The enterprise change process from IBM
White paper
July 2009

Rational. software

The IBM enterprise change process.

Alec Bray, IBM

Contents

Executive summary

Managing change requests is difficult. The needs of customers must be satisfied, and product defects must be resolved without reducing productivity. Some changes require extensive analysis—others are simple to analyze, but their resolution involves many components and groups throughout the enterprise. No change management process seems to fit an enterprise involved in complex systems development.

The IBM® enterprise change process (ECP), an extensible, industrial-strength process based on years of best-practice experience, coordinates enterprise changes that reach across the business, from marketing to training, requirements gathering to testing, and modeling to methodology. The ECP manages the development of systems and systems of systems. Metrics identify where error is introduced in the product, pinpointing process improvement solutions. Cumulative records improve estimation and project performance.

The IBM enterprise change process solution is an enabler for higher enterprise-wide development maturity, supporting initiatives to implement Capability Maturity Model Integration (CMMI), Six Sigma and IT Infrastructure Library® (ITIL®) standards, as well as other maturity and capability models and standards. With the ECP, tailored perhaps to suit the needs and processes of the enterprise, there is now a way to introduce, manage, track and report on customer-deliverable changes that involve all parts, all groups and all management areas of the enterprise.

Introduction

A company that designed and developed office-based networking products received an urgent phone call from the hardware manufacturing subcontractor. It turned out that a specific software-programmable chip was failing at final assembly test. In effect, a complete production run was lost, at some considerable cost to the hardware subcontractor.

Highlights

The hardware designers had specified a certain manufacturer's chip by name and part identifier. But the hardware subcontractors' suppliers had substituted a generic equivalent, which had a different substrate thickness. This changed the latency of signals through the chip semiconductors and so changed the timing of signals through the gates.

A defect that can be addressed with a single change request can still cause a ripple effect of changes throughout the project.

Although a single change request could be issued to correct the defect (simple replacement of the problem chips on the motherboard), there were, in fact, a raft of changes that needed to be put in place to rectify this situation.

The hardware subcontractor had to work with the supplier to source the specified part, and change its own receiving inspection to identify specific parts, rather than generic parts, where required. The hardware designers had to be reeducated to design a revised motherboard using generic components, rather than rely of the performance of a specific component that might prove to be difficult (and expensive) to resource. All data sheets and hardware design/Computer Aided Design (CAD) tools had to be reworked to allow for generic component specification. This meant, too, changes to the data sheet information used. The software designers had to both rework the code downloaded to the LCA and make sure that their application software did not rely on the throughput timings based on substrate thickness. Quality, hardware manufacturing, invoicing and many other departments had to ensure that the parts lists were brought up to date. Manuals had to be checked, and if necessary revised, where any mention was made of the programmable logic. Quality, sales and technical support departments informed existing customers that hardware failure might involve software upgrade as well as hardware rework.

Although it was only a single change to the product, this change touched nearly every department in the company. It changed, too, the process for product development.

Highlights

Enterprise change management can boost profits by enhancing productivity and tying together business needs and IT infrastructure.

This real-life example demonstrates the interrelatedness and complexity of change. As products become more complex (indeed, many products are systems of systems), more components and more areas of expertise within an enterprise are touched by one single requirement for change.

A series or set of change requests could be raised to cope with the diverse changes needed to requirements, manuals, design and development, subcontractor and supplier control. But without a way of recognizing these relatively independent changes as implementations of one overarching change request, it becomes extremely difficult to implement the change as a whole and to identify when the product and production process have been corrected.

This document introduces some of the features of enterprise change management (ECM). We consider change from the perspective of the activities needed throughout the enterprise to deliver the change requested or required by the customer. Tool support is essential to provide the control required, and the ECP from IBM is described.

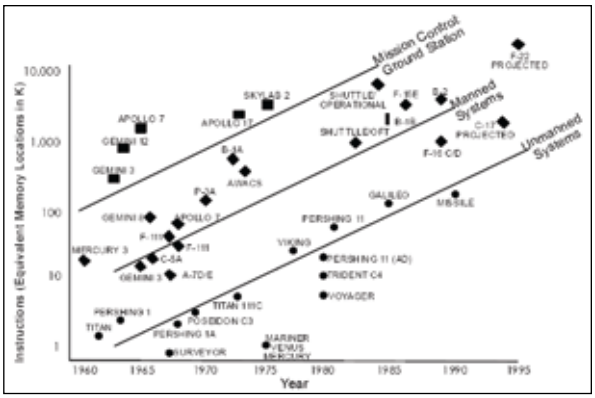
Enterprise change management can improve the bottom line because it enhances productivity throughout the enterprise. ECM looks at the enterprise's lines of business, tying together the needs of the business and the developing IT infrastructure.

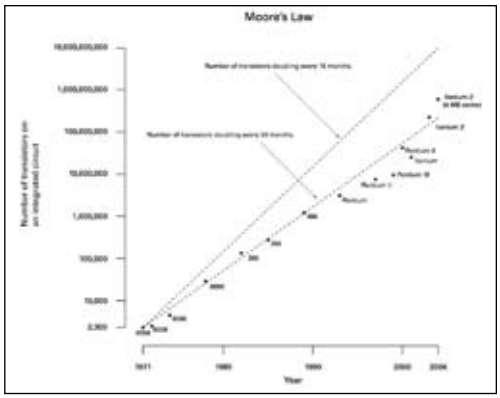
Enterprise change management

Change is the only thing unavoidable in software development.

The real-life example described above illustrated hardware and software development issues of a generation ago. The complexity of product is increasing each year. For example, today's mobile phones contain more software than the original lunar module.

The U.S. Air Force has published figures for the size of the software in U.S. manned and unmanned space missions. There is interdependence, of course, between the software onboard the spacecraft and the software in the ground station. Note that the number of instructions axis uses a logarithmic scale. This graph can be compared to "Moore's Law" for hardware.





Software instructions in spacecraft **Transistors per integrated circuit**

Figure 1: Software instructions spacecraft and transistors per integrated circuit

There are other interesting comparisons between the world of software and the physical world—and in particular to the zeroth, first and second laws of thermodynamics. Under this analogy, it is possible to consider some "laws" of software development in rather general terms.

Software systems tend toward a match with their environment. If the environment is changing, the software and software systems have to change, so software change is unavoidable.

However complex a software system may be, changes to the software do not reduce the amount of complexity. This seems to be true at all levels, from methods through classes, packages and components to systems.

Complexity in delivered applications is increasing very quickly, especially since the advent of object orientation and component-based development. Encapsulation and reusability allow systems developers to build higher levels of functionality. However, software complexity tends to create more disorder. Simply put, the "laws" of software development are as follows:

- *Software change is unavoidable.*
- *Incremental changes do not change inherent software complexity.*
- *To avoid disorder, software change needs to be managed.*

Highlights

An effective enterprise change management system will help ensure correct communication between software developers and systems engineers.

Not only is change the only thing unavoidable in software, but to prevent growing disorder, increasingly organized knowledge is required, in terms of change management, refactoring, software structuring and organization of the development—in other words, management of all aspects of software change. Here, there is more than an echo of "fail to plan = plan to fail."

With the increasing complexity of software, software redeployment and development in modular- or component-based systems, each component taking its own identity and release cycle, is becoming more and more a systems engineering problem. A relatively high-level overview of development is needed to ensure correct communication and information transfer between the software developers delivering the components and the systems engineering (and configuration management and release management) teams who assemble the system. Systems engineering and software development processes interact in a variety of ways, and any failures in communication can cause substantial productivity and quality issues. This, therefore, needs integrated control and management.

An ECM system provides a high-level overview of the needed changes and helps ensure that the right component is developed to support the final delivered system.

In a small organization, it may be possible for the change control board or executive committee to understand, contain and manage the changes across many of the organizations functional areas. This is not often possible in large enterprises where many factors come into play.

In a large organization with self-contained management areas ("silos"), each with its own objectives and budgets, the individual departments' concerns can take precedence over the enterprise's aims and direction. In a recent large-scale change, various agencies within an organization cooperated to deliver the required process change. But it was clear that there was no single person or agency responsible for minimizing disruption and cost to the enterprise.

Highlights

As a result, there could have been three major downtimes for one application: move, rework and upgrade. Only when the big picture was outlined did the managers realize that, with cooperation between departments, downtime and overall cost to the enterprise could be minimized.

An ECM system provides a high-level overview of needed changes and includes all departments or functional areas that must collaborate to implement the changes in order to meet the needs of the customer.

In a large organization, there may be other issues to consider. These include cross-discipline teams, which may understand the issues of enterprise change and which have processes in place to accept and implement it. There may be development areas that use agile or evolutionary approaches to development or use innovative design techniques, and whose current internal change processes may be alien to the enterprise as a whole. There is a need to develop a suitable method to control change across such systems of systems, especially where the individual systems may be developed under different processes, methodologies or design philosophies. In a small entrepreneurial organization, there may well be multiple simultaneous engineering functions and the need to control change across all disciplines.

Beyond simple project change control, ECM can address cross-discipline changes to organizational structures, business processes, roles and responsibilities.

However, ECM is more than simple change control; it is concerned with change to a customer deliverable, the implementation of which depends upon more than a single organizational unit. “Enterprise change” can therefore include changes to organizational structures, business processes, roles and responsibilities, and technology at the enterprise level. Because the change management is based on a process fundamental to the enterprise, change management effectively controls the change process. ECM becomes, in effect, enterprise process control. At times, these terms become almost interchangeable.

Highlights

Any ECM system has to recognize these various possible uses. It is possible to begin to look at the outline requirements for such a system:

- *Common process definition*
- *Extensible process areas*
- *Process control*
- *Reporting and metrics*

Common process definition

The ECM system has to be suitable for all activities within the enterprise or organization. To some extent, this becomes a "lowest common denominator" of the processes in use for each area. To identify these areas, the activities needed to implement any sort of change must be identified.

Each change must be analyzed to determine its cause, its potential effect to the organization and possible resolution paths.

In every case, the "change request" that has been raised needs to be analyzed. While there are many dimensions to this analysis, it is usually concerned with delivery to the customer: Was there a problem with the deliverable or could it be enhanced? In case of the latter, enhancement requests are necessarily considered in terms of the enterprise's strategic goals, whether these are financial (business), operational or performance (for example, quality accreditation) goals. There is usually no clear distinction between these. Some enhancement requests must be abandoned as they do not fit the organization's strategic objectives.

Once the analysis is complete and the change agreed, the change request is resolved. Resolution may take very different forms in different areas of the organization. Process change is undertaken by management and process engineering, monitored by a quality assurance function. Software change may involve recoding, rebuilding and reconfiguration of a deliverable or generation of new code, a new component or new deliverable variant (or combinations of all of these).

Highlights

With the change resolved, the outcome has to be evaluated before delivery to the customer. Again, evaluation covers different activities in different parts of the organization. Technical authoring output, for example, is reviewed, proofread and checked for assembly integrity (contents pages, index pages). Software development typically goes through formal acceptance and regression testing.

Extensible process areas

The common process definition provides the basis for a change system throughout the enterprise, but inevitably it cannot address the needs of every possible deployment. There are additional requirements:

- *Use in large enterprises*
- *Use in small enterprises*
- *Ease of use in straightforward change situations*
- *Adaptability for use in extensive or complex change situations*

Adaptability is key to an effective ECM system for both large and small enterprises.

An ECM system has to be inherently adaptable to the widest possible set of change situations faced by the enterprise, whether that enterprise is large or small. The smaller the enterprise, the greater the need for a straightforward approach that needs little inherent maintenance. Larger enterprises, generally more risk averse (not least due to the nature of their customers), need something that can address their various change management needs. One type of change may require more complex analysis than another, while another may require full formal system and user acceptance testing. Organizations need a solution that can extend any one of the process lifecycles while still maintaining overall control.

Process control

A change management system essentially represents the workflow of change, and as such, implements the process for change. Traditionally, a business process is viewed as a flow of activities, a "workflow," but some commentators view a business process as a trajectory in the space of all possible states. Within each state, there are three categories of behavior (ISO/IEC 10746-2):

- *Obligations: A particular behavior is required.*
- *Permissions: A particular behavior is allowed.*
- *Prohibitions: A particular behavior must not occur.*

Highlights

A robust ECM tool can track a business process's various obligations, permissions and prohibitions throughout all its stages.

Where a tool is used to represent the states reached by each individual change and to manage the obligations, permissions and prohibitions, the change process is directly supported by the change state tool. For example, a tool can mandate that certain information is provided at an appropriate stage in the process—and can ensure that the person in the correct role provides that information. Specific users can be barred from taking action—or specific actions can be prevented if prescribed conditions are not met. These sorts of control are difficult to implement in a "mandraulic" system.

Reporting and metrics

An enterprise change request, by its very nature, is not limited to an individual project, product line, line of business or process. Many different agencies within the enterprise need to know what the change represents and how the change is progressing.

If there are a number of changes outstanding on a specific delivery to a customer (which may be, for example, a specific development stream or release of software), it is important for product and project management to identify how near to completion that set of changes may be. The Maturity Index (MI) considers the number of changes in each of the states for the particular delivery and weights the results. The MI is used for trend analysis over the life of the project. The closer the MI number is to zero, the more mature the project. It is of interest to note that this approach is not specific to software development. Mature industries use a similar approach. Olive farmers, for example, use the number of olives in each stage of development in a sample in a weighted analysis to identify when the olive orchard is itself "mature" and should be harvested (or, of course, delivered to the customer).

Change may be managed throughout the enterprise, using resources available in many different geographical locations. Because a virtual team needs possibly more management than a small, co-located group, communication becomes vital. For project management, one important metric is the "burn down" of the resource allocation for the change, which may be calculated on the basis of hours used, but in agile developments it may be in units consumed.

Highlights

The "burn down" rate is an equation for determining hours used or units consumed and is an important metric for effective project management and change control.

The "burn down" rate depends, of course, on a good estimate of the effort needed to do the work. The enterprise change process provides for the capture of the actual time spent in each of the three main phases of analysis, resolution and evaluation. These actual times can then be compared to the initial estimates, and the results fed back into the estimation process. Using this comparative data and the accumulating history of project estimation data by size (however calculated), project managers and team leaders can monitor and improve the accuracy of the estimation process. This is critical for improving overall project management. Many standard processes fail to optimize and to give an audit trail of management decisions.

Following high-profile financial scandals, particularly in the U.S., there has been increasing concern for accountable corporate governance, which has resulted in legislation such as the Sarbanes-Oxley (SOX) Act in the U.S. While SOX compliance requires the prevention of manipulation, destruction or alteration of financial records, the remit requires the enterprise to define specific processes for appropriate compliance audits with further inspection of and policing of conduct and quality control. Other agencies require compliance to specific legislative requirements or codes of conduct. Healthcare systems may, for example, require strict observance to privacy requirements and ensure the security (in all senses) of patient data. Realtime safety-critical systems need to assure that changes to the system do not increase risk to the user, and often need traceability of change down to the individual lines of code, or to sentences in documentation or to individual requirements that are changed.

Capability maturity

Institute of Electrical and Electronics Engineers (IEEE) standard 610.12 considered the application of engineering to software development. But "engineering" developed slowly over many years and through many improvements in process. Compare the way that a blacksmith produced individual door handles, often to a bespoke design, to a modern industrialized design and development approach to manufacture of door handles.

Highlights

The processes are very different, the product of today has known tolerances and is a "component"—an alternative design with the same specified interfaces ("fit") is known to be usable. Software design has frequently been approached as a one-off activity with low potential for reuse, but is now maturing toward a systematic, disciplined, quantifiable approach to the development, operation, delivery and maintenance of reusable components.

The quality of a product is largely determined by the quality of the processes that are used to develop and maintain it.

It has been shown that the quality of the product is largely determined by the quality of the processes that are used to develop and maintain it. So the underlying process needs to be improved to improve the delivery to the customer.

CMMI

The Software Engineering Institute's initial Capability Maturity Model (CMM), initiated in part by the U.S. Department of Defense, developed out of customer dissatisfaction with continually receiving deliveries that manifested known, consistent software problems. The success of total quality management (TQM), Crosby's maturity grid and IBM's process grid led to the identification of features of development process maturity for software development. The Capability Maturity Model has now been through many revisions, and has integrated Capability Maturity Models from other disciplines. Capability Maturity Model Integrated, CMMI, is therefore not just about software development; as with all the various models, it looks at the maturity of process. That echoes IEEE standard 610.12: the application of a systematic, disciplined and quantifiable approach to the development of software.

Process maturity levels, as measured by CMMI, can provide insight into a company's ability to control organizational uncertainties and manage change effectively.

Process maturity levels identify how an enterprise moves from an ad hoc, chaotic environment to a mature, disciplined process. In the beginning, an organization may achieve some success but have no idea how the success was achieved and be unable to repeat the process with the next delivery to the customer.

Highlights

The organization is reactive—often it is a case of management by crisis. As the organization begins to "get a grip" on things, each individual project may have a suitable process, but the organization is still too frequently reactive. However, as good practice begins to be appreciated, the organization as a whole sees the value in applying process, which is then subject to measurement, analysis and control. At this stage, metrics mean money. With process in place, measured and managed, the most mature organizations look at continually improving the process across all areas of the enterprise.

Of course, there is more to CMMI than this, but if you have worked in an organization where the development teams are hacking away on their keyboards minutes before the due delivery deadline, or in an organization where management always seems to be saying "No" to any and every initiative, it's easy enough to place that low maturity organization in the CMMI maturity levels without much of an audit (level 1 and possibly level 2, respectively). (Note that capability is not the same dimension as maturity.)

As an enterprise moves toward higher levels of maturity, the enterprise is expected to consider its business objectives and then use them to define the process performance and quality objectives needed to achieve the business goals. Once the processes are defined, metrics are identified for the process (and subprocesses), collected and statistically analyzed to demonstrate how the various subprocesses are managed to achieve the business goals (as for example at maturity level 4 in the CMMI).

Six Sigma uses a wide variety of metrics to help organizations improve their processes and achieve better outcomes.

Six Sigma

Another flexible approach to process improvement (and product improvement) that uses metrics is the collection of techniques characterized as "Six Sigma." While Six Sigma uses data- and fact-driven process management, and has an avowed aim of improving the bottom line ("metrics mean money"), other themes are central to the Six Sigma culture. There is necessarily a focus on the understanding of the customer's current requirements and expectations, with proactive management and "boundary-less collaboration." There is of course a drive for perfection (the six sigma of the name), but process and process improvement are the keys to success.

Highlights

One critical phase in Six Sigma is the "analysis" phase [the five phases are define, measure, analyze, improve and control (DMAIC)]. Logical causal analysis looks to see where and how defects are possibly introduced to cause the failure mode as observed, and to be successful, it requires a set of accurate data—sufficient, adequate, correct metrics. With such data, it becomes possible to identify the variables (the environment, the equipment or the users) that correlate with the problem identified, and to identify the sorts of problems that occur most frequently. With this information, it is possible to be a "process detective"—a "forensic" approach to defect prevention. Perhaps users need more training or equipment that is adequate for the job. Perhaps the process needs reworking. Metrics are key.

Tools support helps eliminate manual change management steps for tasks such as adding and changing user permissions and tracking interdependencies between cross-department requirements and materials.

Tool support for process

Change management used to be paper intensive. For example, in one organization, a paper-based change request (CR) would be raised, based on appropriate grouping of paper-based problem report forms. Following a change control board (CCB), an engineering change order would be generated. The developer would then raise a paper-based form to request checkout of source. This, authorized by the departmental manager, would be countersigned by the quality manager, copied and filed. The developer, holding the red-marked form (the "token"), was duly authorized to access source.

This type of system breaks down very quickly when multiple agencies are involved in implementing a single primary change. Keeping track of the rights and permissions of individual users and roles, and the interdependencies between requirements, specifications, documentation, marketing collateral, training materials and code requires fast access to the appropriate information.

Further, tool support is needed to keep track of the parent system change and its individual dependent change requests through their complete lifecycles. Each CR needs to collect and integrate information from other sources. Interfaces to a multiplicity of source are required to ensure that all information is gathered.

Highlights

It is possible to set out the high-level requirements for a tool needed to support an ECM system. The tool has to be scalable to suit the type of change that is requested. Once the system is put into place, there are likely to be changes that are relatively easy to manage, and the tool should not impose high overhead on such work. A complex change, covering many disciplines and implementation areas, must also be handled by the tool without trivializing important activities and data collection. The ability of the tool to collect suitable metrics and display the analysis of such metrics in clear and concise reports is vital.

Ideally the tool should be deployable "out of the box" in a wide range of enterprises with minimal customization or configuration. The user interface should be familiar and easy to use and require minimal training.

The IBM enterprise change process helps organizations meet their change management needs by offering an "out-of-the-box" change management solution based on industry best practices.

ECP—features and functions

The enterprise change process from IBM is based on real-life experience of enterprise change management extending over many years in a variety of development environments, including research and domestic product development. The process is supplemented by best practices developed and implemented by IBM Rational® Change software consultants in many different enterprises, from governmental agencies (including defense) to service providers in a number of areas.

The ECP provides an "out-of-the-box" enterprise-wide change process solution suitable based on current best practices in change management. The IBM ECP enhances and extends the functionality provided by IBM Rational Change software and optionally offers direct integration with requirements change management where the requirements are managed in IBM Rational DOORS® software.

Highlights

The enterprise change process offers a number of features that address the issues raised in this white paper:

- *Five phases–submission, analysis, resolution, evaluation and conclusion*
- *Extensible stages–both depth and breadth*
- *Optional interlocked requirements change management*
- *Change request resolution management*
- *Reports to help support high maturity (CMMI) and Six Sigma*
- *Easy-to-use notifications*

The ECP offers five phases of change management. The three main phases are analysis, resolution and evaluation.

Five phases

The enterprise change process offers five phases for management of change:

- *Submission–A change request is raised.*
- *Analysis–The relevance and impact of the change are investigated.*
- *Resolution–Following acceptance, the change is implemented.*
- *Evaluation–The implementation of the change is checked.*
- *Conclusion–The actions on the change are closed.*

The three main phases of the ECP are analysis, resolution and evaluation, each of which contains two states:

- *A "doing" state or project phase*
- *A review and acceptance state or project phase*

In many change processes, review and acceptance of work done at one stage (or state) in the process and the subsequent transition of the change to the next state all occur in one single "transaction." The enterprise change process splits these activities by having two states in each main phase: a state in which the work necessary for the phase is undertaken, and an "end" state in which the work is checked, analyzed, reviewed and accepted. Only after the review at the end state, a state where the person responsible for the phase captures data for the phase and can add recommendations, is the change progressed to the next phase.

Highlights

As an example, consider the analysis phase of an enterprise-wide change request. The analyst may conclude that there is a technical risk associated with the implementation of the change request: It could compromise product quality or product performance. The recommendation made at the end of the analysis phase could be to not implement the request because of the level of risk. Nevertheless, the change control board decision could be to continue, to assign the change request for resolution. By establishing a clear separation between the "doing" phase and the following "review and acceptance" phase, an audit trail is created which can later be used to understand what happened if something goes wrong.

The extent of the review and acceptance undertaken is, of course, dependent on the nature and extent of the change. This can range from a sign-off by team leader or project manager right up to detailed analysis by a change control board. In this way, the ECP can be assimilated into many existing change processes and can offer the tool support needed for the process.

The phases identified in the ECP accommodate the activities needed for virtually any change. Specific areas of an enterprise may have their own definitions of activities needed, but these can map easily into the three phases of the ECP. Requirements engineers may, for example, review the outlined changes, prepare the changes for implementation and then review the set of requirements. Note that the ECP recognizes that requirements engineering necessarily has some specific detailed implementation needs. But at the high level, the process fits the analysis, resolution and evaluation paradigm. Software developers review the change, rework the code and test. Marketers review the proposed change to see if it fits their budget and marketing strategy. If it does, the changes are implemented and reviewed before publication.

The ECP provides a visual depiction of its current state and phase within the change process.

To provide some context for the current change request, the ECP offers a visual indication of its current state and phase within the change process as a whole. The layout of this is configurable. The sample below shows one possible implementation.

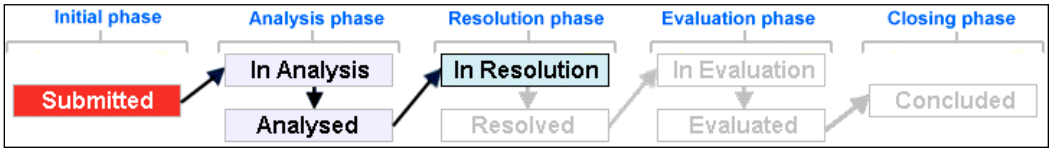


Figure 2: Example of a visual indication of current state and phase within the change process as a whole

This diagram is optionally shown within the current CR. Managers can clearly see that this CR is being worked on. Still to come in the process is a check on the current work before the complete change identified in this CR can be offered for evaluation.

Extensible stages

If a change request requires extensive analysis, a child change request is created with a lifecycle identical to that of the parent.

Some changes require extensive analysis before they can be considered for acceptance by a change control bay. Many change implementations successfully use this approach.

The enterprise change process approaches this problem somewhat differently. A child change request is created to handle the extended work in the analysis phase, but this child change request takes exactly the same lifecycle as the parent.

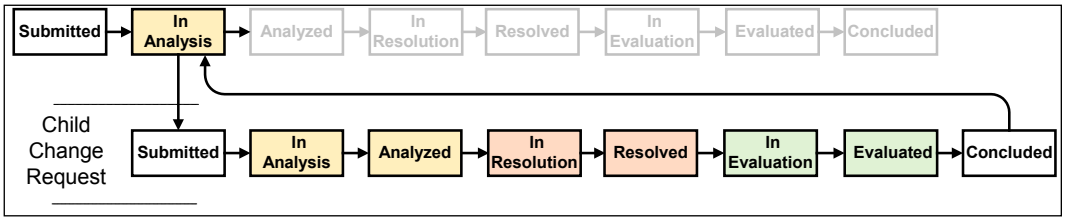


Figure 3: A child change request taking the same lifecycle as the parent

So the analysis phase itself had the phases of analysis, resolution and evaluation. For this child change request, the analysis phase is expected to be quite short. The resolution phase is where the bulk of the analysis work, which "resolves" this particular change request, occurs.

A child change request can be created at each of the main phases, and each child request follows exactly the same workflow (or "lifecycle") as the controlling or system change request. (There is one exception: a requirements change request. See below). Further, there is a single show screen for all users for all states for all workflows. This helps make it easy for all users in the enterprise to make use of the ECP because it is not necessary to learn specific workflows for different layers or elements of the total change process.

A single show screen may present too much information to a user at any stage in the change request's workflow. To help users focus on the information that is relevant to their role and task, IBM Rational Change software enables users to create role- and state-specific views. The enterprise change process approaches the same need from a different viewpoint. Although the show screen contains a lot of detailed information, the section relevant to the current state is automatically shown in full ("expanded"). By default, the other phase information is "collapsed." Navigation bars, typically at the top and bottom of the page, take the user directly to the phase information in which he or she is interested. To assist in this navigation, the three main phases are color-coded (colors are carefully selected to assist users who may have color-vision issues), and the colors are continued through the navigation bars and process help diagrams.

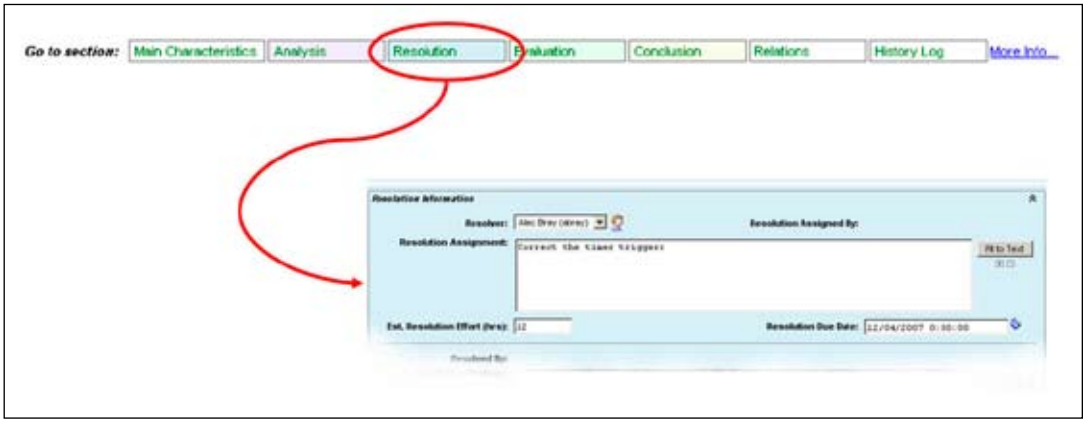


Figure 4: A role- and state-specific view of the change request workflow

Requirements change management

The enterprise change process optionally includes the change processes needed to support change to requirements managed in IBM Rational DOORS software. The main elements of the integration between Rational DOORS and Rational Change software include the use of system and implementation change requests and the requirements change request itself.

The requirements change request, developed with input from requirements engineers and requirements managers, is a proven workflow. In this workflow, a requirements engineer is assigned to make a change either to a requirement or to a logical grouping of requirements. The changes are not, however, made in the live requirements. At this stage, they are proposed changes only. After the proposed changes have been completed, the changes are submitted for approval. At this stage, they can be returned for rework. If the changes to the requirements are approved, the changes can then be applied to the requirements set. The main states in the workflow are as follows:

- *Created*
- *Assigned*
- *Reviewed*
- *Approved*
- *Applied (commit)*

Compared to the ECP, the requirements "resolution" phase is split around the "evaluation" phase—requirements are reworked without being implemented, and are then reviewed (evaluated). It is only after evaluation and approval that a "commit" is made into the requirements database.

Because of these differences, a specific requirements change request is maintained in the ECP. This has the benefit of allowing existing users of Rational Change in DOORS to transfer easily to an enterprise change system.

Using the ECP, a requirements change request is raised as a child of the main enterprise change request. This enterprise change request is best regarded as a system-wide change request, as it potentially covers all disciplines and customer deliveries of the enterprise. As such, this takes the place of the system change request that is provided as part of the "standard," "out-of-the-box" integration between Rational DOORS and Rational Change software.

Any implementation work required as a result of requirements changes in a "stand alone" DOORS change implementation handled by specific implementation change requests are, in the ECP, handled using the ECP child requests. These change requests themselves control changes in models, code, additional documentation and training resources. By using IBM Rational Change software, internal and external traceability in DOORS software, tasks in IBM Rational Synergy and Unified Modeling Language (UML) diagram relationships and code generation, there is more comprehensive traceability of the change and its various dependencies.

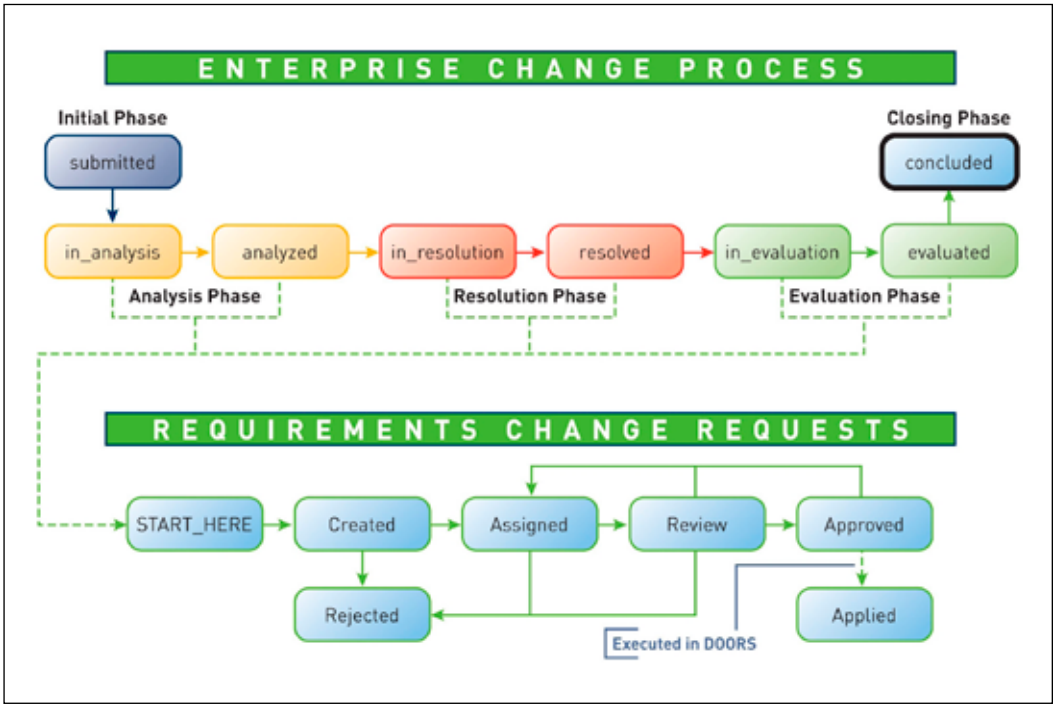


Figure 5: The steps of the enterprise change process

Highlights

Though the rate of change request submissions is unlikely to match the rate of resolutions, it's important to track these rates to ensure they fall within expected boundaries.

Change request resolution management

The rate of submission of change requests is rarely the same as their rate of resolution. Submissions are likely to be bunched, for example, with high rates of submission at particular stages in the complete project lifecycle. It is likely that resolution lags behind submission at every phase. Only when the product is deprecated are the cumulative totals for submission and resolution going to be the same, and not always then.

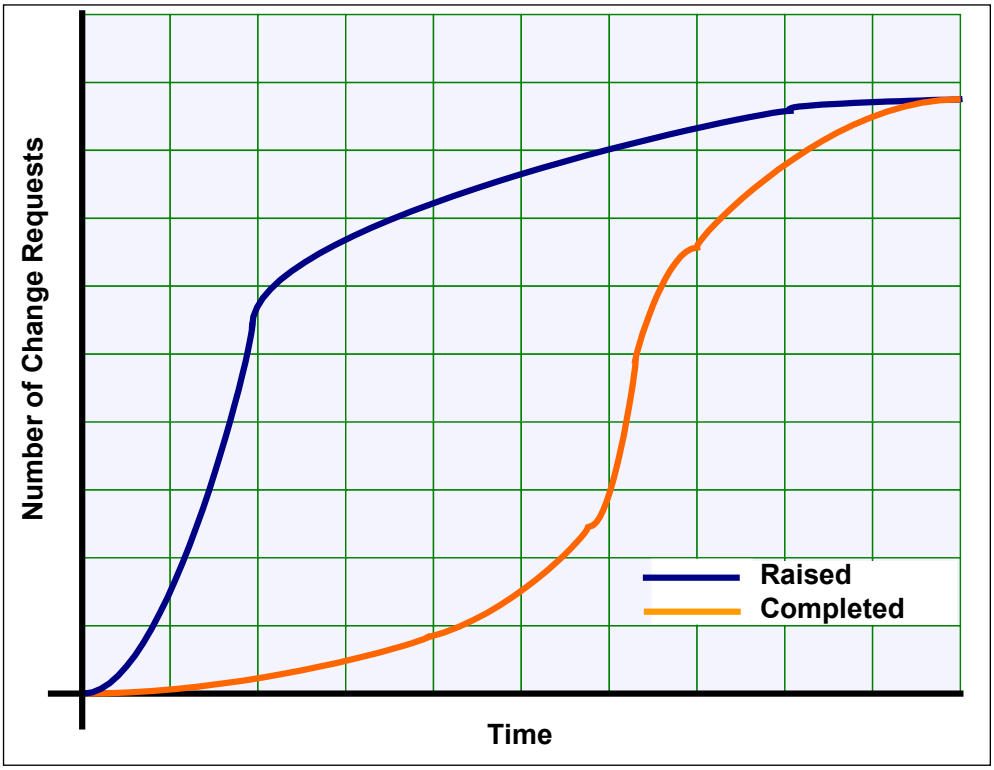


Figure 6: The number of change requests raised and completed over time

However, within each phase it can be expected that there should be a submission rate for change requests that is within certain boundary limits. If the rate of submission of change requests exceeds the upper bound, for example, the team should investigate why there is this unexpected increase in submission rate, as it may relate to error introduction at an earlier phase in development.

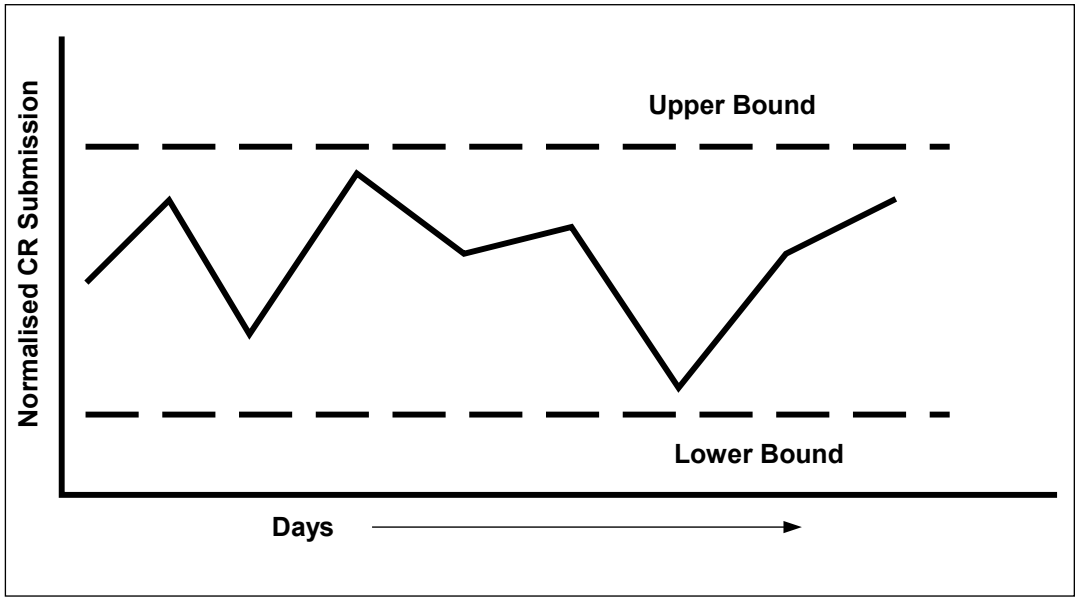


Figure 7: Change request volume fluctuating between expected limits

Each enterprise change request requires a number of people, environments and tools at each of the three main phases: analysis, resolution and evaluation. Providing resources for the resolution of change requests as they move through their various phases is a project management activity in itself.

Highlights

In the ECP, each change request has an identified manager, and the time spent waiting for a management decision is recorded. It is possible, for example, to determine the wait time between the end of a phase and the commencement of the next phase. Note that, in this context, when small teams are using the ECP, the assignment to the next phase can be automated through appropriate triggering. The change request manager, in the more complex environment, can identify when a change request has completed the "doing" part of a phase, and is now ready for review and possible approval. The manager knows exactly when he is expected to take a decision (a phase has ended) and can readily identify the next step. Tracking the latency between phases provides a measure of management reactivity.

Handling of change requests may be delegated or automated to alleviate or eliminate the change control board bottleneck.

In many enterprises, the change control board is often a bottleneck in the processing of change requests, especially when the rate of submission of change requests is high. Take, for example, the situation where the CCB is formally constituted to meet only on a regular basis. Large numbers of change requests accumulate between meetings, and the CCB has to deal with too many requests. Change requests are not processed quickly enough, adding to the issues of resource management later in the project, and possibly reducing the quality and functionality ultimately delivered.

An identified manager for a group of change requests, with delegated responsibility (within defined limits) to take decisions without referring the change request to the CCB, can accelerate change resolution. The assignment may even be automated based on some specific criteria of the requested change for a product or product family, development stream, component or subsystem. This approach releases the change control board to concentrate on the more complex or more costly decisions.

The use of a change manager can significantly improve the throughput time of enterprise change requests—some organizations implementing the ECP have measured an improvement of close to 50 percent of the throughput time.

Highlights

The ECP extends the powerful reporting capabilities of Rational Change and contains powerful metrics to pinpoint the process stage in which an error occurred.

The closer the discovery of an error is to the point of introduction, the cheaper it is to resolve.

Reports

Rational Change software includes powerful reporting capabilities, and these are extended in the ECP. Reporting project and product performance is a significant differentiator between low- and high-maturity organizations. The enterprise change process provides metrics to help improve delivered product quality and to better predict project performance.

Causal analysis and error containment

One powerful set of metrics identifies the process stage in which error is introduced into a delivery for a customer compared to the stage where the error is discovered. The closer the discovery of the error is to the point of introduction, the cheaper it is to resolve.

The analysis of the point of insertion of the error and its subsequent surfacing as an error, defect or failure can highlight inadequacies in the total development process. For example, error introduced during development and discovered during integration testing may point out the need for better developer training or improved working conditions or a more suitable development and unit test environment. An error introduced during the gathering of user requirements that goes undetected until user acceptance testing may point out that the user requirements were not adequately collected or understood in the first place.

One useful way to show this information visually is to use a matrix, identifying the number of errors, defects or failures identified at each stage, compared to the point of introduction of the error.

In the example below, an error introduced during the user requirements phase was not found until user acceptance testing. This is a serious problem for the process. In fact, the more points that gather toward the “point” of the arrow, the more the process is out of control. (And, conversely, the more points that cluster to the left side—particularly the top left—of the matrix, the more the process is under control.)

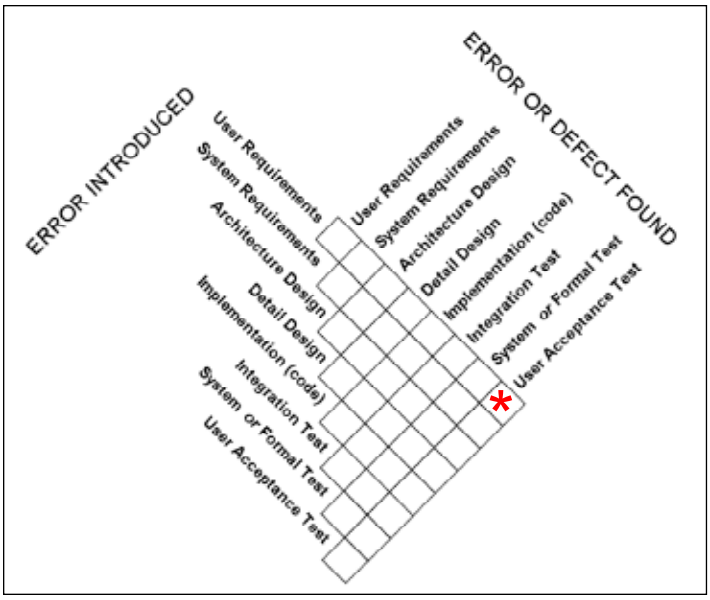


Figure 8: A matrix showing error introduced and error or defect found

However, one of the difficulties with this sort of analysis is understanding the phase in which error is introduced. The enterprise change process addresses this by requiring identification, during the change analysis phase, of the development phase during which the error was introduced (or caused). The stage at which the error is surfaced or discovered is also reported on the change request. Using this data, specific user home page matrix reports can be constructed to show the required metrics "at a glance."

Allied to this analysis are two other measures. The first considers the percentage of error introduced and detected within a phase compared to totality of error introduced in that phase (in the matrix above, reading the "rows" from top left to bottom right). This determines how effective the process is at identifying error within the phase of introduction (in terms of the matrix, keeping the points clustering on the left side).

The second measure looks at the percentage of error introduced in earlier phases of the process and detected in the current phase compared to the totality of error introduced in earlier phases of the process. This is equivalent to an analysis of the information maintained in the bottom right to top left diagonals (or "columns") of the matrix.

These two measures are termed phase containment effectiveness (PCE) and phase screening effectiveness (PSE) respectively, and, as can be seen, are tightly related to basic causal analysis.

Phase containment effectiveness looks at the number of errors that are introduced, detected and reworked within the same phase, compared to the total number of errors introduced during that phase irrespective of where these errors are surfaced later in the process. In the diagram below, the number of errors raised and identified during the coding phase has been recorded.

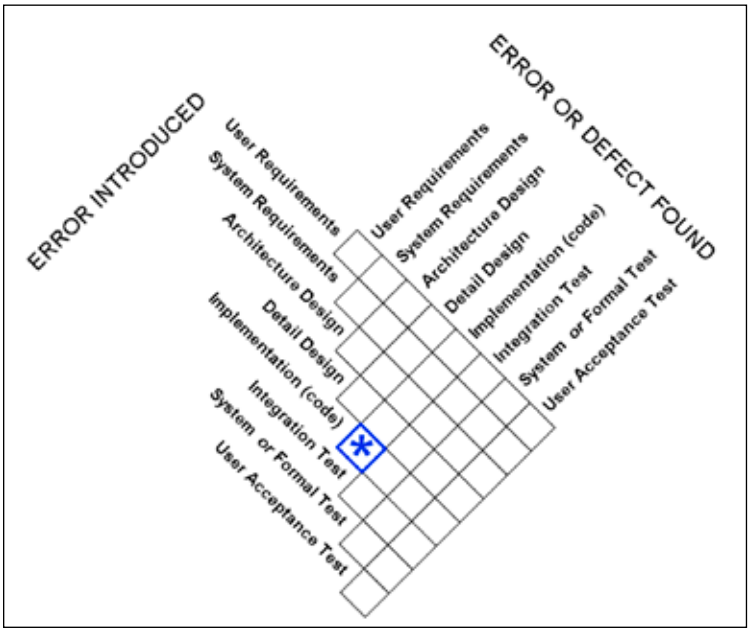


Figure 9: A matrix showing error introduced and error or defect found

The phase screening effectiveness looks at the number of errors that are detected within a phase, irrespective of where these errors were introduced, compared to the total number of errors recorded through the development lifecycle. In the diagram below, the number of errors identified during the coding phase has been recorded.

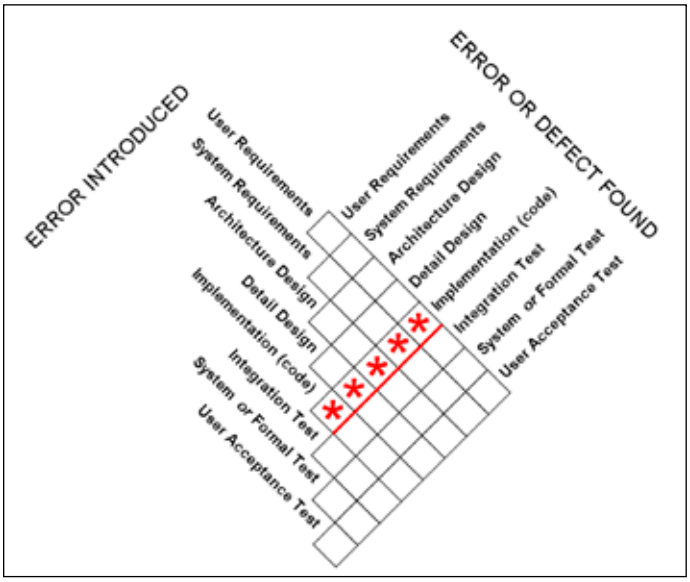
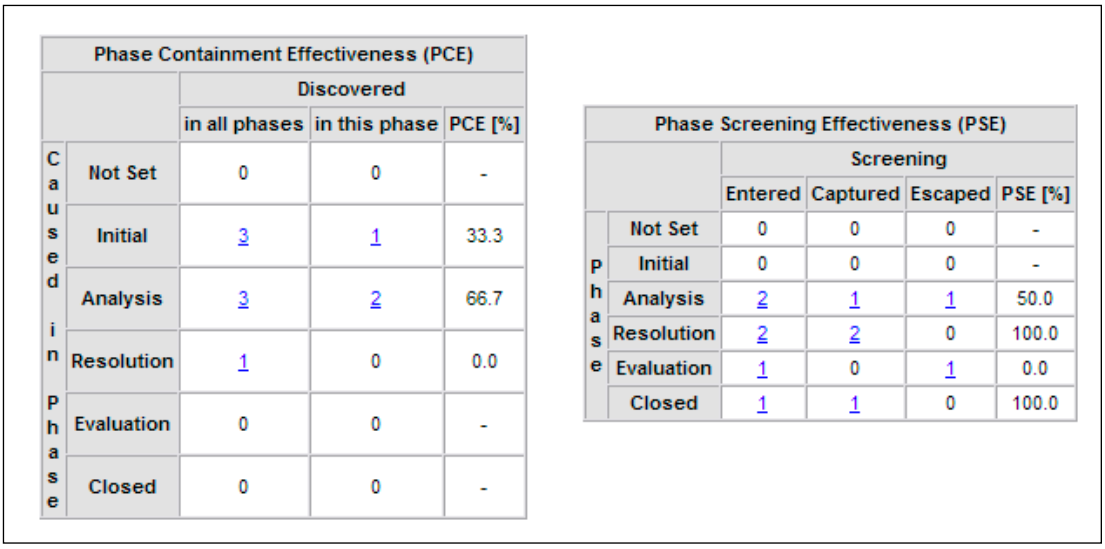


Figure 10: A matrix showing error introduced and error or defect found

There is a relationship between these measures: As phase containment effectiveness increases, the phase screening effectiveness decreases.

For more detail about the algorithms underlying these measures, the appendix shows an example calculation of the phase screening effectiveness from the causal analysis table. In the ECP, these reports are ready to run (no calculation required).



Phase Containment Effectiveness (PCE)

Caused in Phase	Discovered: in all phases	Discovered: in this phase	PCE [%]
Not Set	0	0	-
Initial	3	1	33.3
Analysis	3	2	66.7
Resolution	1	0	0.0
Evaluation	0	0	-
Closed	0	0	-

Phase Screening Effectiveness (PSE)

Phase	Screening: Entered	Captured	Escaped	PSE [%]
Not Set	0	0	0	-
Initial	0	0	0	-
Analysis	2	1	1	50.0
Resolution	2	2	0	100.0
Evaluation	1	0	1	0.0
Closed	1	1	0	100.0

Figure 11: Phase containment effectiveness (PCE) and phase screening effectiveness (PSE)

This approach considers the phase of introduction of the error as the cause of the error, in that, for example, a poorly structured requirement may have caused a design problem that surfaced as a defect during integration testing. However, another aspect of casual analysis looks at the "reason" or cause of the error—in the example, what caused the poor requirement to be written.

Highlights

The ECP enables comprehensive error analysis to help determine the error's primary cause and reduce the overall cost associated with it.

Using IBM's Orthogonal Classification scheme, the cause of defects can be considered to be due to:

Oversight	*(not all use cases or conditions were considered)*
Communication	*(not all necessary correct information was received)*
Education	*(information processed by inadequately trained personnel)*
or Transcript	*(simple error)*

The analysis provided from the ECP can be used to help determine the primary cause of error in a project, and so contribute to the reduction in cost.

Estimates of resource utilization and project maturity

Managing change includes understanding the technical detail of the change and the identification of interrelated changes needed to provide the updated deliverable to the customer. The changes, however, require resource provision, allocation and scheduling, which means that project management has a critical part in the successful delivery of change. Without good project management, there cannot be a good process or a good delivery to the customer.

Each change request in the ECP provides outline report data for project management. A "progress bar" shows the percentage used time to date compared to the original estimated time for the change.

Figure 12: A percentage bar showing time used against budgeted time

Note that this is per individual change request and is not a full burn-down chart for the complete enterprise change.

Highlights

Project management and release management are both concerned with the progress of achieving full implementation of the changes to be delivered to a customer.

With the ECW, a product can only be released or delivered to a customer when a predefined maturity threshold has been reached.

Where changes are implemented for a specific development stream, the current maturity of the deliverable is calculated by means of a user-defined formula, weighting the number of changes, by severity, in each of the states of the workflow to generate a single process maturity measure. The product can only be offered for release or delivery to the customer once the predefined threshold maturity of the project has been reached.

Easy-to-use notifications

The ECW's template-based notification system helps ensure information gets to the right people, at the right time.

It is very important that information about a specific change is made available to those who need to know. In any enterprise, it is convenient to distribute alerts concerning the status of any change by using e-mail. The enterprise change process uses a template-based notification system, which is designed to be easy to implement and administer, and which provides targeted e-mails with appropriate content to those people who need to take action.

The enterprise change process in summary

When looking for tool support for process, an enterprise should consider the costs and benefits of in-house development compared to deployment of commercial off-the-shelf products.

The enterprise change process can help organizations efficiently meet their process and workflow needs throughout the entire lifecycle of a project.

The enterprise change process, a commercial off-the-shelf solution that is designed to require minimal further investment, can be deployed in many enterprises "out of the box." With this supported application, an organization can rest assured that its investment is backed by IBM support and services. Further development of the change process to meet an enterprise's process or workflow control needs can be undertaken by consultants or customers with wide experience of process development.

The enterprise change process:

- *Provides an efficient and systematic enterprise-wide approach to change management*
- *Scales to support change management needs for large or small enterprises, for straightforward or complex change requests*
- *Optionally includes management and change of requirements (using IBM Rational DOORS software)*
- *Optionally supports requirements-driven development through creation of change requests directly from requirements and requirements-related documentation (using Rational DOORS software)*
- *Helps support initiatives to achieve higher capability and maturity levels by delivering metrics and reports that include support for:*
  - *Project maturity (project evolution)*
  - *Resource utilization per change request*
  - *Phase-based causal analysis (error introduction/error discovery)*
  - *Phase error screening and phase error containment analyses*
- *Helps provide the compliance support needed for Sarbanes-Oxley, CMMI, International Organization for Standardization (ISO) 900x, ITIL and other appropriate accreditations*

The enterprise change process is a core process within application lifecycle management solutions from IBM.

Appendix

How to calculate the phase screening effectiveness

To calculate the phase screening effectiveness from the causal analysis table, consider the pattern of distribution of error found compared to error introduced shown in the table below.

Phase:	Introduced:	UR	SR	Design	Detail	Code	Integ	Formal	UAT
Found:	UR	N1							
	SR	N2	M2						
	Design	N3	M3	P3					
	Detail	N4	M4	P4	PSE				
	Code	N5	M5	P5					
	Integ	N6	M6	P6					
	Formal	N7	M7	P7					
	UAT	N8	M8	P8					

Phase screening effectiveness for the detailed design phase (column 4, phase 4 in this table) is given by:

$$\frac{(N_4 + M_4 + P_4) * 100}{\sum_1^8 (N_n + M_n + P_n)}$$

Key

N4: number of errors introduced at phase 1 (user requirements) and discovered during phase 4 (detailed design)

M4: number of errors introduced at phase 2 (system requirements) and discovered during phase 4 (detailed design)

P4: number of errors introduced at phase 3 (design) and discovered during phase 4 (detailed design)

Stages in development

Abbreviation	Phase or stage of development
UR	User requirements or stakeholder requirements gathering
SR	System requirements definition
Design	(Architectural) design
Detail	Detailed design
Code	Development (code preparation, document authoring etc)
Integ	Integration testing or equivalent
Formal	Formal test, system test or equivalent
UAT	User acceptance testing

Project maturity

An example calculation for project maturity is shown in the table below.

Severity	1	2	3	4	5
State A	*A1 WA1*	*A2 WA2*	*A3 WA3*	*A4 WA4*	*A5 WA5*
State B	*B1 WB1*	*B2 WB2*	*B3 WB3*	*B4 WB4*	*B5 WB5*
State C	*C1 WC1*	*C2 WC2*	*C3 WC3*	*C4 WC4*	*C5 WC5*
State D	*D1 WD1*	*D2 WD2*	*D3 WD3*	*D4 WD4*	*D5 WD5*
State E	*E1 WE1*	*E2 WE2*	*E3 WE3*	*E4 WE4*	*E5 WE5*

In the table above:

A1: number of change requests in the state A and severity 1

A2: number of change requests in the state A and severity 2

… …

E5: number of change requests in the state E and severity 5

WA1 represents the weighting applied to the number of change requests in the appropriate state and severity.

The maturity of the delivery to the customer is given by:

$$\sum_{A1}^{E5} (M_N * W_{MN})$$

That is, the total, for all states and severities, of the weighted number of change requests for each state and severity.

For more information

To learn more about IBM Rational software, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/rational

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
July 2009

RAW14050-USEN-01