Vertical Sky Evolution Management Solution

Issue-based Change Management for eBusinesses

A Technical White Paper

	Worldwide Offices:	
> CORPORATE HEADQUARTERS Vertical Sky Incorporated 2500 S. Highland Ave. Lombard, IL 60148 TEL: 630 495 2108 FAX: 630 495 3591 SALES: 800 633 1235 WWW.VERTICALSKY.COM	> CANADA TEL: 519 884 2251 FAX: 519 884 8861 SALES: 800 265 2797 > GERMANY TEL: +49 711 351775 0 FAX: +49 711 351775 11	> UK TEL: +44 1483 733900 FAX: +44 1483 733901 SALES: +44 1483 733919 > BARBADOS TEL: 246 228 4217 FAX: 246 437 8223

Table of Contents

The Problem

Few business analysts or consultants will disagree that eBusiness is having an impact on the post-2000 business environment. With each day another story is published about a company that has developed an innovative way to solve a business problem or redefine an existing business process.

In most cases, the World Wide Web is an integral part of the new environment. Having successfully launched their initial site, these businesses are faced with the same challenge on their Web site that they face elsewhere in the business; change is constant.

Giga Information Group outlines some driving forces for eBusiness change management as:

- increased focus on quality
- continuous change of eBusiness applications
- distributed teams
- outsourcing of applications and use of packaged applications
- new types of components and staff

Customers, suppliers, partners, development teams and others all have legitimate business reasons for wanting changes or enhancements that impact the Web presence. Making sure that these changes are tracked, prioritized, implemented, and ultimately that the requestors are informed of the progress of their request is integral to the success of the eBusiness operation.

However, solutions that have worked in the past need to be revolutionized for managing eBusiness change. Not only is the pace of change quicker than it has been historically for most organizations, but there is a wide variability in the scope of the requested changes ranging from fixing a typo on a page in the Web site, to adding functionality to allow customers to initiate, review, and monitor business transactions from the site.

International Data Corporation notes "poor management of code and content can easily lead to business impacts such as revenue loss and drive customers to a competitor's site. The way to avoid problems is through good management and control over the entire life cycle of code and content development and deployment – the software assets of any eBusiness solution."[1]

Gartner Group, Inc. has stated "through 2005, one in five large, commercial Web sites will incur significant costs from legal action, service interruption or loss of business arising from poor controls and coordination of code and content management."[2]

In addition, with consolidation and telecommuting a constant reality in the established industries, development teams tend to be geographically dispersed, operate on different platforms, and have technical expertise in many diverse skills. All of this diversity needs to be focused on evolving the eBusiness.

[1] *Tools Required for Managing Electronic Assets*, International Data Corporation, 2000
[2] *Software Change Management*, Gartner Group, Inc., 2000

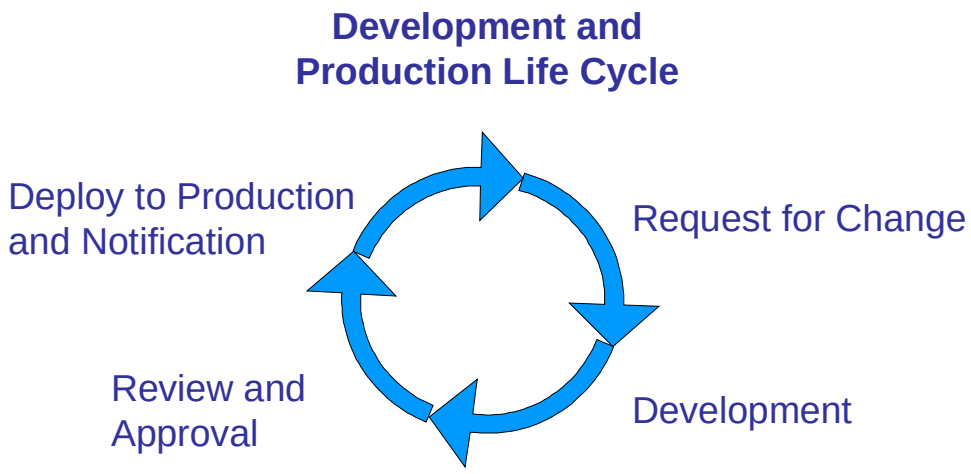

Figure 1: eBusiness Development & Production Life Cycle

In this paper Vertical Sky presents a discussion of the ways issue-based change management can help your eBusiness maintain control over the forces pulling it in different directions.

eBusiness Development and Production Life Cycle

Figure 1 illustrates the eBusiness Development Life Cycle. Issues are raised and enhancements requested. The requests are evaluated and a decision on how to proceed is made. If that decision is to act upon the issue, it is assigned. Once assigned, the request goes through the appropriate development life cycle (software or content). Once the development is completed the resulting work product is submitted for review and approval. From there it is deployed into production and the cycle begins anew.

In a world of constant and rapid change, the ability to define and enforce the processes and workflows required to implement all aspects of the eBusiness Development Life Cycle is at the core of the Vertical Sky Evolution Management solution. The Vertical Sky Evolution Management solution is based on the premise that while change is good, uncontrolled change can quickly and easily lead to chaos, and ultimately negatively impact business performance. This paper examines the different phases of the lifecycle and describes the processes that typically occur in those phases.

The Request for Change

The purpose of rapid evolution management systems is to assist organizations in reaching their evolving business goals. The *Request for Change* (RFC) can initiate anywhere, either from inside the organization's technical or line-of-business parts, or from outside the organization with customers, partners, vendors, or other stakeholders. For the purposes of this discussion, assume that the RFCs are an identified subset of all customer, partner, or stakeholder communications.

In the case of customer interfaces, a *Customer Relationship Management* (CRM) system may be the primary support system for managing the customer interface. Any RFC is evaluated using the process defined for that interface. That process usually involves informing the customer of the "Incident Number" the customer can use in all future communications with the company, to track the progress of the request.

As more of the company's business is transacted on the site or impacted by the site, many RFCs are assigned to the site team. The RFC from the CRM may be passed directly to the Web site *Change Control Board* (CCB), which evaluates all change requests and allocates resources based on priority, urgency, and importance of the change. The nature of the change can be as simple as a spelling mistake on one of the Web pages or as complex as a request to provide customers the ability to manipulate their portfolio through the Web interface. The change may require a simple HTML edit to one page, a whole new servlet, an updated interface, a change to legacy back-end systems, or anything in between. The decision on the disposition of the RFC by the CCB may need to be communicated back to the CRM, so the customer can be informed of the company's decision on how to proceed with the change.

The process for addressing partner or internal requests may not already be supported by automation tools, in which case the Vertical Sky Evolution Management solution may be exposed directly to all those stakeholders. In all cases, the progress of the RFC through the process (which may be different for external partners than for internal constituents) needs to be available to the requestor so they can monitor progress of the RFC.

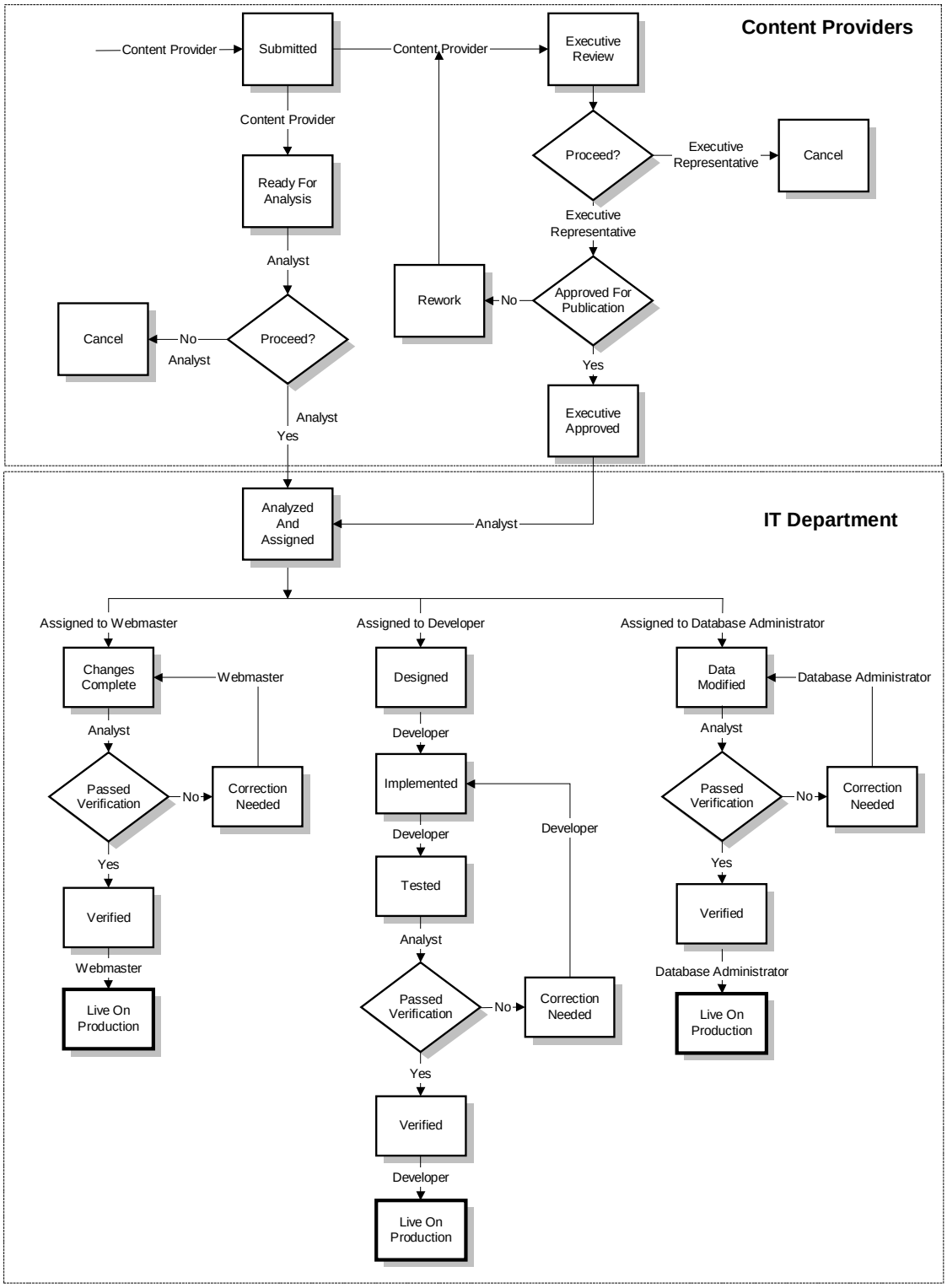

Figure 2: Example Process Flow for Change Requests Related to a Web Site

Assuming the decision is to proceed with the change, the RFC is assigned to development where it enters the next part of the life cycle.

Development

The development process can be defined as those steps or activities required to produce the artifacts that end up on the Web. Vertical Sky's focus is to provide staff involved in the development process with an environment responsive to their needs, and to help manage the different phases of the development process.

Depending on the nature of the requested change it may be assigned to a software development group, a content development group, or a line-of-business team member who has the expertise to deal with the issue (for example, answer a FAQ, update a job posting, post a recipe for yesterday's special). The development lifecycle differs wildly depending on the scope of the solution required to address the issue. It is important the tools that support the process are sufficiently flexible to help automate the different development processes.

Software Development

Software development typically is conducted by the software team using whatever development process is most appropriate for the size of the team, the nature of the software being developed, and the quality standards that need to be adhered to. The application development environment today is expected to execute faster, better, and more cost-effectively to respond to competitive business dynamics and guaranteed customer satisfaction. To best meet these expectations, software development teams need to have a change management system that integrates seamlessly into the development environment. Software developers want to be able to do their task in the development environment they are familiar with and do not want to have to learn a new tool to do their jobs. The software development life cycle tends to be longer than content development life cycles and the effort required to effect changes in the software is usually orders of magnitude larger than that required to effect content changes.

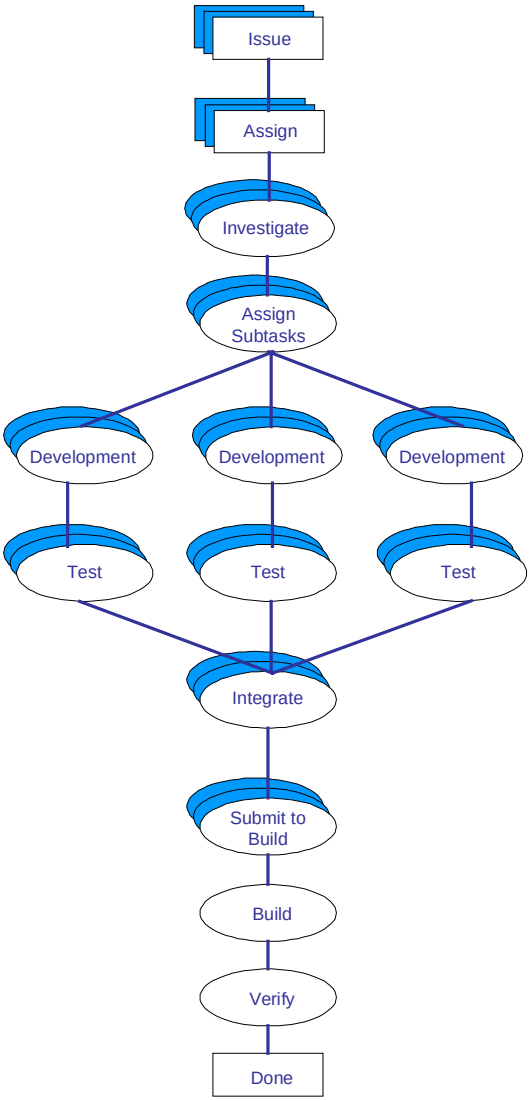

Figure 3: Software Process Flow

A typical software development process is presented in Figure 3.

In this simple process, issues are assigned to team leads who investigate the possible solutions. They allocate tasks to individual developers, possibly assigning different parts of the resolution to different people. Each developer does his or her part. Then all collaborating developers integrate the components and, once they are satisfied that the issue is resolved, they submit their changes to the build. In the build step, the changes to source code required to effect the different fixes that have been assigned to the build are applied to the code base and a built artifact is produced. At this point independent verification may occur prior to deeming the built artifacts suitable for submission to production.

In addition to the steps of transforming source files into deployable artifacts, this process is characterized by a level of collaboration and verification that is not usually found in the content development process.

Content Developer

By contrast, the content developer and the line-of-business experts tend to have a much simpler process flow, characterized by smaller teams and submission without extensive intermediate steps of aggregation and transformation.

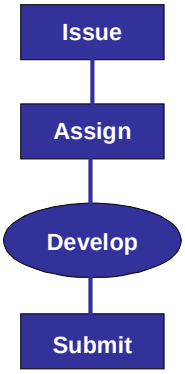

Figure 4: Content Development Process Flow

Content developers typically fall into two categories; casual contributors or high usage contributors. In both cases, participating in the change process needs to be sufficiently unobtrusive so that it does not discourage them from participating.

These content contributors could include:

- Graphic Designers
- Contributors / Creators of Audio and Video
- Members of internal departments such as marketing (collateral), Human Resources (career postings), legal staff
- Webmasters
- Page and template designers
- PR firms and advertising agencies
- Suppliers, vendors, partners

The tools of choice for the high usage content creators typically include: DreamWeaver, HomeSite, FrontPage2000.

Casual contributors who are not familiar with the above tools don't want to have to learn to use them to contribute. They need to be presented with an easy way to submit their contribution (for example, templates that effectively separate form from content) without needing to know anything about working in HTML, the site, Web technology, or system administration.

Review and Approval

The purpose of this step is to ensure that the face that is about to be presented to the outside world through the site meets the business goals set out for the site. Because the interpretation of those business goals cannot be left to individual developers, it makes sense that there be a simple way to assess requested changes to ensure that they meet the stated goal and the intended goal.

Inherent in Review and Approvals are the following notions:

- Notifications
- In Context Review
- Staging
- Promotion

The implementation that embodies all these notions provides an environment where change can be reviewed and approved with minimal delays, allowing the required changes to propagate through to the production site in the required time, while providing a high degree of confidence that the changes made to the site are correct.

Notifications

Knowing that one's input is required as part of the change management process is the catalyst to participation. This knowledge can be gained by either polling the task list that is generated by the change management system or through notifications. Users can select to be notified by email, pager, or any other device supported by the corporate messaging system.

Notifications must be available for any specified event in the workflow. These events include being assigned a task, having a fix submitted where a user's approval is required, having a submission rejected, having a deploy to a production server fail, and so forth. Any event or transition on the system can be configured to cause a notification.

In Context Review

Content is reviewed by different people for different purposes. One reviewer may need to approve the look of the page, while another may be concerned only with the marketing described in the submitted package. Different approvers need to review content in different stages of maturity. A peer review is usually performed as early as possible in the review process while QA review is usually performed as the last step of the review process. Each reviewer needs to review the content in the context that is useful to him or her. In the case of the peer review, that usually means performing the review in the context of other changes that are early in the review process. In the case of QA, the review needs to occur in a context that is the one proposed for final publication.

Each reviewer needs to be able to view the changes as they will appear in the final site. If the changes are made to HTML pages, the user needs to be able to follow all the links, as well as review the content of specific pages submitted. If the content is personalized data, the user may also want to access the system using a profile that will display the submitted content. Regardless of the type of content submitted, the reviewer must be able to view the submitted content as it will appear to the outside world once deployed.

Staging

Staging is a technique used in many business environments to segregate content of different phases of maturity. Content of dubious maturity is kept in a location, or stage, different than content that is known to be ready for release. On one hand, a developer may need to view changes in the context of other content in development. QA, on the other hand, typically wants to view changes in the final stage of approval with all the content that either has already been deployed into production or content that is awaiting final approval

Stages are physical entities (as opposed to logical entities), similar to a production site. All the content that is subject to approval on the production site also resides on each stage, though possibly at a more

recent version.[3] The figure below illustrates how different versions of the same content may appear in different stages.

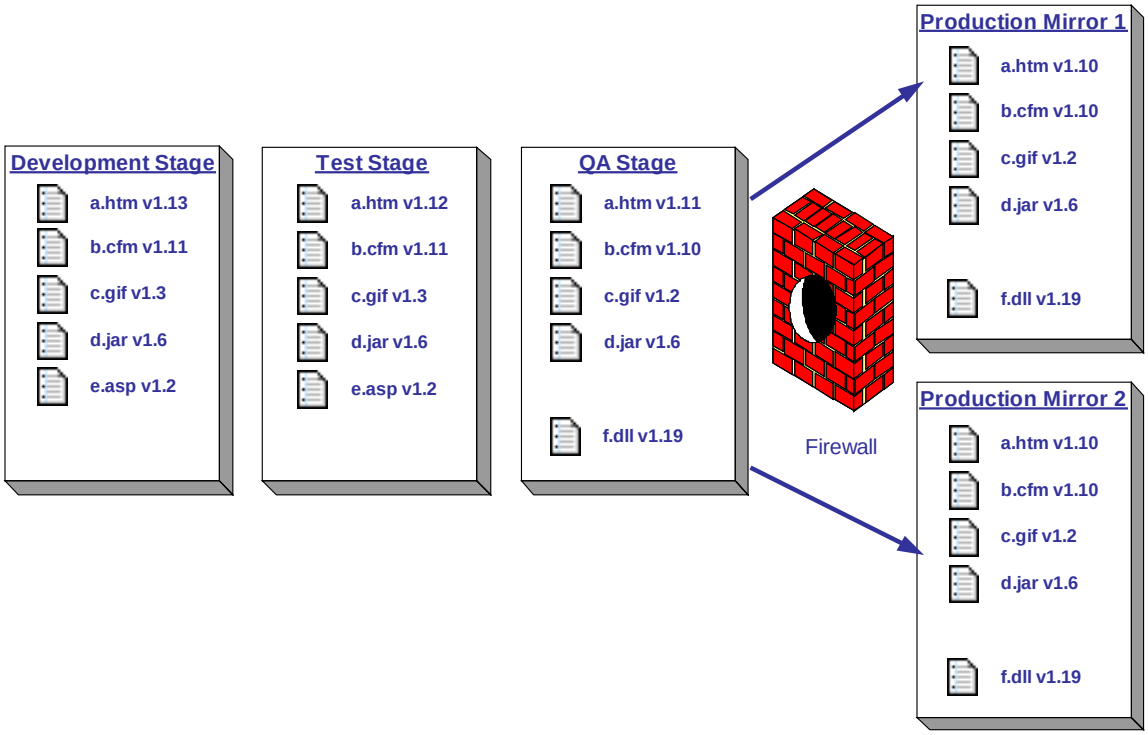

Figure 5: Example of Content Versions in Different Stages

In this example, there is a different version of the artifact `a.htm` in each stage as it awaits review prior to being promoted onto the next stage. The file `b.cfm` is awaiting approval out of the development stage, `c.gif` is awaiting approval out of the test stage, and `d.jar` is the same version in all the stages. The file `e.asp`, which provides functionality previously provided by another file, `f.dll`, only has been approved out of the test stage and doesn't yet exist in the QA or Production stages where `f.dll` still resides.

Under this scheme, all artifacts that exist in a given stage must exist in all the upstream stages (unless it is in the process of being deleted as `f.dll` in the above example) in a version that is greater than or equal to the version in the given stage.

Obviously then, one of the requirements of the staging system is that in the event that a change is rejected (which can happen in any stage except the first) the appropriate versions of all the files that were affected by the change need to be rolled back to the appropriate versions.

Promotion

The number and types of approvals required in any stage to allow a change to move from one stage to the next is defined by the business requirements and implemented in the staging and approval system. Once all the approvals required in one stage have been received, the change is promoted to the next stage.

[3] Note that syndicated content or live streaming content may not be subject to approval and thus only be referenced on each stage as it would be in the production environment.

Flexibility is a key element. Depending on the needs of the business, the approvals required in one stage may be as simple as having any peer review and approve the change. In other cases, the business may require that person A *and* person B *or* person C signify their approval prior to the change being promoted to the next level. The rules governing approval can be defined to any level of complexity required by the business. These rules define what hurdles need to be cleared. In the event that an emergency fix needs to be applied, someone with sufficient privileges can override the default approval rules.

Deploy to Production & Notification

The purpose of the change management system is to get changes out to the site. However, the site is usually hosted on a number of computers, each of which handles part of the load on the total site. This can include multiple peer servers that are behind a load-balancing switch behind a firewall. Depending on the functionality of the site and the business imperatives, the type of content that may need to be deployed includes:

- HTML pages
- Images – gif, jpeg, flash, bitmaps, and so forth
- asp
- jsp
- .DLL
- .exe
- Server Side Includes
- Personalization data:
    - Metadata
    - Rules
    - Portal configurations
    - Properties
    - User data
- Audio, and Video streaming content

On the surface, deploying content to a server is only a matter of moving files from one place to another. That is, in fact, one of the end goals (other goals being versioning, rollback, and recovery). However, the challenge lies in doing this in a manner that:

- handles all the types of content that need to be deployed to the production site
- cannot be used to compromise the integrity of the site (by introducing a security hole)
- applies all or none of the change (no partial application)
- tolerates failures at any point of the chain
    - failure in the host component prior to the update being sent
    - failure during transmission of either the source, the link, or the target
    - failure of the target after transmission but prior to the application of the change on the target
- recovers automatically from failures after they are rectified (except possibly catastrophic failures which may require operator intervention)
- provides status on the state of the deployment and notification in the event of failure
- allows operator intervention in the event of failure
- does not adversely impact the performance of the site (memory or CPU)
- can be scripted to perform any deploy dependent processing such as restarting servers or applications under certain conditions

Once a change has been successfully deployed to the production site, the originator of the change request can be notified that the change has been successfully deployed. This may further require the CRM be notified that the issue is closed.

The Vertical Sky Evolution Management Solution

Many technical managers view managing change as managing files. The Vertical Sky Evolution Management solution views changes, issues, and change requests as the atoms of control. Regardless of what you call them, the changes to the eBusiness are initiated for a business reason, either internally generated or initiated by a customer, supplier or partner. The company cares more about tracking the progress of these changes initiated for good business reasons than what may be happening to any individual file.

Developers, of course, work in the realm of files and development environments. The Vertical Sky Evolution Management solution unifies the business need with the developer needs by providing components that allow the different constituents to have the views into the system that they require to be active and effective participants in the change process.

The following diagram illustrates the different components of the solution.

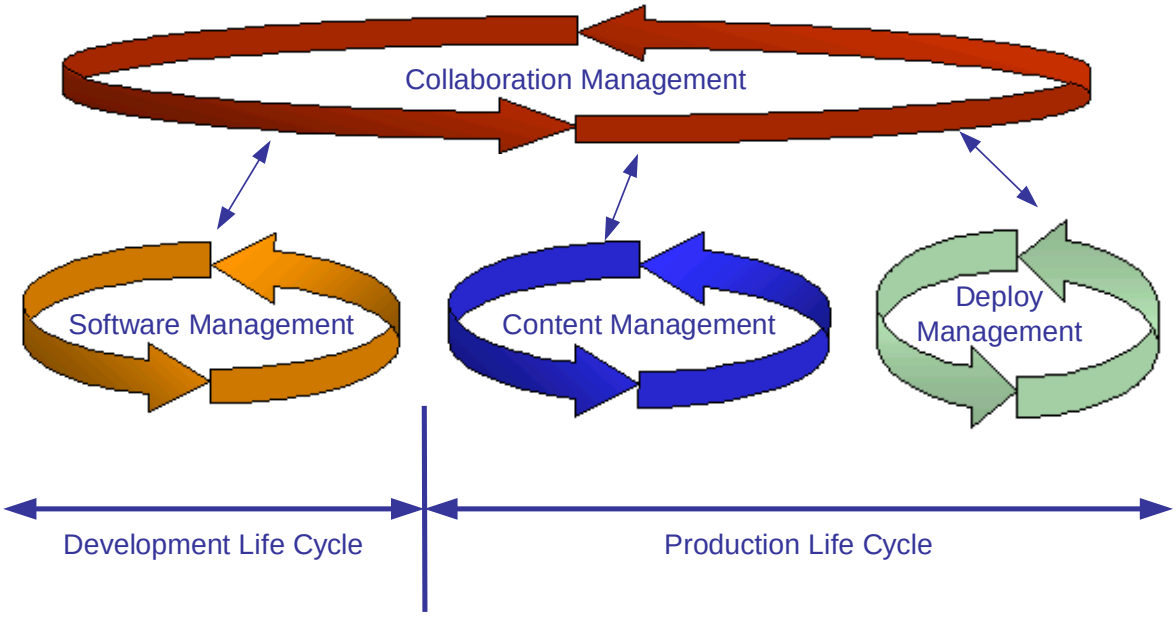

Figure 6: Components of the Vertical Sky Evolution Management Solution

Vertical Sky Collaboration Manager

Vertical Sky Collaboration Manager is tasked with all aspects of coordination between the different elements of the solution. It also acts as the manager's window into the change process. It is through Vertical Sky Collaboration Manager that the issues needing to be tracked are created, assigned, tracked, aggregated, and reported on as a collection of changes. It is the main interface point to other process pieces such as the CRM. It can allow customers and partners a view on the progress of important enhancements. It allows the definition and enforcement of workflows required to implement issue-based change management. By providing interface points to other solution components (both Vertical Sky and third party components) Vertical Sky Collaboration Manager acts as the traffic cop directing events, notifications, transitions to the appropriate parties. Figure 6 illustrates the different constituents that interface to the Vertical Sky Collaboration Manager.

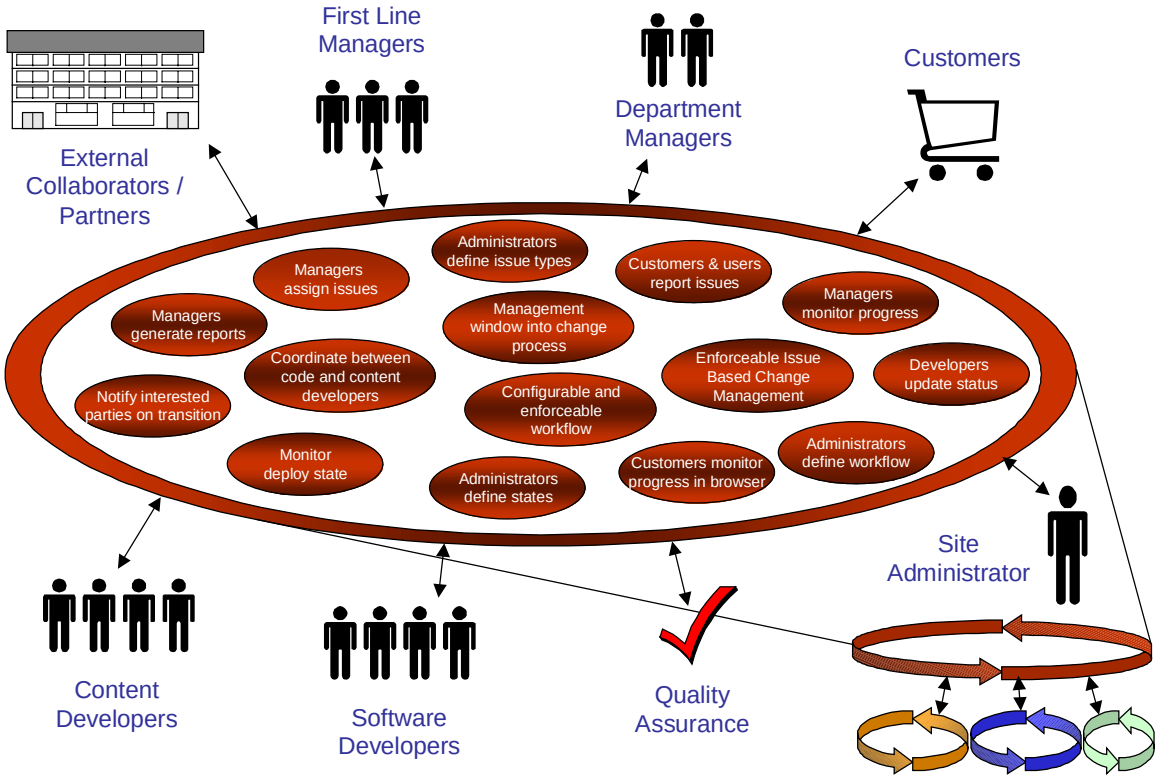

Figure 7: Vertical Sky Collaboration Manager Interactors

Vertical Sky Collaboration Manager is infinitely configurable. It can be used to define any kind of issue that can contain any number or type of field that can be used to collect information about the process or about the issue. The issues can be linked and all aspects of Vertical Sky Collaboration Manager are subject to the permissions granted by the administrator.

Vertical Sky Content Manager

The purpose of Vertical Sky Content Manager is not only to implement the staging and approval policies that an eBusiness requires to meet its change management needs, but also to provide all participants who need to be involved in evolution management with a window that gives them a view of the approval process appropriate to their level of technical expertise. Vertical Sky has emphasized providing approvers and different levels of content contributors with a view of the approval and staging system that makes sense to them.

At its core, the staging and approval system moves content from stage to stage so it can be viewed in context by the specified approvers[4]. Getting the content from the contributor to the approver requires that

[4] Once approved for production, the content is handed off to the Vertical Sky Deployment Manager, which is tasked with getting it out to the production machines.

three levels of contributor be supported: software developers, content developers, and non-technical contributors.

Software Developer

The software developer can use one of two methods to submit the built artifacts to the approval and staging system.

One, the developer establishes the list of artifacts that need to be deployed and the location to which they need to be deployed. Having established the mapping, the developer can then submit the built artifacts by using the integration between Vertical Sky Software Manager and Vertical Sky Content Manager. This Vertical Sky Software Manager integration into the developer's IDE handles the packaging and submission steps so the developer, having once defined the list of artifacts and their mapping, can submit subsequent versions of the artifacts from the IDE if required.

Alternately, the developer can explicitly move the built artifacts to their appropriate location in the Vertical Sky Content Manager sandbox and use the Vertical Sky Content Manager Sandbox view to package up the built artifacts and submit them to the approval system. This is the mode of operation for software developers using tools that do not support the integration into Vertical Sky Software Manager.

Both these methods allow the developer to specify which issue is being resolved with the submission.

Content Developer

For the content developer, Vertical Sky Content Manager provides two mechanisms for submitting their content for approval.

First, tight integration into the IDE allows the developer to bracket all the changes that relate to one issue by the creation and submission of a staging package. All checkouts and checkins happen within the context of the staging package. Integrations are provided for most popular content creation tools such as DreamWeaver Ultra, FrontPage, Visual Interdev, HomeSite, and ColdFusion Studio.

Second, for developers not using one of the supported IDEs, a sandbox view is provided that allows the content developer to explicitly create staging packages and check-in files into the staging package.

Non-Technical Contributors

The challenge in allowing non-technical contributors to participate in the change process is to strike the right balance between usefulness, ease of use, and complexity. By offloading much of the complexity to system managers and administrators, the non-technical contributor can be presented with an easy-to-use form that simply requires him to enter the information only he can provide. For example, a domain area expert (say, a product specialist) can be provided with a form where a customer's question and the expert's response are entered. The resulting form processing makes sure that the FAQ list and the FAQs are updated with the expert's input, without anyone having to deal with the HTML directly. Job postings, short articles, recipes, and meeting minutes are examples of content where the non-technical contributor can have an easy-to-use interface that results in valuable content being published with minimal interaction.

The non-technical contributor is optionally offered an opportunity to identify the issue addressed by his submission, simply by selecting from the list of issues assigned to him.

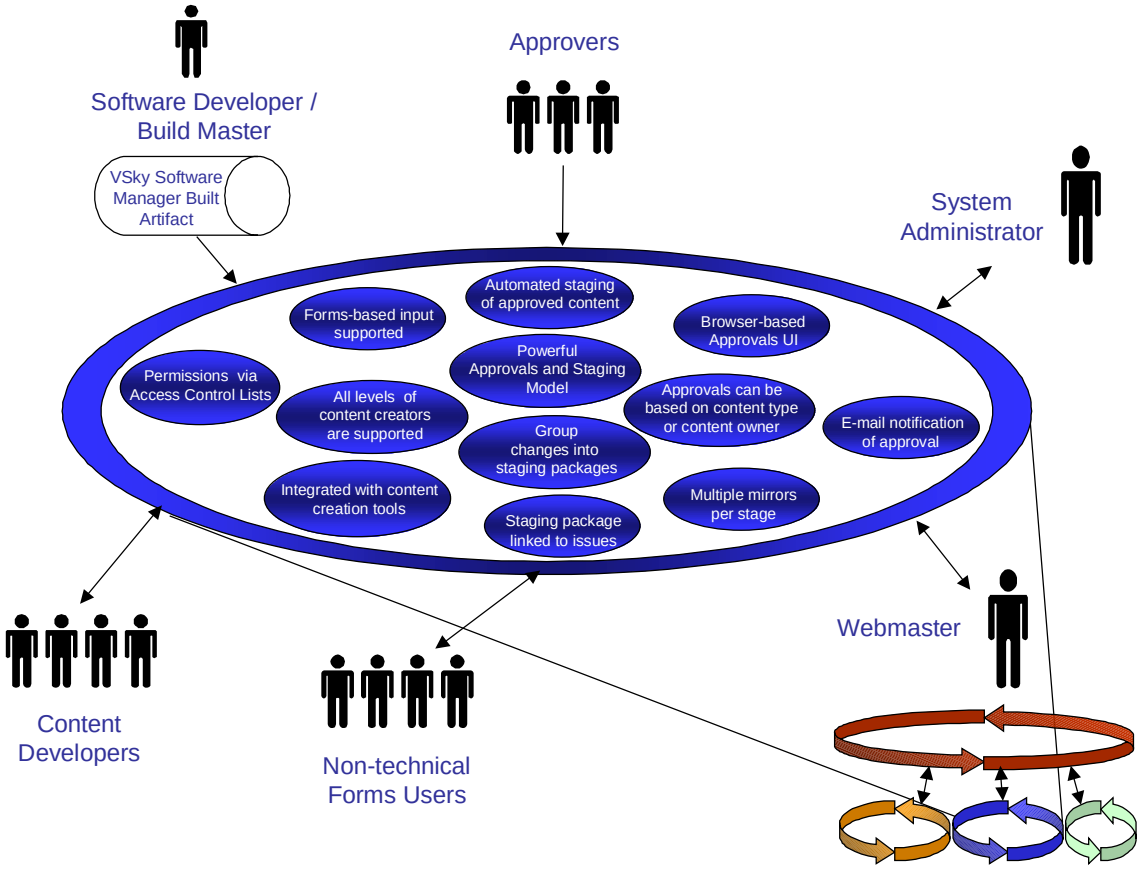

Figure 8: Vertical Sky Content Manager Functionality and Interfaces

Because approvers need to view the submitted material in context, the approver interface is hosted in the Web browser of their choice. Approvers are presented with the list of changes requiring their review in the selected stage. They can view the difference between the submitted change and the version of the changed file in the next stage. They can preview the change in the context of the current stage. They can promote a change to the next stage or demote a reviewed change to the previous stage. The staging system ensures that the correct version of the file required on the next (or current in the case of a rejection) stage is physically moved to that stage.

Vertical Sky Software Manager

Vertical Sky Software Manager provides software developers with all the features and functionality required to manage source files in large, distributed software development projects. As businesses evolve to eBusinesses and as eBusinesses evolve, change is required to corporate systems and to core software systems, to meet new needs. Software development teams are being asked to plan and execute changes of varying complexity. Getting a change out to the production site while ensuring it is applied to all the current development paths can only be successfully managed with issue-based *Software Configuration Management* (SCM).

Vertical Sky Software Manager allows developers to identify all the changes that, together, effect a particular change. Other members of the team, including the build master, can decide to incorporate the

change in their build area. Vertical Sky Software Manager allows individual developers to package all changes to the software project (adding files, dropping files, or modifying files) within a single change package, which others can then apply based on their need to incorporate an issue into their build environment.

The following diagram illustrates the functionality and interfaces to Vertical Sky Software Manager.

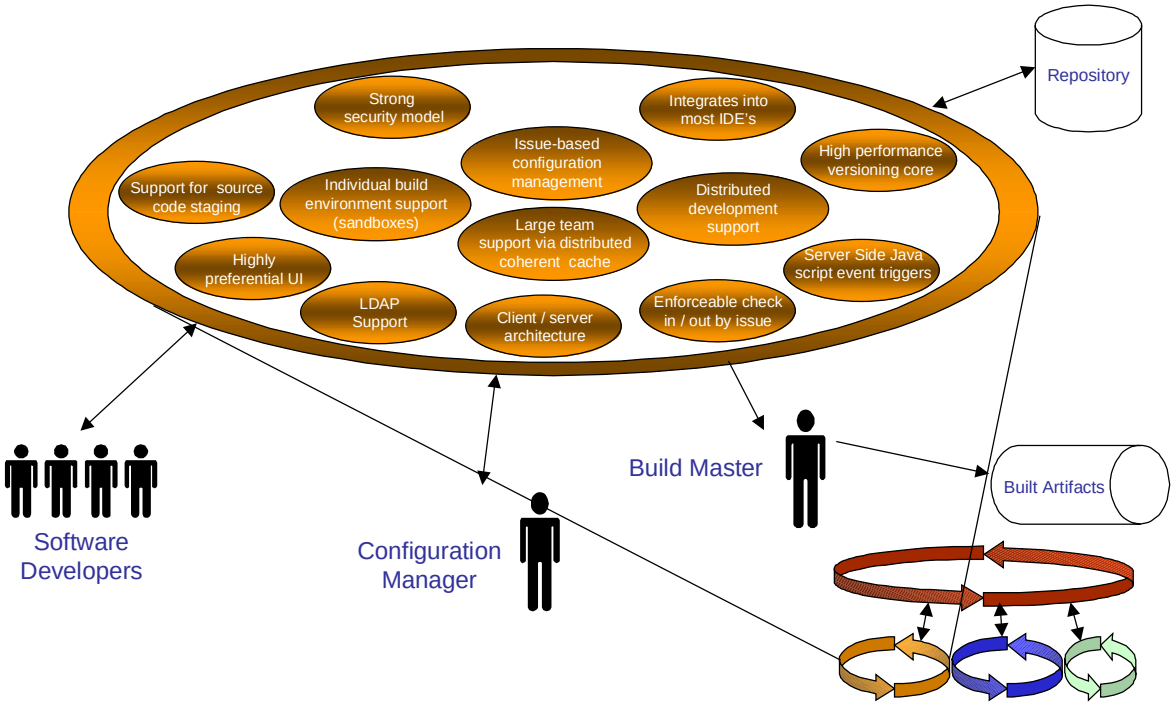

Figure 9: Vertical Sky Software Manager Functionality

Vertical Sky Software Manager is geared towards providing a very technical audience with the SCM tool they need to collaborate on large projects. It supports integrations into the popular IDEs as well as the CLI (command line interface) for developers who do not use an IDE, or who have a need for scripting. Its high performance distributed coherent cache is tuned to optimize the performance of the operations used most often.

Vertical Sky Deployment Manager

Once the approval hurdles have been cleared and a change has been deemed ready for deployment to the production site, site administrators may opt to have the Vertical Sky Deployment Manager immediately send out the change to all affected servers, where the deploy targets receive the change in a digitally signed container. After verifying the provenance of the update request, each target unpacks the change and moves the sent files to their appropriate location on the file system of the target. If no errors occurred

during the deployment of the sent changes, the change is committed and the status of the deployment is updated. If any error occurs either on the host, during transmission, or during the application of the change on the target, the operator is notified, the change is rolled back, and the source and the target coordinate the error recovery as soon as the error condition is rectified.

The following diagram illustrates the key capabilities of the Vertical Sky Deployment Manager.

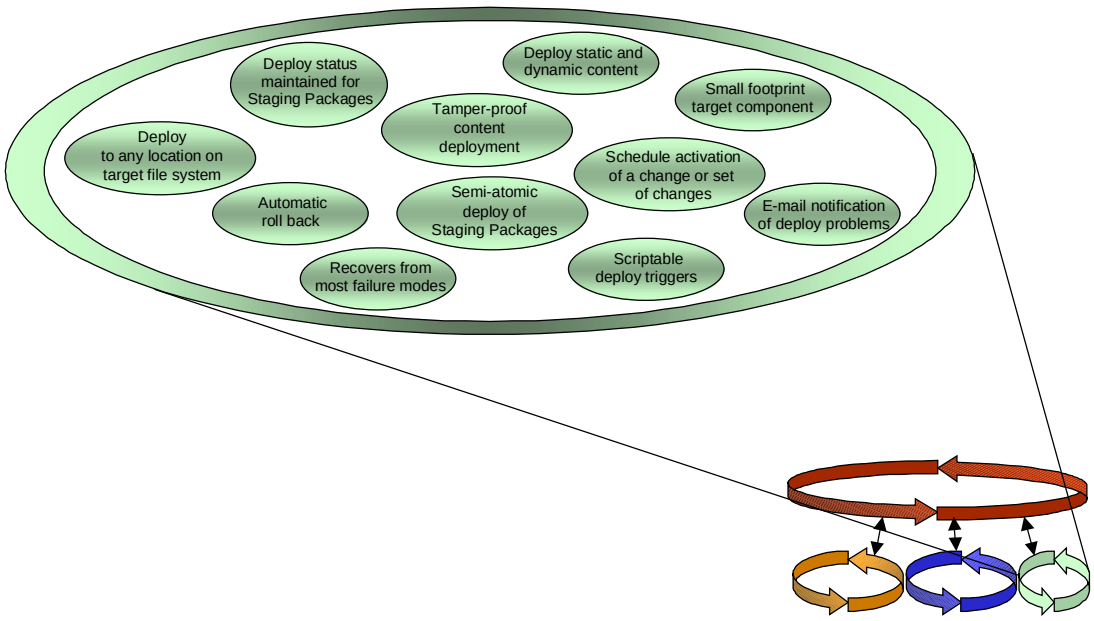

Figure 10: Vertical Sky Deployment Manager Capabilities

Vertical Sky Consulting Practice

It is always true that no two customer situations are identical. Different customers will naturally have different processes they want to implement, different types of non-technical contributors they want to include in the change process, and different platforms in their production environment. In order to close the gap between customer-specific needs and the capabilities provided by the components described earlier, Vertical Sky offers a wide range of services using a proven methodology for introducing change management at client organizations.

Starting with an assessment phase aimed at discovering the needs of the client with respect to the introduction of issue-based change management, and ending with post-project reviews, the process is aimed at ensuring that the change is introduced in a manner to capture ripe opportunities for establishing a track record of success with the newly introduced tools and processes.

Each company's specific needs dictate how much emphasis is placed at each phase and what the measure of success is for each phase. A company's ability to absorb change may be very high if all involved have previously experienced the pains of unstructured change, and as such there can be a very short time from assessment to deployment. In all cases Vertical Sky Consulting Practice are there as a partner in the installation, configuration, customization and rollout of the solution including assisting with the elaboration and implementation of the workflows and training of the trainers.

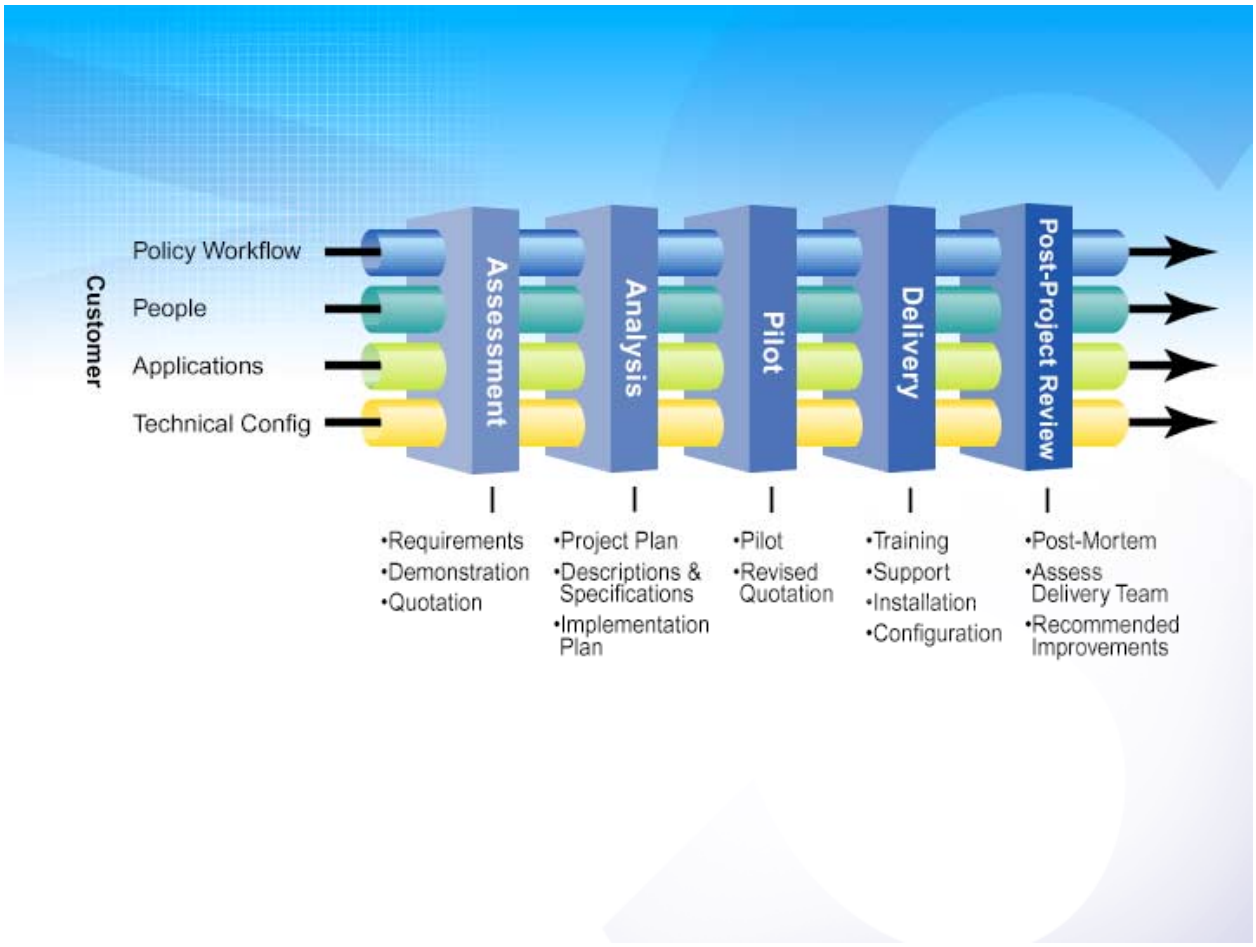

Figure 11: Vertical Sky Consulting Practice Process

The Vertical Sky Consulting Practice organization has the expertise to identify the needs of customers, identify the gap between the needs and the offerings of the individual components, closing those gaps, and ensuring that the solution installed for the customer provides immediate benefits and long term solutions.

Conclusion

Management of electronic assets is an absolute requirement for eBusiness survival and growth. Managing evolution in the new Internet economy includes all the challenges traditionally associated with that problem space, but accelerated to a rapid pace never experienced before. Companies who had previously been able to delegate the implementation of change to IT departments now find that, as an eBusiness, they need to engage everyone in the corporation to keep up with customer, partner, and market

demands. The circle of change management can be expanded to include different technologies, different skill levels, and different processes required to meet the business needs, only if the tools used to assist the business:

- are flexible enough to represent the business processes the company needs to implement,
- can implement the company policy decisions about process with the level of rigor selected by the company – "always enforced" to "bypass with sufficient permissions,"
- can implement the level of access control and security required by the business environment,
- present a view of the process that is appropriate to the level of technical expertise of the participants in the process.

The Vertical Sky Evolution Management solution, a mix of off-the-shelf components tailored by consultants with expertise in implementing change management systems, ensures that the particular company needs for issue-based change management are delivered in the manner most appropriate to the specific customer.

By elevating change management from managing files to managing issues directly, management can regain control over the full range of change complexity. Instead of the typical cycle where non-urgent requests become urgent and cause a crisis, important but non-urgent requests are less likely to fall through the cracks when management is able to constantly review the progress of change through the organization, spot bottlenecks, and remedy process problems before they get out of hand. By integrating code and content the planning horizon can be extended to allow informed tradeoffs to be made between the short term and the long-term impact of decisions.