Change Request Management Best Practices
A “How to” document based on CQM by SQA

Revision 1.5
Abstract

This paper expresses the SQA Custom Quality Methodology (CQM) Best Practices for Change Request Management. It focuses on the challenges and benefits of implementing a well focused, disciplined, approach to Managing Change within the Software Development Lifecycle (SDLC).

SQA believes that effective Change Request Management is one of the cornerstones in the SDLC infrastructure. Understanding how an organization manages changes in software and documentation gives keen insight into what they think is important. According to James Bach, “Change control is vital. But forces that make it necessary also make it annoying. We worry about change because a tiny perturbation in the code can create a big failure in the product. But it can also fix a big failure or enable wonderful new capabilities. We worry about change because a single rogue developer could sink the project, yet brilliant ideas originate in the minds of those rogues, and a burdensome change control process could effectively discourage them from doing creative work”

Working with a wide variety of organizations and industries, SQA realizes that many standards exist in the CM arena. The Institute of Electrical and Electronics Engineers (IEEE) developed 828-1998 IEEE Standard for Software Configuration Management Plans. Carnegie Mellon University’s Software Engineering Institute (SEI) developed the Capability Maturity Model (CMM) and dedicated an entire key process area to Configuration Management. And the International Standards Organization (ISO) has developed both ISO 10007:2003 Quality management systems -- Guidelines for configuration management and ISO/IEC TR 15846:1998 Information technology -- Software life cycle processes -- Configuration Management. This document presents an implementation that takes the best practices from each of these standards.

In the Employee’s Survival Guide to Change from the Jeffrey Hiatt, editor for the Change Management Learning Center at www.change-management.com, we hear that “The field of change management can be confusing and sometimes complicated to research and study, especially for new practitioners. Change management is the application of many different ideas from the engineering, business and psychology fields. As changes in organizations have become more frequent and a necessity for survival, the body of knowledge known as ‘change management’ has also grown to encompass more skills and knowledge from each of these fields of study.” With this information SQA has assembled a group of Software Quality Engineers that can help an organization wade through the many types of change in their software projects and help to put them on a course for anticipating the change.

Benjamin Disraeli, Lord Beaconsfield 1804-81 said in a speech at Edinburgh, 29 October 1867, “Change is inevitable … Change is constant”. We at SQA know it is going to happen and know that our clients need to be prepared.

Authors: John J DeMassi, Ann Danby
SQA, Solutions Group
Table of Contents

I. What is Change Management? ................................................................. 6
   A. Change Request Management Process ........................................... 6
      1. Submitting ........................................................................... 6
      2. Evaluating ......................................................................... 6
      3. Reviewing ......................................................................... 7
      4. Scheduling ........................................................................ 7
      5. Implementing ...................................................................... 7
   B. Terminology ............................................................................. 7
      1. The System ......................................................................... 7
      2. The Change Request .......................................................... 7
   II. How Does Change Request Management Work ................................. 9
      A. Actions ............................................................................... 9
         1. Identification .................................................................. 9
         2. Instantiation ................................................................... 9
         3. Triage ............................................................................ 9
         4. Update ........................................................................... 10
         5. Closure .......................................................................... 11
         6. Deployment ..................................................................... 11
      B. Forms .................................................................................. 12
      C. Fields .................................................................................. 14
         1. Assignee ......................................................................... 14
         2. Description ..................................................................... 14
         3. CR Number ...................................................................... 14
         4. Status ............................................................................ 14
         5. Submitter ................................................................-------- 14
         6. Title ................................................................................ 14
         7. Dates & Times .................................................................. 14
         8. Attachments ..................................................................... 14
         9. Comments ....................................................................... 15
         10. Disposition ...................................................................... 15
         11. CR Type .......................................................................... 15
         12. Priority ........................................................................... 15
         13. Project \ Sub-project ......................................................... 15
         14. Resolution ....................................................................... 15
         15. Keyword ......................................................................... 15
         16. Severity .......................................................................... 15
         17. Reproducible Flag .............................................................. 16
         18. Customer ......................................................................... 16
         19. Where Found .................................................................... 16
         20. Version \ Build ................................................................. 16
         21. Release ............................................................................ 16
         22. Other Custom Fields ........................................................ 16
      D. Workflow .............................................................................. 17
         1. Change Request Transitions .............................................. 17
         2. Key to Flow Diagrams ....................................................... 18
         3. CR Flow .......................................................................... 18
         4. Urgent Issue Process ......................................................... 20
         5. Development Issue Process .............................................. 22
   III. Summary .................................................................................. 24

Bibliography ..................................................................................... 25
   A. James Bach ........................................................................... 25
   B. Carnegie Mellon University, Software Engineering Institute (SEI) ............................................................................. 25
   C. Jeff Hiatt, ............................................................................... 25
D. Tim Creasey, ............................................................................................................ 25
Why Manage Change?

As companies strive to manage their Software Development Lifecycle (SDLC) projects, they are inundated with requirements, software and data changes and defects. In order to manage costs, customer expectations and priorities, a Change Request Management methodology must be developed early in the process.

Like sailing a boat in a true straight line, continuous course corrections are needed. In software development organizations, knowing how to triage, prioritize, organize and manage change allows the boat (your SDLC) to stay on course.

Since change is inevitable, it would be insanity to know it was going to happen and to do nothing to manage it. The Change Request Management (CRM) System is a tool to be used throughout the SDLC. Here are just three benefits of a Change Request Management System:

1. A solid risk management program requires detailed information from the CRM.
2. Metrics from the CRM allow a project manager to evaluate the efficiency of and the effectiveness of the SDLC process.
3. Triaging and prioritization of issues and resources are determined by reviewing the details of the CRM.
I. What is Change Management?
Webster’s defines change in the following ways:

1. To cause to be different; An example of this is when the requirements implemented are changed. The change may be minor as in a color change from Green to Blue in a result set. The change impact may be well defined and clearly understood.
2. To give a completely different form or appearance to; Change can also be vast as in the change of the number of items in a results set from 10 to 1000. The impact of a large change is rarely thoroughly understood at the onset of the change request and requires analysis and planning.
3. To give and receive reciprocally; Sometimes a change occurs as the result of another change. You agree to make the Color Blue (from Green) if I agree to change the font from 10 to 12 points.
4. To exchange for or replace with another, usually of the same kind or category; Often the change requested is like swapping out one feature for another. The expectation is that one feature has the same impact as the newly requested one and can, therefore, be swapped out.
5. To lay aside, abandon, or leave for another; When the schedule is tight and the features are overwhelming your resources, a common type of change is the change that drops, deletes and negates a required feature.
6. To give or receive the equivalent of; Some times the change that is requested is swapping out a feature for something that is less difficult to implement but as functional as the original request.
7. To put a fresh covering on; Many changes are requested with the sole desire to keep the system up to date with the latest trend, style, fashion or technology.

A. Change Request Management Process

“All conservatism is based upon the idea that if you leave things alone you leave them as they are. But you do not. If you leave a thing alone you leave it to a torrent of change”. G. K. Chesterton, Orthodoxy (1908) chapter 7. If change happens, whether you leave things alone or not, then it behooves you to define a process for managing change. The American Society of Quality (ASQ) has developed a Body of Knowledge (BOK) for the Certified Software Quality Engineer (CSQE). The CSQE BOK breaks the change management process into 5 parts. We define a process derived from these parts.

1. Submitting
This is the process of formally documenting requests and submitting them for evaluation. Submitting, in companies where quality tracking tools have been implemented, is accomplished via an on-line, form-based tool. The data is typically stored to a commonly accessed database. Best practices regarding the building of form-based tools for implementing a Change Management System are provided later in this document.

2. Evaluating
During this process, a person or group with the authority to determine the priority, evaluates the request and sets the timeframe for its implementation so as to achieve the highest probability of contributing to the success of the current or future project or release. Later in this document, we define a team-based approach called a Software Configuration Control Board (SCCB) for evaluating, prioritizing and assigning
requests using the Change Management System.

3. Reviewing
This function is usually performed by a team of people who are tasked with defining the content of releases in order to meet or contribute to the product roadmap. In addition to re-assessing pending proposals (requests for change), proposals are re-evaluated for their effect on the release schedule, resource availability and improvement to the process. This group must judiciously determine those changes that will contribute to the product success with the least impact on schedule and resources. Later in this document, we discuss the Project Team’s focus in setting strategic and tactical direction for the team using the Change Management System for input.

4. Scheduling
Once the priorities and functional considerations with respect to the product roadmap have been evaluated, the schedule must be determined. A committee is needed to schedule requirements gathering cut-off dates, design time boxes, review types, development order and verification timelines. Later in this document, we use one example method to show scheduling and managing the release contents using the Change Request Management System.

5. Implementing
During this phase, Quality Assurance and Development work together to define the extent of each change. New test cases are devised and reviewed. Content is inspected. Documentation is reviewed and updated. Later in this document, we describe a method for keeping requirements in synch with your test case repository and in another section of the document we show the use of the Change Request Management System to manage the testing effort.

B. Terminology

1. The System
Rarely is a Change Request Management System referred to as the “Change Request Management” system. Most times it is referred to by the same name as the commercial product that is implementing the system. There are many vendors. Some cater to large companies with large products while others are geared more to the startup environment and other fall into the middle somewhere. In a search for Change Request Management Systems, you will invariably hear of tools like DDTS, Clarify, Test Director, TeamTrack, PVCS Tracker and many others. One piece of advice is to “sell” the team of people who will use the system on the label for the Change Management system. Use the label during the evaluation process, the PO process, in all usage documents and in the Kickoff Meeting for deployment of the new tool. The Change Request Management System is sometimes referred to by the name given to a change item. Some examples are MR system, ECO system, CR system etc…

2. The Change Request
Individual changes are often referred to by the name defined in the package that implements the process. (See Table I-1)
Here are examples of names for changes:

<table>
<thead>
<tr>
<th>CR</th>
<th>Change Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug</td>
<td>An error in logic or requirements</td>
</tr>
<tr>
<td>PR</td>
<td>Problem Report</td>
</tr>
<tr>
<td>Defect</td>
<td>An error in logic or requirements</td>
</tr>
<tr>
<td>ECO</td>
<td>Engineering Change Order is a type of request in the H/W or System development world</td>
</tr>
<tr>
<td>Enhancement</td>
<td>A change or addition to an existing piece of software</td>
</tr>
<tr>
<td>IR</td>
<td>Incident Report - presumes that the issue is a problem not a feature request</td>
</tr>
<tr>
<td>Issue</td>
<td>A generic name for an item to be reviewed</td>
</tr>
<tr>
<td>MR</td>
<td>Modification Report</td>
</tr>
<tr>
<td>Case</td>
<td>A generic name used for issues</td>
</tr>
<tr>
<td>TR</td>
<td>Trouble Report</td>
</tr>
</tbody>
</table>

Table I-1

The group responsible for purchasing and deploying a new system may employ the terminology of the new tool or may keep the terminology of the previous tool or may bring with them the terminology of the tool from a prior work arrangement.

The system proposed below uses Change Request (CR) as the name for a change. The label, CR, is neutral with respect to blame, cause, egregiousness and priority of the change being described. It is important to form a strong working relationship between those people that can issue a CR and those whose responsibility involves resolving a CR. To this end, a CR is NOT always a defect. To refer to every change as a defect, incident, problem or bug indicates that the creator of the software or system made a mistake. In our experience, there are many types of CRs. Some are defects and others are enhancements for new work. The labels ECO, CR and CR seem to best apply this principal. CR Types are described in the next section.
II.

How Does Change Request Management Work

Change Request Management is the process of tracking all of the changes that come from different sources and become part of a product. These Change Requests are tracked from the first awareness of need until the final disposition. Through the use of SQA’s CQM (Custom Quality Methodology) for Change Request Management, you are able to triage, prioritize, organize and manage change within in a client’s existing framework. Within this document are a series of actions, forms, fields and sample workflows that you can use as a guide of best practices for Change Request Management. We identify areas of variability where client’s framework may be different or non-existent with recommendations on how to address missing process.

A. Actions

To define “How” with respect to Change Request Management requires a definition of actions and workflow. Actions are described below. A discussion of workflow requires more in depth knowledge of the specific situation and an assessment of the gaps that need to be bridged.

1. Identification

This is the point at which a change request is recognized. It is an important step in the process. Changes may be initiated from many interactions. Discussions, analysis and use of the end product are very common initiation points for change requests. It is important to capture the context of the discussion, analysis or use accurately so that the change request can be evaluated from the appropriate perspective. Identification is part of the Submitting Phase.

2. Instantiation

This is the point where the change request is documented. It can be referred to as the creation, entering or inserting of a CR into the Change Request system. We recommend that the CR be documented in an on-line system. For the purposes of this document, all potential change requesters are provided access to the Change Request Management System for the purpose of instantiating new change requests. We understand that some organizations limit this access and perform Triage (see below) before instantiation. On-line submission is facilitated with a form designed to enter critical data for triage. The form has key fields like a Description, Title, Reference Number, Assignee, Submitter, Severity, Priority, etc. These fields, and the potential forms where you might see them, are defined later in this document. Instantiation is another part of the Submitting Phase.

3. Triage

If all change requestors are able to enter Change Requests, an efficient and effective means of disseminating work and defining the disposition of each change is needed. We suggest that a team be created. The team is called the Software (or System) Configuration Control Board (SCCB). This group’s members typically include the following: QA, Development, Test, Product Management and Business Analysis. While this list may vary depending on your organization, the intent is to include representatives for each stakeholder organization who have the authority to make decision on priority and ROI for a given CR. The primary leadership and first level of
There are two primary concerns for this group. The first concern is to review the fields entered by the requester with the following question in mind, “Is the CR properly categorized \ classified?”. This means that the SCCB is tasked with maintaining consistency in the content of the CR as well as overall enforcement of a common standard for the assignment of severity and priority. The second major concern is to define the next step in the life of the CR. (See the CR Workflow in a later section in this document for the possible transitions from the SCCB.)

There are other processes at work during the SCCB meetings.

1. The SCCB decides on next steps for each CR, which quite often involves defining the contents of the next Service Pack or Maintenance release.

2. The SCCB assigns work to groups or individuals outside of their normal chain of command. The department leadership needs to be active members of the SCCB to enable the assignment of work tasks.

3. Upward delegation is potentially a solution to a problem that can not be resolved due to the cross-functional nature of the request. In these rare cases, the SCCB assigns the CR to the person in the common leadership that is able to define a solution. Everyone in the process must be held accountable.

4. The final word resides with the SCCB. This means that the SCCB has to be shrewd enough to make the decision and when a higher authority is needed. Even when an SCCB does not exist, we always recommend a clear identification of the final authority for defining the disposition of each Change Request.

Details and tools for running the SCCB are discussed later in the document. Triage is part of the Evaluating & Scheduling Phase.

4. Update

The CR record tracks each step in the life of the Change Request. An online system has a form to record updates. Permissions vary by user, group, type of field and state of the CR. Because permissions vary by user, group, type of field and state of the CR, the forms for updating vary in the fields that are displayed. An individual or group may have the following permission for a field:

1. No view - no ability to see or modify the field
2. View - only an ability to see the value but no update capability
3. O_View - normally you can not see this but if you are the owner or requester you can see the value in the field
4. Add - an ability to see and add a value where it is blank
5. O_Add - normally you can add to this field but if you are the owner or requester you can add a value to the field
6. Update - an ability to add or update a value
7. O_Update - normally you can not update this field but if you are the owner or requester you can update the value in the field

The update form invariably contains many of the fields that are on the submit form. Permissions determine the user’s capability with any given field on any given form. Some reasons to update a CR are:

1. A new assignee is needed
2. More requirements are needed
3. More information is needed
4. A comment is needed
5. The change is ready for testing
6. The change is completed

Update is part of the Implementing Phase.

5. Closure
This is the act of setting a CR to the closed state. There is special processing and meaning associated with the process. “Closed” means the CR is ready for deployment. It has successfully completed testing. When a CR is closed, the following fields are typically set:

1. State (gets set to “Closed”)
2. Date Closed
3. Version Resolved
4. Release Name

Closure is part of the Implementing Phase.

6. Deployment
This step occurs when the change is actually deployed to the user. The CR represents a change that is being installed in the field. This installation may, and often does, occur after the closure of the CR. The change is ready but the field deployment date has not yet arrived. When a release of CRs is deployed, there are a few fields that get set:

1. Deployment Date
2. Deployment Release Number
3. Deployment Engineer

Deployment is part of the Implementing Phase.
B. Forms

Actions are applied to CRs via forms. Even if an online system does not exist, these forms should be used in paper form. The Submit, Update and Closure forms are the primary forms. However, forms are actually just groups of fields and their permissions, so to say that there is one submit form fails to realize the power of setting permissions to see and use fields depending upon the group that you are associated with in the CR database. Below is a basic Submit form from the Test Director™ tool from Mercury Interactive. A field that is red with an Asterisk means that the field is required. There are two forms of notification for a required field to aid the color visually-impaired. Notice that the submit form for this user (jdemassi) has mostly required fields. The desire with this solution was to have few fields with discrete choices.

Below is an update form. Notice that the submit form has only one tab but the update form has four tabs in order to capture all of the possible additions to the change record. There are many more fields on the first tab of the update form than the submit form - notice the vertical scroll bar on the right. The other tabs contain mechanisms for adding detailed comments; adding attachments such as pictures, log files and spreadsheets; and displaying the history of changes to the CR. Each tool has its own way of displaying this information. Tools typically also vary in capacity and field capability, such as maximum size of attachments, note and description fields, number of ancillary fields, amount of history and customization.
By the time the CR is ready to close, many of the fields are populated. The closure form may be an extension of the update form, as is the case with TestDirector™ or it may be a separate form that is only able to be launched when the CR already has certain fields filled or it may be a separate form all together with separate permissions.
C. Fields

There are many possible fields that a CR might need to describe, annotate and specify a change request. The following fields are needed to support the most basic workings of a Change Management Process. The critical fields are most often used for reporting status and providing general metrics. The primary fields and there descriptions are provided below.

1. Assignee
This person is currently “on-the-hook” to perform some action. The action may be to provide information, to solve a defect, to cost a change or to just wait and do nothing. The Assignee may be a person or a group. The process in the Workflow section of this document describes this dichotomy.

2. Description
This field is used to relay details of the CR. The who, what, when, where, why, how and whatever else is needed to describe the CR. One of the most important pieces of information for a defect type of CR is the list of steps to reproduce the CR. This is the place to be succinct, but detailed.

3. CR Number
The CR number is a unique tracking number that is most often system generated. In many tools you are able re-label this field to be anything that you want. In others, it is set to a specific label like one of the ones specified above in the Terminology section of this document.

4. Status
CRs are like people - they are born, they grow-up and they die. Hopefully, they do this cycle in a much shorter time period than a human life. We suggest that there are 3 main statuses - Open, Fixed and Closed. All other statuses are really qualifiers of these. In the system that we suggest, there is one additional state prior to “Open” called “New”. New CRs are ones that the Triage Team must review.

5. Submitter
This is the person or group that submits the CR for review by the triage team.

6. Title
The title is a short but descriptive statement of the CR. Try to keep this close to key words, modifiers and linkage type wording. E.g. XYZ window crashes when A field and B field are submitted at the same time. Use the description to go into more detail.

7. Dates & Times
There are several date fields that are important. The most important date for a new CR is the date and time when an issue first occurred. This is often referred to as the Date Found or Date Detected. Chronologically it is the “Born” date for the CR.

   • Date Closed
Date Closed is listed as a required field but most thorough systems set this field automatically when the user sets the CR to “Closed”. This field is used most often for reporting along with the Create Date. A simple subtraction gives the total time that it took to resolve the CR, which usually part of most Service Level Agreements (SLA).

   • Date Resolved
The Date Resolved gets set when the status of the CR gets set equal to “Fixed”. When you subtract the Start Date from the Date Resolved, you get the time spent in development. When you subtract the Date Resolved from the Closed Date, you get the time spent in testing. This is a simple way to determine how much of the maintenance effort is spent in resolving a problem vs. testing a problem.

8. Attachments
Very often a Change Request Management System has a way to attach files or pictures to aide in the description or re-verification process. Objects attached to a CR are most likely stored in a repository on the server running the Change Request Management software. In some cases, they are stored as Large Object fields in a relational database.
9. Comments
We recommend that the Triage Team and the Submitter be the only people able to change the description of a CR. The intent here is to ensure that the original message is never lost. With that in mind, we suggest that the Comments field be used as a chronology of events for the CR. Some systems even provide a header listing the date, time and person adding the comment to the beginning of each new comment.

10. Disposition
The disposition field is useful when the organization is large enough to be split into several groups and the business of resolving a CR must transition from group-to-group or person-to-person. The disposition field is a modifier of the Open state of a CR. Several values for this field might include the following:
1. Awaiting Requirements
2. Developing Technical Specification
3. Architecture Approved
4. In Development
5. Awaiting New Build for Test
6. Rejected Fix
7. Not Readily Reproducible

11. CR Type
The CR Type is used to define the type of problem. It must be set in order for the Triage Team to assign the CR to a person or group to investigate. Several examples are as follows:
1. S/W Defect
2. Requirement Defect
3. Data Defect
4. Documentation Request
5. Report Request
6. Enhancement
7. Data Correction
8. Project

12. Priority
Priority is used to assess the impact upon the Customer. We prefer to leave the number of priorities at 3 - High, Medium and Low. Quite often, it is necessary to split the High to better understand the customer's pain. In this case, the questions are simple, “Is the product useable without this feature?” If not, “Why not?” If so, “Is this feature needed at all?” If so, “Why?”

13. Project \ Sub-project
Very often, large projects are decomposed into several smaller projects, modules or components. In larger organizations, CR assignments are determined by the Project or Sub-Project. This field facilitates that arrangement and the complexities of managing large groups. It is also very useful to track this field in order to determine where the majority of the team’s effort is being exerted. If a Project has a high degree of error, it must surely be a candidate for redesign or at the very least, more thorough testing.

14. Resolution
This field should be required upon closure of the CR. It is critical to know what the resolution was so that accurate tracking can be accomplished. Think of this field as the reason why a CR was closed. Some possible values are as follows:
1. Fixed
2. Enhancement Implemented
3. Not a Defect
4. Overcome by Events
5. Not Reproducible
6. Data Issue Resolved
7. Report or Document Produced

15. Keyword
This field is used to categorize the CR. In larger Projects, CR assignments are determined by category or specialty. Similar to Project/Sub-Project, this field facilitates an understanding of the quality areas or problem areas of the product. If a category has a high degree of error, it must surely be a candidate for redesign or at the very least, more thorough testing. Some possible values are as follows:
1. Performance
2. UI Navigation
3. Accuracy of Data
4. Error Handling

16. Severity
Severity is one of those controversial fields. It is important to avoid confusing Severity with Priority. Think of Severity as the egregiousness of the...
problem. We recommend a special category for those Change Requests that are so urgent that they require a special process.

**Urgent** - a required function of the application is crashing or failing, data corruptions are occurring or the user is unable to or refuses to use the product. Urgent issues will trigger an Escalation Process. Issues of this nature are resolved as quickly as possible and often occur twice - once with a patch release and then once within a maintenance or major release.

In addition to Urgent, we think that 3 standard levels of Severity are plenty.

**High** - a required function or feature of the application fails but has an acceptable workaround or short-term solution. Does not include data corruption issues but does include navigation failure issues and specific performance issues like slow running, non-real-time reports and complex ad hoc queries not deemed essential to product operation. High severity issues will be solved in the next maintenance release or if the risk assessment indicates a low probability of error, the next patch release.

**Medium** - a feature or function that works as designed but requires improvement for usability. Navigation suggestions and new user requested features are examples. Medium issues will be resolved as necessary during maintenance and major releases.

**Low** - A new feature or suggestion related to cosmetic and formatting issues. Low severity issues will be resolved as necessary during major releases.

17. **Reproducible Flag**
This field is an indicator to tell other whether the problem is readily reproducible via some procedure. This field has impact on how the Triage Team decides on the disposition of a CR. (See the Workflow section later in this document.)

18. **Customer**
Changes require a champion for the Customer. Often the Submitter is the champion, but we need a field to capture who the most likely Customer is for a change. Knowing the Customer, and the Customer’s specific needs, will help to prioritize the change.

19. **Where Found**
Because of the many places where a CR may be born, we have a field to capture the phase or environment where and when the Change Request is initiated.

- CRs may be generated as the result of a production problem.
- The change may come from a Product or Project Manager who is setting the roadmap of the product by entering enhancements.
- The change may come from Development in the form of code examination or design discussions.
- CRs also come from the testing, evaluating and analyzing the results of a program.

20. **Version \ Build**
This field represents the version or build of software where the CR was detected. There may be several of these fields. Another might hold the version of software where the fix resides and yet another might represent the version actually released. This field presumes that the software delivered has a build or release version that pertains to all of the software.

21. **Release**
Often releases are given names. This field captures the named release for a given CR.

22. **Other Custom Fields**
Any good Change Request Management tool has fields that are customizable for growth and capturing further metrics.
D. Workflow

1. Change Request Transitions

Entry into the Change Request Management System begins with a Change Request. Change Requests have a finite set of state transitions. The diagram below shows one example of transitions in a Change Request Management System. The descriptions on the transitions give the user reasons why a transition might occur. Based upon your specific implementation, you may define other reasons for the transitions or you may define a different set of transitions. Keep this diagram as simple as possible and enforce it with business rules in the Change Request Management tool, and you will help your SDLC on-track.

Figure 1: CR Transition Diagram
Table II-1: CR Transition Table

<table>
<thead>
<tr>
<th>State</th>
<th>Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Opened</td>
</tr>
<tr>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>Opened</td>
<td>Fixed</td>
</tr>
<tr>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>Fixed</td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td>Opened</td>
</tr>
<tr>
<td>Closed</td>
<td>Opened</td>
</tr>
</tbody>
</table>

2. Key to Flow Diagrams
The following is a Legend to help the user understand the objects chosen in the flow diagrams below.

3. CR Flow
The example CR flow described below assumes that there are 2 main development branches for software development. This is a simplified example for the purposes of this document. It feeds into a simplified SDLC. Your SDLC and communication, tracking and reporting needs will cause a more complex flow. There is a current branch where Urgent issue changes will be made and there is a development branch where features and non-critical bug fixes for the next release will be made. If your organization is larger and has more branches, more yellow diamond decision points will be needed. For the purposes below, an Urgent-5 is a priority fix. The only time that a change is made in production is for issues deemed urgent by Product Management or Customer Service. It has been our experience that Product Management and/or Customer Service are the driving forces behind urgent production change request. Again, your SDLC may be different.

The flow below also introduces the SCCB’s role in the process. The SCCB is a frequent updater of CRs. The process flow below shows fields that are updated by the SCCB and next step for a CR once the SCCB has performed its review.

The assigning of a CR to a “Future Release” means that the relative priority is well understood and the change can wait some predefined period of time before it is implemented. The idea is worthy but not time critical. These CRs will feed into future release content discussion to Product Management and Customer support.

The assigning of a CR to the “Current Dev Release” indicates that the change will be made as part of the next release or at least it will make it onto the backlog for that release.
Assigning a CR to a person generally means that either more information or investigation is needed or the change must be acted upon in a more timely fashion. The Urgent Issue process is employed in the latter case.

Figure 3: Beginning CR Flow
4. **Urgent Issue Process**

In the workflow described above, there are two main branches for software development. We have a current branch, which supports the release in production. And we have a development branch, in which all production fixes are made and new development occurs. The figure below describes the Urgent Issue process.

The left hand side of the flow is designed to ensure that the Development Manager has the appropriate information in order to analyze and solve the problem. Once the CR is thoroughly defined, Development gets involved. Either the problem is resolved via a “Patch” release into production or a workaround is defined and communicated to the user community.

Patch releases come in two flavors - immediate updates and maintenance releases. The timeframe for an “Immediate” change is defined by the Product Manager whose has an intimate knowledge of the function and the users. “Immediate” may as soon as possible, especially if the next scheduled maintenance release is a significant time in the future. However, “Immediate” may mean the next scheduled maintenance release, especially if it is so close in time that it is not reasonable to prepare two separate releases within the same time period without jeopardizing the release process.
Figure 4: Urgent Issue Process Flow
5. Development Issue Process

The development process that is identified in the flow below is a generic methodology built on the concept of regular communication meetings and some regular period for milestones. For the purposes of this document, we will assume 30 day milestones. A development team is supposed to produce a working example of the feature or sub-feature for review by the customer every 30 days or so. The customer may be the person who requests the feature or the person or group that defines the requirements for the feature. Feedback from the previous milestone and usually, more features or sub-features are rolled into the contents of the next milestone. Frequent customer feedback and Risk Analysis are the compelling reasons for showing this methodology. The concept of a backlog is used to guide the team on feature priorities in the event that the original milestone is completed in less than 30 days or as a place to put items that do not meet the priority or severity criteria for the current release.

One of the major advantages of this process is that software is tested and reviewed much earlier in the SDLC. Problems not usually found until a complete product is handed off to the test team are found sooner with this process. There is a great incentive in the market today to produce software that has fewer problems and even less problems found by the customer. Design rework and misunderstood or poorly communicated requirements are found earlier in the SDLC with reviews after each milestone. Problems found late in the process are the major contributing sources of wasted time and money in software development efforts.

This particular flow is used for new development and potentially spawns another process called the Test Automation process. Test Automation is an important topic in itself. Test Automation and Risk Analysis mentioned here will be dealt with in separate papers.
Figure 5: Development Issues Process
III. Summary

This paper has been a practical examination of the objects that form a Change Request Management System. SQA uses these guidelines in any engagement. SQA has experience working in a variety of development environments. We have found that an effective Change Request Management System benefits all types of development methodologies in all industries. This paper has presented specific workflows to represent the use of formal Change Request Management and industry tools.

While this paper is generic in form, it represents SQA’s philosophy of customizing quality principles for our customers. In software development organizations, knowing how to triage, prioritize, organize and manage change allows software development to stay on course.

A successful implementation of CRM means reduced costs, well managed requirements and achievable development schedules and higher customer satisfaction.

It’s all about Actions, Fields, Forms and Workflows.
Bibliography

A. James Bach,

B. Carnegie Mellon University, Software Engineering Institute (SEI)

C. Jeff Hiatt,
Author Employee’s Survival Guide to Change at change-management.com

D. Tim Creasey,
Editor for the Change Management Learning Center at change-management.com