

# Identity Management Project Roadmap



This document will guide you through the entire life of a successful Identity Management project, including:

- A needs analysis.
- Who to involve in the project.
- How to select the best product.
- Technical design decisions.
- How to effectively roll out the system.
- How to monitor and assure sound ROI.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Needs analysis</b>	<b>2</b>
2.1	User productivity . . . . .	2
2.2	Excessive administration cost . . . . .	2
2.3	Inconsistent user data . . . . .	3
2.4	User service . . . . .	3
2.5	Security vulnerabilities . . . . .	4
2.6	Audit and reporting . . . . .	5
2.7	Regulatory compliance . . . . .	6
<b>3</b>	<b>Technology requirements</b>	<b>7</b>
3.1	System functions . . . . .	7
3.2	Technical infrastructure . . . . .	7
<b>4</b>	<b>Organization</b>	<b>9</b>
4.1	Mandate . . . . .	9
4.2	Budget . . . . .	9
4.3	Participants . . . . .	9
4.4	Ownership . . . . .	10
<b>5</b>	<b>Selecting a product</b>	<b>11</b>
5.1	Technical requirements . . . . .	11
5.1.1	Functionality . . . . .	11

5.1.2	Target systems . . . . .	12
5.1.3	Integration . . . . .	13
5.1.4	Deployment . . . . .	13
5.1.5	Flexibility . . . . .	14
5.1.6	Security . . . . .	14
5.2	Vendor profile . . . . .	15
5.2.1	Financial stability . . . . .	15
5.2.2	Quality of support . . . . .	15
5.2.3	Deployment time . . . . .	16
5.2.4	Single source . . . . .	16
5.2.5	Future direction . . . . .	16
5.2.6	Partners . . . . .	16
<b>6</b>	<b>Project management</b>	<b>17</b>
6.1	Project startup . . . . .	17
6.2	Product selection . . . . .	17
6.3	Acquisition . . . . .	18
6.4	Product design, pilot and deployment . . . . .	18
6.4.1	Design . . . . .	18
6.4.2	Installation . . . . .	19
6.4.3	Software configuration . . . . .	19
6.4.4	Training . . . . .	20
6.4.5	User roll-out . . . . .	20
<b>7</b>	<b>Ongoing system ownership and administration</b>	<b>21</b>
7.1	Ongoing support and upgrades . . . . .	21
7.2	Directory cleanups . . . . .	21
7.3	Reporting and auditing . . . . .	21
7.4	Functional and scope extensions . . . . .	22
<b>8</b>	<b>Summary</b>	<b>23</b>

## 1 Introduction

As today's organizations deploy an ever-growing number of complex systems and manage existing or new staff, manual administration of user access to systems becomes costly and ineffective:

- Requesting, routing, approving and acting on requests for new access, in particular for new staff, takes too long. New employees and contractors are unproductive as they wait.
- Too many administrators are tied up in routine user management chores.
- Access is not terminated promptly or reliably when people leave the organization, creating serious security vulnerabilities.
- It is difficult or impossible to say who has access to what systems and data, let alone who had access in the past.

Clearly, these problems call for automation, to consolidate and rationalize the administration of user identity data across a variety of systems.

This document will guide you through the entire life of a successful Identity Management project, including:

- A needs analysis.
- Who to involve in the project.
- How to select the best product.
- Technical design decisions.
- How to effectively roll out the system.
- How to monitor and assure sound ROI.

## 2 Needs analysis

The first step in an identity management project is to document the needs of the organization.

The needs analysis should identify the problems associated with existing business processes. The resulting set of requirements should be mapped to technical specifications, to be fed into subsequent technology selection and implementation design.

Following are the most common user access administration problems, and a brief description of available technology that can address them:

### 2.1 User productivity

#### The problem

Newly hired employees usually are not productive for an extended period of time, often weeks, as they wait for their accounts to be provisioned.

Existing users are not productive as they wait for updates or changes to their existing accounts.

#### Business process solutions

Streamline the “on-boarding” process whereby systems access for new employees and contractors is requested, routed, approved and provided.

#### Technological solutions

Implement:

- Automation to leverage data that is already entered into an HR system, or some other system of record, to automatically provision new users with at least basic systems access.
- A workflow system to accept change requests from users directly, request and track authorization, and provision access once requests are completed and approved.

### 2.2 Excessive administration cost

#### The problem

Users require access to multiple systems, such as the network, e-mail, Intranet, ERP, mainframe and more. Each system’s administrator must perform essentially the same tasks to create, change and terminate user access. This redundant effort is expensive and causes inconsistencies.

### **Business process solutions**

Consolidate routine user management to at least eliminate the possibility that several different administrators will have to work on a single request for a single user.

### **Technological solutions**

- Remove routine administration from security administrators entirely, and transfer it to automation and a workflow system.
- Use a consolidated user management facility to “front-end” a variety of systems, allowing a security administrator to manage one user across a variety of systems with a single program / GUI.

## **2.3 Inconsistent user data**

### **The problem**

Different systems may have different and possibly contradictory data about the same user. This makes it difficult to reliably provide services (e.g., sending e-mail to an external address), and to make security decisions (e.g., should the user be allowed to initiate a financial transaction?).

### **Business process solutions**

- Manage the administration of multiple systems in concert, so that updates are more likely to be made to every system where a given user has a record.
- Maintain a master record to track to which systems each user has access, and on which systems each user has a record.

### **Technological solutions**

- Correlate login IDs on different systems into a master directory that connects user records across the enterprise.
- Use a periodic process to extract user data from each system, search for discrepancies, and respond either by applying corrections directly or by requesting authorization for change requests.

## **2.4 User service**

### **The problem**

- Users may have different login IDs on different systems, which are hard to remember.

- Users have too many passwords, spread across multiple systems. Each password has a different expiry date and must satisfy different composition rules. Users forget or write down their passwords.

### **Business process solutions**

- Assign consistent login IDs on each system.
- Actively rename login IDs on non-compliant systems, to reduce the problem.
- Formulate a global password policy, and encourage users to maintain a single password to all systems, in compliance with this one policy.

### **Technological solutions**

- Reconcile login IDs across systems, and batch-rename IDs so that users have the same ID everywhere.
- Implement and enforce a uniform, global password policy.
- Provide password synchronization, to eliminate multiple hard-to-remember passwords.

## **2.5 Security vulnerabilities**

### **The problem**

- Delays in terminating system access when people leave an organization creates security vulnerabilities.
- Ex-staff may abuse access.
- Intruders may attack unused accounts without raising alarms.
- New accounts may be set up with trivial initial passwords, or initial passwords may be communicated to new users via insecure channels (e.g., plaintext e-mail). This opens a window of vulnerability after new accounts are set up, and before their owners change passwords.
- Users tend to accumulate access privileges as their responsibilities change and evolve over time. Many users wind up with more privileges than they require.
- Incorrect account configuration and evolving standards mean that many users are initially given inappropriate privileges.
- Users may be granted access to systems without appropriate authorization.

### Business process solutions

- Clearly document all systems access in a global repository, to make it easier to terminate access in the future.
- Ensure that HR, contractor management and others clearly and promptly communicate to IT whenever staff are terminated.
- Assign random passwords to new accounts, and find a secure method to communicate them either directly to new staff, or to their direct managers.
- Periodically review each user's systems access. Involve management in deciding whether access is appropriate given current responsibilities.
- Periodically audit account configuration on each system, to measure compliance with standards.
- Review access request / authorization / provisioning processes, to ensure that they meet business needs, and that they are adhered to. Note that violated policies are normally symptomatic of processes that are too slow to meet business needs.

### Technological solutions

- Tie systems of record, such as HR, to automation, which automatically terminates systems access.
- Initialize new passwords either to secret personal data, drawn from a system of record, or to values entered by the manager of the new person.
- Periodically run reports about user access privileges, and provide them to managers, for review.
- Create all new accounts in compliance with standards – for example by cloning model accounts.
- Provide a user-friendly, fast and reliable change authorization workflow system.

## 2.6 Audit and reporting

### The problem

For reasons of good governance, and in many cases for regulatory compliance, it is important to be able to track what users have access to what systems and data.

In a heterogeneous environment, this information may be scattered across multiple systems, uncorrelated, or simply unavailable.

### Business process solutions

- A master record should be built to record which systems each user has access to, and on which systems each user has a record.
- Periodic reports from each system can capture what local privileges users have.
- Local access reports can be reconciled, though this may be a costly exercise.

## Technological solutions

- Reconcile login IDs across systems, and batch-rename IDs so that users have the same ID everywhere.
- Automatically extract access rights (attributes and group memberships) from target systems, and report on them centrally.
- Record the times when access changes were requested, approved and implemented, and report on this historical data as well.

## 2.7 Regulatory compliance

### The problem

Regulations governing customer privacy (e.g., in the health care, financial and retail industries), sound business processes (e.g., in the pharmaceutical industry) and good corporate governance (e.g., for publically traded companies) generally require basic security practices:

- The ability to securely authenticate users.
- The ability to control user access to sensitive systems.
- The ability to ensure that user access to systems and data is only granted when appropriate.
- The ability to measure and prove all of the above.

These requirements can be restated in terms of the consistency, security, audit and reporting criteria laid out above.

## 3 Technology requirements

### 3.1 System functions

The business needs analysis above produces the functional requirements for an identity management system listed below. In any given organization, some subset of these requirements will be relevant:

- **Automation** to leverage data that is already entered into an HR system, or some other system of record, to automatically propagate changes to target systems.
- **Workflow** to accept change requests from users, request and track authorization, and update accounts and access rights on target systems.
- **Consolidated user management** to “front-end” a variety of systems, allowing a security administrator to manage one user across a variety of systems with a single program / GUI.
- **Delegated user management** to allow local managers and IT resources to manage just some users, on just some systems, using the consolidated user management facility.
- **An auto-discovery process**, to periodically extract user data from each system, search for discrepancies, and respond either by applying corrections directly or by requesting authorization for change requests.
- **A login ID reconciliation system** to connect user records across systems.
- **Password management** that spans every system, including a strong, uniform password policy engine, password synchronization and self-service password reset.
- A secure process for **password initialization**.
- **Audit logs** that track all access change requests, authorizations and implementation.
- **A reporting engine** that can provide reports about current and historical user access privileges.
- **Standards enforcement** for change authorization, and new account configuration.

### 3.2 Technical infrastructure

An identity management system should have some core technical capabilities, including:

- **Scalability**

The identity management system should support the number of users and systems under consideration and should scale to support future growth. Run-time performance and load placed on the network and on target systems should be reasonable.

- **Flexibility**

The identity management system should be able to implement organization-specific requirements, such as how to assign new login IDs, how to identify suitable authorizers for change requests, how to map changes in a system of record to target systems, etc.

Flexibility also relates to target systems: it should be easy to manage users on new, non-standard systems, such as custom and vertical-market applications.

Finally, flexibility may impact the exact meaning of updates to target systems. For example, disabling a login account on a target system may mean setting the “account disabled” flag in one organization, while it may mean moving the account to a different container in another organization.

- **Security**

The identity management system must be secure, not only in the sense that it enhances the security of existing users and target systems, but also in the sense that it is well protected, and does not introduce new security vulnerabilities.

In practice, this means that it should be locked down, and implement internal access controls and encryptions to limit the risk of unauthorized activity.

- **Rapid deployment and minimal ongoing maintenance**

An identity management system is of no value until it is operational, and of limited value until it is fully in use. Moreover, many identity management systems offered in the past have been exceedingly difficult to implement, and deployments have aborted more often than completed.

Technology that assists in rapid deployment is essential to a successful identity management project. This includes at least:

- A rich set of pre-built agents, to connect to target systems.
- A toolkit to facilitate easy integration with new types of target systems.
- A robust auto-discovery engine, to periodically extract user profile data from target systems.
- Limited if any installation of agents locally on target systems.
- Easy configuration of target systems, roles, policies.
- Simple definition and maintenance of automation rules.
- Simple configuration and maintenance of change authorization workflow.

## 4 Organization

To be successful, an identity management project must have a mandate, a schedule and a budget. Persons in an organization with a vested interest in identity management must be involved early in the project. This ensures that their requirements are met at the design stage, and that they will not object to any part of the project during deployment.

### 4.1 Mandate

A account management project must start with a clear mandate to solve specific business problems. [Section 2 on Page 2](#) outlines the most likely issues to be resolved.

Projects that start without this mandate may fail when the time comes to request resources and the support of groups within the organization.

### 4.2 Budget

It is often helpful to verify, at the onset of an identity management project, whether or when sufficient funds will be available. The following items require funding:

- A software license for the selected product.
- Annual support costs.
- Training.
- Hardware and associated software costs (including operating systems, network management software, installation).
- Professional services – to install the selected product and to manage a roll-out.
- Internal resources – for project management, product selection, installation and ongoing system administration and support.

### 4.3 Participants

Early involvement by all interested parties in an organization ensures that the final design reflects all needs, and that no objections will be raised late in the project.

The following groups are typically involved in an identity management project:

- Security administration:  
Must understand how to use the system and its impact on their work.

- Enterprise security and/or audit:  
Must define security policy and audit requirements that the system will enforce. Will likely use the system to monitor policy compliance, access rights and change history.
- Systems administrators:  
Must understand the impact of an identity management solution on the systems they manage.
- IT security:  
Must understand the impact on overall security policy and design.
- Human resources:  
Must agree to provide authoritative input information to the system, at least for employees, and ideally for contractors as well.

#### 4.4 Ownership

It is crucial for an identity management project to include the system's long-term owner, as early as possible.

Ideally, the long-term system owner and the system's technical administrator(s) will have a strong influence over product selection. These people will have to work with the system and its vendor, so they are more likely to take the time to make a critical analysis of product documentation, and undertake a technical laboratory evaluation of candidate products.

It is risky, on the other hand, to have one team select a product, and a separate team install and manage it.

## 5 Selecting a product

The ideal identity management product should meet all of the project's technical requirements, and be supported by a stable, mature and helpful vendor.

The following sections describe the technical and business requirements that an identity management system vendor should meet.

### 5.1 Technical requirements

#### 5.1.1 Functionality

Functional requirements for an identity management system are summarized in [Subsection 3.1](#) on Page 7.

Additional detail for each function is presented below:

Capability	Requirements
<b>Automation</b>	Support multiple systems of record; be able to filter out just changes of interest; be able to transform changes to multiple formats, suitable for multiple target systems.
<b>Workflow</b>	Allow any business users to submit change requests; be able to validate input requests; assign authorizers based on requested resources, on the identity of the requester, and on other request attributes; ask authorizers to approve requests by e-mail; support reminders, delegation and escalation to ensure that authorizers respond in a timely manner; allow requesters to track the status of their request, and to cancel open requests if appropriate; record change history for later reporting.
<b>Consolidated user administration</b>	Allow security administrators to manage any user on any system from one point; implement access control lists to control what each administrator can see and do; be fully accessible from a web browser.
<b>Delegated user administration</b>	Allow organization-specific logic to leverage existing information to make dynamic delegation decisions (i.e., can this administrator sign in, and if so what can he or she do, to whom?).
<b>Auto-discovery</b>	Be able to efficiently extract full lists of users and groups from each managed system, nightly; be able to extract arbitrary user attributes and group memberships from managed systems, without impacting run-time performance too much.
<b>Login ID reconciliation</b>	Support both systems where login IDs are standardized/consistent, and those where they aren't. Allow users to attach non-standard, uncorrelated IDs to their own profiles in a secure manner.

Capability	Requirements
<b>Password management</b>	Include password synchronization, self-service password reset, assisted password reset, global policy enforcement, early warning of upcoming password expiration, etc.; be accessible from a web browser, workstation login prompts and a telephone (IVR).
<b>Password initialization</b>	Allow users entering change request to specify initial passwords, to eliminate insecure default values and transmission.
<b>Audit logs</b>	Track user access rights on managed systems, and full change history.
<b>Reporting engine</b>	Support both built-in reports (e.g., who has what), and an open schema and reporting interface, to allow for custom reports.
<b>Standards enforcement</b>	Ensure that change requests are properly authorized. Create new accounts in compliance with standards, for example by cloning model IDs.

### 5.1.2 Target systems

A successful identity management system should be able to manage login IDs on most or all of the systems to which users login.

Target systems should work “out of the box” in as many cases as possible.

Where this is infeasible (e.g., home-grown applications, vertical market applications, legacy applications), the product should be open enough to make it possible to easily integrate with applications:

- Some applications include an API for managing passwords. While rare, this is a useful mechanism to integrate an identity management system. It’s useful to check the language bindings of any such API, and compare these to what the account provisioning system supports.
- Some applications include command-line tools that allow account operations such as creation, deletion and update. The account provisioning system should be able to execute these – on whatever platform they are available.
- Some applications store their account information in a database, where an identity management system may manipulate it directly. This includes client/server applications and web applications with DBMS back-ends.
- Some applications run on midrange or mainframe systems, and can be manipulated by scripting interaction with a terminal login session.
- Some applications present a web GUI, and an account management system can interact with them by simulating the actions of a web browser.

### 5.1.3 Integration

An identity management system should integrate seamlessly with existing IT infrastructure, including:

- **Authentication systems:**

Users should be able to authenticate using existing infrastructure – be it a network login ID/password (such as a Windows NT domain), security tokens (such as an RSA SecurID) or another technology.

- **Support systems:**

The system should automatically create issues / tickets in any help desk call tracking system, to allow for follow-up in the event of a problem (e.g., target system outage or unresponsive authorizer).

- **Electronic mail:**

The system should be able to interact with users, administrators and authorizers by e-mail – for example, to notify requesters that the request for new account has been processed and the appropriate accounts have been provisioned.

- **System monitoring:**

Existing infrastructure should be able to monitor the identity management server's health, and react to alarm conditions.

### 5.1.4 Deployment

Deployment should be as simple as possible. Features supporting this objective include:

- **Auto-discovery**

The system should be able to automatically and periodically extract user profile information from target systems. This should not be a one-time or manual process.

- **Login ID reconciliation**

Login IDs on different systems should be automatically correlated to one another where possible (i.e., where they are consistent, or where correlating data exists and is both complete and reliable). In other cases, self-service reconciliation should be supported, to avoid a massive central reconciliation effort.

- **Minimize installation of agents locally on target systems:**

Installing agents on a production server normally involves a lengthy change control process. Using existing client software to communicate with servers reduces deployment time.

- **Integrate with existing databases and directories:**

An identity management system should take advantage of existing user profile databases, which may include information on various accounts for a given user.

An identity management system should be able to use the existing database as a source of information for additional account provisioning.

- **Simple, web-based configuration**

The system should be configured and managed with a simple, web-based interface.

### 5.1.5 Flexibility

The system should cope with both current and possible future requirements for:

- **User interface:**

The user interface should be customizable, and support different appearances for different users (such as multiple languages or user groups).

- **Login ID assignment:**

The mechanism for assigning login IDs to new users should be externalized, to allow organizations to implement their own logic.

- **Workflow:**

The workflow system should leverage existing data to identify authorizers, and to find suitable people to receive escalated change requests.

The workflow system should be able to validate and fill-in attributes in change requests.

The workflow system should be able to limit what users can request, in order to simplify the change request input process.

- **Automation logic:**

The business logic for monitoring changes to user profile information in systems of record (e.g., HR), and for filtering, transforming and pushing out changes to target systems should be highly configurable.

- **Help desk integration:**

The system should be able to respond to a wide variety of conditions by asking for assistance, in the form of writing a call tracking ticket in a help desk automation system.

- **Password policy:**

The system must be able to enforce a global password policy for newly provisioned accounts.

### 5.1.6 Security

An identity management system literally owns the “keys to the kingdom” and consequently must meet the most stringent security requirements:

- **Encryption**

- User access to the system must be encrypted (this typically means only HTTPS should be supported as a user interface).
- Any sensitive data stored in the system should likewise be encrypted or hashed, as appropriate. This includes administrative passwords of people authorized to manage the system, as well as passwords used by the system to manage target systems. This also includes any sensitive user profile data.
- The system should support encrypted communication with all target systems – including those that do not natively implement an encrypted client/server protocol (e.g., most DBMS servers, mainframes, etc.).

- Encryption should rely on well-known implementations of well-known, trusted encryption and hashing algorithms.
- Encryption keys should be managed effectively. For example, public keys must be signed by a real certificate authority (and not by the vendor). Private keys must be obscured and protected by operating system ACLs.
- The amount and type of information available in e-mail or any other non-secure notification should be configurable (e.g. the ability to exclude any sensitive and/or private information that was used during the provisioning process).

- **Authentication**

- The system should allow access to authenticated users only.
- Any authentication failures should be tracked. The system should be able to lock users if the authentication failure threshold has been reached.

- **Accountability**

The system must record every possible event, so that users and administrators alike can be held accountable for their actions.

- **Hardened platform**

- The system should operate on a locked down operating system.
- The system should support a diversity of web servers, so that if a given web server is deemed to have an unacceptable history of vulnerabilities, it can be avoided.
- The system should be accessible across web proxies, so that it can be installed in a protected subnet, and accessed across a firewall without opening non-HTTPS ports.
- The system should not require the installation of (possibly insecure or vulnerable) client software.

## 5.2 Vendor profile

As with any vendor, the company supporting an identity management system should offer sound support, effective professional services, good relationships with other relevant vendors, and long term stability.

### 5.2.1 Financial stability

In the interests of long term support for the technology, it is important to verify that prospective vendors are financially sound: growing rather than shrinking, and profitable rather than burning cash reserves.

### 5.2.2 Quality of support

Quality technical support is crucial to project success. This is best measured by implementing the identity management system in a test environment, and evaluating the ability of the vendor to assist in the installation process.

### 5.2.3 Deployment time

Vendors should be able to offer turn-key or assisted deployments. A good vendor will be able to successfully deploy the system in a minimum amount of time. A good product can be deployed without intrusion – without installing desktop software, and with limited use of server agents.

The deployment effort in a large organization should not take more than 10-20 supplier person/days.

### 5.2.4 Single source

It is easier and safer to work with a vendor that can provide all the required technology directly. This eliminates the risks of using third party technology, such as:

- Increased cost.
- Uncertain future product availability and revision.
- Limited, poor or inconsistent technical support.

### 5.2.5 Future direction

The successful vendor should have a clear direction for future growth and technology advancement. This helps to ensure vendor stability, and a sound future for the product.

### 5.2.6 Partners

identity management products must inter-operate with other IT infrastructure supported by current suppliers. Relationships between an identity management vendor and the vendors of other infrastructure or services can streamline interoperability and ongoing support.

In particular, it is helpful if the identity management vendor has a working relationship with providers of:

- ERP and HR technology providers.
- IT and help desk outsourcers.
- Security infrastructure.

## 6 Project management

The following sections outline the objectives of each phase in a identity management deployment project.

### 6.1 Project startup

To begin the project:

1. Perform a needs analysis, as described in [Section 2 on Page 2](#).
2. Document technical and business requirements, as described in [Section 5 on Page 11](#).
3. Establish a project whose mandate is to resolve the problems identified in the needs analysis.
4. Identify prospective vendors and products.
5. Allocate and approve people, systems and a budget.

### 6.2 Product selection

To make an effective product selection:

1. Perform some research to find out what products are currently available. Analyst firms generally know which vendors have significant market share, and can identify prospects.

Another excellent source of information is the Internet: use a search engine to find sites that mention:

- “identity management,”
- “user provisioning”
- “user access management.”

2. Once you have identified prospective vendors, forward your technical and business requirements document to them, and request a proposal.
3. Provide the prospective vendors a list of key decision-makers in your organization and their selection criteria. This will help vendors to focus their efforts on what matters most to you.
4. Evaluate the product in either a laboratory environment or with a pilot group of users and systems. Evaluating products based on paper only is very risky. You may reach final conclusions based on unfounded or inaccurate information.

Vendor RFP responses are no substitute for lab testing: some vendors will respond to RFPs based on what they believe the customer wants to hear, with no bearing on what their product can actually do, on the theory that “we can either build it later, or convince the customer that they don’t need it.”

Analyst reports are also no substitute for lab testing: the analysts do not install products in their own labs, and instead rely on every vendor for an assessment of their own capabilities. Specific vendor claims are not verified.

Ensure that all features defined in the requirements document are tested and compared. This exercise will highlight differences between products and vendors in a way that a paper process cannot.

5. Compare vendor proposals, technical evaluation results and prices.

## 6.3 Acquisition

Once a product has been selected, negotiate on a price and project deliverables and sign a contract. Fixing the price and deliverables (professional services, milestones, level of support) mitigates project risk.

Include a detailed list of deliverables and a statement of work attached to the contract.

## 6.4 Product design, pilot and deployment

Prepare a detailed deployment plan including: system design, schedule and resource allocation. These should cover the following aspects:

### 6.4.1 Design

Determine:

- Which features will be activated.
- How users will access the system.
- Which security policies (including password policies, authentication method, request/attribute validation, account configuration standards, authorizer selection logic) will be enforced.
- What types of accounts can be provisioned on each target system? Is there a model account that can be cloned? Who can authorize creation of each account type?
- How user privileges are grouped into roles, and who can authorize creation of new users to match roles?
- Whether the system will integrate with a system of record, to implement some form of automated administration, and if so what user objects are of interest and how they map to user objects on target systems.
- How login IDs will be assigned.
- Whether users will be able to submit change requests, and if so:
  - What can they request?
  - What ancillary data / attributes must each request include?
  - How are requests validated?
  - Who must authorize each request?
  - How often should authorizers be reminded to respond?
  - How should requests without authorization be escalated?
- Whether the system will integrate with the help desk issue tracking system, and if so how/when it will create open/closed tickets.
- Whether the system will integrate with e-mail, the events that will trigger e-mail, and the messages to be delivered.
- The number of servers needed.

## 6.4.2 Installation

Determine how you will carry out:

- Operating system and web server installation.
- Application software installation.
- Integration with existing infrastructure, including systems of record, e-mail, help desk systems, meta directories, etc.
- Multi-server replication, load balancing and health monitoring/response.

## 6.4.3 Software configuration

The sequence of feature and target system activation can vary widely from organization to organization. It is normally best to start small, and grow the system's capability over time, as this reduces project risk, and establishes value early rather than late in the project.

Following is a typical feature installation sequence:

- Install the base software.
- Configure an initial set of target systems.
- Implement user interface customization / localization.
- Implement e-mail, directory and meta directory integration.
- Verify that basic operations, such as creating and deleting users, work.
- Connect to a system of record, and configure basic automated administration (e.g., create network access for new users, terminate all access for staff who left).
- Implement a basic workflow – e.g., to create new users and accounts, and to manage membership in security groups on target systems.
- Pilot test with security administrators (consolidated user administration), requesters, authorizers, and HR-initiated automated updates.
- Move initial functionality to production.
- Add target systems.
- Add managed attributes and groups on target systems.
- Implement help desk call tracking system integration.
- Extend automation business logic.
- Extend workflow business logic.
- Pilot test new features with a broader pilot group.
- Move extended functionality to production.
- Repeat as necessary.

#### **6.4.4 Training**

Determine how you will:

- Make users aware that new change requests will be made through the system.
- Train HR staff to ensure that their input into their system must be timely, accurate, and must capture key information that will drive automation.
- Train security administrators to use the system to manage users globally, rather than continuing to use local administration tools.
- Train security officers and auditors to use the system to extract global access rights and access history reports.

#### **6.4.5 User roll-out**

Determine how you will notify users about the system and, if appropriate, ask for registration.

## 7 Ongoing system ownership and administration

While the bulk of the work in an identity management deployment process ends with user roll-out, more work is required on an on-going basis. This includes:

### 7.1 Ongoing support and upgrades

A technical resource must be assigned to ongoing system support. In particular, this person must:

- Monitor the system.
- Act as an advocate for the system, to encourage utilization.
- Answer user and help desk questions.
- Periodically add target systems.
- Troubleshoot any problems that may arise.
- Alter integration business logic as help desk, authentication, meta directory and e-mail systems are changed.
- Install software upgrades.
- Modify the product if business requirements for provisioning change or new ones get created.

A mature product should allow you to minimize the amount of effort required to perform these duties.

### 7.2 Directory cleanups

Once deployed, an identity management system can be used to identify and remove dormant and orphan accounts on each managed system.

Another type of directory cleanup made possible by an identity management system is login ID normalization, to eliminate the need for users to remember multiple, different login IDs on each system.

An identity management system can report on user access rights across systems, and these reports can be given to management to verify appropriateness, with the result being reduced system privileges where they are not needed.

### 7.3 Reporting and auditing

As mentioned earlier, security officers and auditors will most likely want to run reports using the system to track user access to systems.

## 7.4 Functional and scope extensions

Over time, the functionality and scope of a successfully deployed identity management system tends to grow. This includes:

- Adding target systems.
- Adding attributes and groups on target systems to the scope of management.
- Extending the business logic for automated provisioning.
- Delegating the ability to manage some users, on some systems, to local managers and departmental IT resources.
- Expanding the set of security requests that users may submit.

## 8 Summary

Identity management systems offer a simple way to improve the process of creation and maintenance of user objects across the IT infrastructure.

A successfully deployed identity management system improves user productivity and service, reduces security administration operating cost, and improves both security and accountability.

A typical project begins with a concept, proceeds through a needs analysis, produces a set of technical and functional requirements, includes a broad deployment team, and proceeds through product selection, system design, deployment and ongoing management.

If managed effectively, an identity management project can go from conception to production in under six months, and yield a positive return on investment within 2-3 months of deployment.