain-specific aspects of it.

Improving IT Quality Control

ECM also improves on IT quality control, especially when ECM applications integrate with testing and deployment tools. Merant Dimensions, for example, works will tools for building applications from source code, executing test suites, and deploying applications to production environments.

Improving ROI on Development

As noted earlier, ECM practices can improve factors that influence project ROI. The practices have a direct effect on reducing lost changes to code, poor version management, and policy enforcement. Managing to requirements, effective bug tracking, and providing transparent reporting on project status indirectly contribute to project success rates.

Improving Infrastructure Management

When software, business process, and systems configuration management information is integrated, organizations can better manage the dependencies between assets. Improved infrastructure management will lead to fewer unanticipated consequences when assets change.

Predicting Project Life Cycles

Finally, ECM applications track detailed information about the course of project, the life cycle of assets, and the dynamics of organizational infrastructure. This data becomes a foundation for understanding typical project life cycles and daily operations in the organization. More data leads to better analysis, and better analysis supports better management.

The Future of ECM

The fundamental elements of ECM, assets, dependencies, workflows, policies, and roles will not change with time. Objects of change management and the breadth of ECM will be the focus of ECM evolution. To complete this guide, let's explore the future of ECM.

Evolving Objects of ECM

The objects of change management are evolving even as ECM emerging. Software development is adopting the new paradigm of Web services. Records management is no longer limited to information in a single database but span systems within an organization and beyond.

Dynamics of Web Services

In the case of software development, Web services present new problems to change management. Traditional mainframe and client/server development assumed relatively controlled execution environments. Applications were fairly self-contained and made limited use of OS services and code libraries. If there was a problem with a build, it could usually be traced back to a problem with the application source code. Web services are changing that model.

Applications designed around Web services invoke programs on other servers, inside and outside the firewall boundary of the organization. Web services use common protocols for defining services, directories for finding services, and exchanging messages. Keeping provider services in sync with consumer services is a challenge because they can change independently of each other. In many cases, this is one of the appealing features of a Web service. For example, if you have an e-commerce operation, you might subscribe to a tax calculation Web service. The service calculates state and local taxes on customer purchases. Tax rules and rates can change without any changes to your application, only the tax Web service needs to change. Consumer applications of the service are unaffected. If a change in tax code requires more information to calculate tax (for example, an indication of whether the product is clothing), the consumer service will need to change the information sent to the provider service. Detecting and managing changes in large numbers of Web services will present new challenges to ECM applications.

Composite Objects and Logical Records Management

Logically related data is becoming highly distributed. Patient healthcare records and financial services customer records are just two examples of regulated information. The fact that each of these is composed of multiple records, or assets, and must be managed as a logical unit will force advancement in ECM models.

We cannot think of assets as autonomous units, such as a source code file, a word processing, document, or a row in a relational database table. Assets are composite objects. Each component is linked to the others but has a distinct life cycle. Separate versions may exist and be considered valid simultaneously.

When a patient record is transferred from one doctor to another, it might include only part of the original logical record. There should be some indication with the original that a segment of the record was sent to another location. That new version may combine with segments from earlier versions of the same patient's information. Tracking the composition process of logical records and managing the effects of independent life cycles are especially challenging.

Evolving Processes of ECM

ECM encompasses traditional change-management domains: software development, document management, and systems configuration management. Other business processes will be quickly drawn into the fold:

- Business process re-engineering
- Research and development
- Product development
- Manufacturing processes

Successfully expanding the scope of ECM will depend on new tools as well as better integration.

The Evolving ECM Toolset

As ECM models evolve, we can expect metadata standards to emerge. These standards should provide a basis for describing abstract objects that are shared across ECM tools and components. The complexity and breadth of ECM repositories will also trigger the development of search and navigation tools, especially visualization applications to aid ECM administrators and users. Analytic operations, such as link analysis for detecting wide-spread dependencies, will also emerge.

ECM is the evolution and merging of practices that have proven their value for decades. Like its predecessors, ECM will meet the immediate needs of organizations while remaining flexible and adaptive to the dynamic nature of today’s enterprises.