KINETIC DATA

Your business. Your process.

Enterprise Request Management: How to Get Started

Introduction

Enterprise request management (ERM) is a business-efficiency strategy that takes a holistic approach to centralizing and automating business processes and service requests. ERM helps to ensure first-time fulfillment, which means lower costs and happier customers.

With ERM, a single Web-based request portal replaces manual processes, and disparate departmental online request form front-ends. An automated task management "backbone" application then manages approvals, scheduling, and fulfillment by securely communicating with and between existing departmental software platforms used by IT, HR, finance, operations, facilities and other areas.

Though the scope of ERM is broad, implementation needn't be excessively difficult, expensive, time-consuming or disruptive to operations. ERM can be implemented incrementally by starting with one or a few processes and expanded as the concept proves its value across the enterprise. Also, ERM utilizes many of the systems you already have in place, so additional technology and training investment requirements are often modest.

This white paper defines the ERM concept and benefits, describes how to begin implementing an ERM strategy, and provides concrete steps to guide you through the implementation process.

How ERM differs from departmental service request management

Service request management (SRM) functions vary in their maturity and effectiveness. While many large enterprises have implemented online IT service catalogs, according to a recent report from Forrester Research, these fall into one of three levels of maturity.[1] At the most basic level, organizations are focused on "delivering IT services to consumers through a standard set of choices and/or requests." At level two, the service catalog automates enterprise services, and at level three it acts as a "service broker."

The ERM concept operates at the intersection of these levels. For example, new-employee onboarding is a process—but one that involves numerous functional areas within the enterprise, including IT, human resources, accounting, facilities, office management, and potentially other functional groups. ERM acts as the "glue" that connects and orchestrates the actions of back-end applications (systems of record) in these areas. It automates processes where possible to reduce the time and cost of request fulfillment, as well as to provide a highly satisfying customer experience. Some of the differences between ERM and service request management systems include:[2]

- With ERM, a single, intuitive request management portal presents services delivered by multiple functional areas within the enterprise, rather than requiring users to learn and navigate different user interfaces and processes for different functions. ERM provides a single point of entry for service requests and is by design flexible and extensible, making it easy to add new service processes.
- ERM is far more than just logging a ticket online. Rather, it is designed to automate most if not all of the tasks within the service request fulfillment processes—centralizing request management, scheduling, approvals, analytics, Service Level Agreement (SLA) tracking, status, chargebacks, billing and reporting—by linking to and coordinating data updates within the software platforms (systems of record) that enterprises already have in place to handle these tasks.
- ERM is "function-agnostic"—it can apply to any type of service (IT, HR, finance, facilities, etc.) and to requests that span multiple functions, such as employee onboarding. The service customer has a single interface for requesting any type of service, regardless of its complexity, at any time and from any device, as well as visibility into the status of the request, without needing to understand the underlying back-end processes.
- With ERM, fulfillment processes are customer-centric rather than based on internal needs. They are designed from the customer's perspective rather than what's most convenient or logical for internal service providers. As a recent report from Interactive Intelligence Group puts it, "Many firms perform their business processes with no attempt to delight the customer. Firms have a strong tendency to build processes based on what their internal organization prefers."[3]

This often results in shared-service environments—in which multiple functional departments deliver business services—wherein each business area processes requests only through its own systems for managing business services delivery. This is the antithesis of ERM, which is designed to allow any kind of serviece request to be requested in an easy and intuitive manner from a single portal interface, yet allowing fulfillment organizations to continue utilizing their systems of record to meet their requirements.

Think small

One of the biggest differences between ERM and SRM is based on the misconception that the word “enterprise” in ERM implies a long and expensive implementation process. As anyone with experience in creating an IT service catalog knows, the effort can involve an extensive amount of planning, analysis, form building and testing before even a prototype can be presented to users for further rounds of testing and modification. Much of the effort is dictated by the structure and constraints of different SRM products, which often restrict users to a predefined look and feel. But as one Kinetic Data customer stated, “That’s not the way the world works anymore. Often by the time you’re done with the analysis, the business has changed.”

ERM is a framework, not a product with built-in requirements and limitations, for a new way to approach service request management. Rather than taking one “big bang” approach to SRM based on a structured service catalog approach, ERM enables you to take an incremental and evolutionary approach. It *does* however require agile tools that enable you to customize, modify, and respond to constantly changing user and business needs and remap processes without rewriting core management and control application code. But much of the technology infrastructure you need likely is already in place.

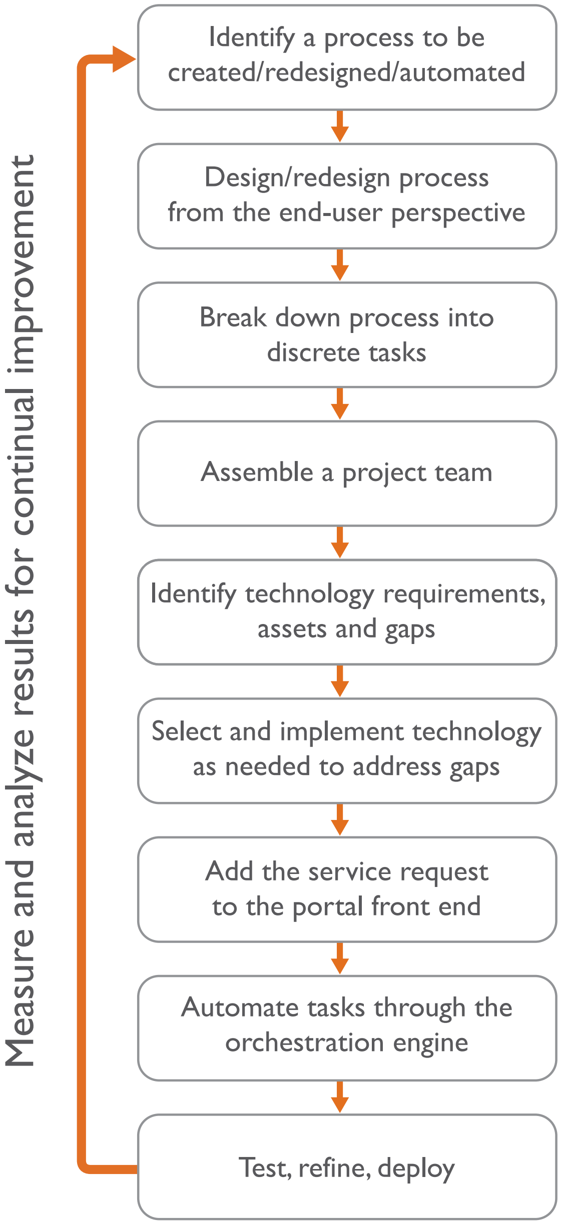
And though the approach is enterprise-wide, individual users are only able to view and interact with requests and processes that are relevant to them and those that they are entitled to view, based on their roles. For example, users in HR can be entitled to access the “increase pay rate” process, while partners may be entitled to request changes to product ship dates. Having an ERM solution that enables a flexible security strategy to control who is entitled to interact with which items is key—both to a favorable user experience and effective data security. This is essential to consider but, done incrementally and with the right tools, is quite manageable.

ERM enables “blue sky thinking”—creative ideas that are not limited by current thinking or beliefs. Nor is it limited by your current technology infrastructure.

Start with questions like: Which business process problems are most common? Which cause the greatest pain? What problem(s) is the business most passionate about fixing? Which initiatives promise the biggest payback with the least of amount of effort?

Those are the starting points for ERM. Unlike with SRM, there is no need to spend months planning every detail. And it doesn't matter where your organization is in terms of service request management maturity level. An ERM project needn't be big and scary. Rather, it requires identifying one or a few request management processes that are especially painful to users and the business itself, then assembling a small, informal project team consisting of a business analyst, a developer, the "owner" of the process, a representative from management, and, most importantly, users themselves, who can articulate the desired outcome in their own terms.

The ERM Implementation Process


Identify a process to be created/redesigned/automated
Design/redesign process from the end-user perspective
Break down process into discrete tasks
Assemble a project team
Identify technology requirements, assets and gaps
Select and implement technology as needed to address gaps
Add the service request to the portal front end
Automate tasks through the orchestration engine
Test, refine, deploy
Measure and analyze results for continual improvement

Next, break the process down into component tasks. What's the thorniest part of the problem? What's easiest to fix first? What will have the greatest impact on user satisfaction? Tackle these problems one at a time. Don't get mired in the current state of your technology or existing process, which is often a recipe for inaction. Often you'll find that if you "think small" by breaking processes down into realistically achievable goals and by building on the momentum these small victories generate, your current technology may not be as inadequate as you first thought. Any organizational change can prompt resistance. But such issues can be minimized by maintaining an open dialogue with business users—particularly once you begin to simplify and fix complex or broken processes.

One approach is to include a message area in your current online service catalogue, informing service requesters that you are making changes in order to improve a particular service request process, and invite feedback. Sample text: “Hi <user>, We know this is a complex process. We are working right now to make it easier/better/simpler. We’ll keep you updated on these activities right here and welcome your feedback.” If you continually engage with users in such a manner, they’ll understand that the changes are in their best interests and will appreciate the efforts.

Then, have the team tackle process problems one at a time. Try out the fix/adjustment/change in the development system—did it work as the team planned? If so, push that change out, update the user message, and continue with the next issue. Incremental changes demonstrate dedication by the support teams and concern for end users. Your team doesn’t have to solve every problem in a process before starting to make changes.

Narrow and shallow to wide and deep

IT service desks are often overburdened with calls. Look at the metrics you are (hopefully) already collecting and categorize the calls you are receiving. Are most of them for password resets? Or are users having problems with software installs?

Use your online self service portal to allow users to log and track common service requests themselves. That’s an example of a narrow and shallow ERM process—it focuses on one specific need in one specific area and a relatively easy way to resolve it. With a service request portal and a back-end process automation engine, ERM allows you to meet such simple needs easily. And with a flexible and extensible design, ERM allows you to add more and different types of requests over time.

ERM Technology Core Components

- Request management (Web portal front end)
- Orchestration engine / task workflow automation
- Departmental enterprise applications (ERP, HR, ITSM, etc.)
- SLA monitoring
- Reporting and analytics
- Costing and billing
- User feedback

A key concept of ERM is using process workflow automation software to orchestrate back-end systems of records in many different departments or functions. This makes it easy to evolve from narrow and shallow to wide and deep processes, automating complex requests that may span multiple parts of the enterprise, using lessons learned from earlier and simpler types of request fulfillment.

New employee onboarding is a common example. Before one large enterprise implemented ERM, the company had a partially automated onboarding process that started with a cumbersome 60-question online form. Completing that form resulted in an email to the service desk, which then had to manually send out tickets to multiple departments. Creating the tickets alone took roughly four hours per request. Since the company was hiring more than 200 new employees each month, the time and cost were enormous.

Drawing on lessons learned from successfully automating simpler self-service requests, the company was able to use a self-service portal to orchestrate and automate most of the ticketing activity and reduce the time to process each request from four hours to just minutes. Most importantly, the process was designed from the new employee's perspective. Employees started work with everything they needed from day one, including office space, computers and phones, system access and passwords, personalized training plans, security badges, and parking spaces. This was an example of a request that the company was passionate about—it wanted to reduce the stress of those first few days on the job as much as possible—and that previously entailed a great deal of pain since new employees sometimes had to wait days before they were even minimally provisioned to do their jobs.

It's also an example of a great ERM strategy: Start small, demonstrate value with early wins, generate organizational support, and then move on to the next pain point until eventually your ERM implementation becomes wide and deep.

How to start

- Every business has request management processes that employees would love to improve. Identify and prioritize processes to improve in terms of what is both realistically achievable and has the greatest impact on user satisfaction.
- Break down the process into discrete tasks. What is easiest to improve in the shortest amount of time? Start there before proceeding to tackle the more vexing problems.
- Assemble your (informal) project team. Keep it small. Larger teams are more bureaucratic and take longer to get things done. Include users on your team early in the process, and design improvements based on their input and perspective.

- Identify gaps in your service delivery technology. Can you create an extensible self-service request management portal? Can you automate request fulfillment processes and interact with needed back-end systems (i.e. ITSM, ERP, HR and others)? Often new front-end "systems of engagement" and flexible process automation tools may be needed, but make sure they are designed to interact with back-end systems of record with little or no modification.
- Take an evolutionary approach. Start with the "low-hanging fruit," or broken processes that have the greatest impact on customers. Working from these successes and the experienced gained, expand efforts wider and deeper into other request fulfillment processes.

Start with the first step

Unlike departmental or functional request management systems, ERM combines a universal front-end portal for any type of request with back-end business process automation of key tasks in the service delivery process. The four main benefits of ERM are:

1. An improved user experience;
2. Centralization of business services;
3. First-time and automated fulfillment; and
4. Leverage of existing systems.

The biggest difference between ERM and SRM is that ERM allows "blue sky" or more creative thinking about delivering seamless request fulfillment services to users in all parts of the enterprise, using one common toolset that interconnects and orchestrates back-end systems of record. Rather than taking a heavily structured and predefined approach to managing service requests, ERM encourages businesses to incrementally automate service request fulfillment and expand as the concept proves its value to the enterprise. From narrow and shallow to wide and deep, ERM is the "glue" that unifies service request fulfillment across the enterprise, regardless of your organization's level of request management maturity.

About Kinetic Data

Kinetic Data has helped hundreds of Fortune 500 and government customers—including General Mills, Avon, Intel, 3M and the U.S. Department of Transportation—implement Enterprise Request Management (ERM), an integrated system with a centralized request portal for automating request management. Kinetic Data was named “Innovator of the Year” in 2009 by enterprise software users and it also has been recognized with awards for its superior customer service and support. The company serves customers from its headquarters in St. Paul, Minn., offices in Sydney, Australia, and through a network of reseller partners.

For more information, visit www.kineticdata.com.

[1] Master The Service Catalog Solution Landscape In 2013, Eveline Oehrlich and Courtney Bartlett, Forrester Research, June 12, 2013

[2] See” Enterprise Request Management: An Overview” at

http://www.kineticdata.com/products/request/whitepapers/KineticRequest-ERM-WhitePaper.html

[3] “Humanizing Business Process Automation: Optimizing Performance for Employees and Customers,” Jesse Clark and Rachel Wentink, Interactive Intelligence Group, January 2012.