Position Paper: Keys to Effective Change Management

Executive Overview

The Institute of Internal Auditors, in its guide to Section 404 of the Sarbanes-Oxley Act, states that IT general controls *"provide assurance that applications are developed and subsequently maintained, such that they provide the functionality required to process transactions and provide automated controls. They also assure the proper operation of the applications and the protection of both data and programs from unauthorized change."*[1]

Change management for the mainframe systems management area presents challenges that are not present in the applications programming area. System support personnel must make changes to a wide variety of products using the many different tools prescribed by the vendors of the products. The scope of changes ranges from day-to-day systems management to the deployment of complete updated systems.

Inability to "actively" track and manage all changes to sensitive system resources opens the possibility of introducing unintended, incorrect or malicious changes. Such changes could lead to the loss of a critical subsystem or the entire production environment. Thus, it is vital to track and manage changes to sensitive resources as they happen. Sensitive resources include any resources that, if incorrectly altered, can lead to a disruption, loss or an outage of service or an application. With the wide variety of tools used in the systems management area, assurance of correct changes cannot be guaranteed unless all of the sensitive resources are actively tracked, controlled and backed up automatically without exception.

Environment for Systems Changes

In the current business environment, where 24/7 availability is expected and IPLs must be avoided for increasingly lengthy periods of time, it is no longer feasible for all system changes to be introduced via IPL. IBM and independent software vendors (ISVs) are constantly introducing new tools and methods that expand upon the types of change that can be made and activated dynamically. However, it is still necessary for certain changes, such as an upgraded operating system release or major system-level maintenance, to be staged and then subsequently introduced via IPL.

Systems programmers must be able to make dynamic changes to running systems using the wide variety of tools at their disposal. These tools of the trade include SMP/E, ISPF, IEBCOPY, IDCAMS, DFDSS, ISMF, HCD, AMASPZAP, security system commands, operator commands, UNIX shell commands, and many more, including proprietary vendor-supplied tools.

Management must be able to ensure that systems continue to operate properly in the face of dynamic changes and demonstrate to auditors that proper controls are in place. It

is necessary to be able to identify all changes that have been made, and to provide the appropriate controls to assure that no unauthorized or undocumented changes have been introduced to the system.

Systems support staff must be able to do their day-to-day work productively while providing management with the necessary assurance that all of the management, reporting and auditing requirements are being addressed without exception.

Typical change management tools, designed for use by application programmers, are unable to successfully track and manage all the changes made using the broad spectrum of tools that a systems programmer must use. A successful systems change management implementation must be able to track and manage all dynamic changes to the production environment, regardless of the tools used to make the changes. It is important that systems support staff can make these changes when required and equally important that these changes are actively tracked and/or controlled.

Critical System Resources and Change Methods

Maintenance and change implementation processes can vary depending on the resource and type of change. A process must ensure the integrity of data from the initial vendor-supplied distribution through the eventual implementation into the production system.

Types of changes that must be tracked and controlled include the following:

- APF list changes,
- Linklist changes,
- LPA changes,
- REXX and CLIST changes,
- Proclib changes,
- Parmlib and IPLPARM changes,
- System changes made or activated by operator commands or z/OS UNIX shell commands,
- Changes in system libraries (with or without the use of SMP/E),
- Changes to key z/OS UNIX files and permissions in HFS or zFS,
- Addition or removal of SVC routines,
- Activation or deactivation of dynamic intercept routines for system functions,
- WLM changes,
- Catalog changes,
- Data set movement with DFDSS, FDR etc.,
- Changes in system automation procedures,
- ISV products and associated SVCs, exits, hooks, subsystems, load libraries, parameters etc.

It is important to note that, for many of these types of changes, it is possible to either make the change immediately via operator command or to make a change by editing a library member and activating the change later by operator command. For example, an APF list change can be made synchronously using the SETPROG APF operator command; alternatively, the same change can be made by editing a PROGxx member of parmlib and then activating the change later by using the SET PROG=xx operator

command. Comprehensive systems change management must be able to manage both of these scenarios.

Critical Application Resources

A properly maintained and managed system is a prerequisite for stable production application operation. Additionally, the production applications themselves must be managed to ensure that only authorized changes are made to their components. This applies to both vendor-supplied and in-house built applications.

Some applications require modification of system configuration items, such as APIs, exits and system automation. In these cases, changes to the system configuration items must be coordinated with systems support staff. In addition, applications generally have their own resources, such as:

- Load libraries,
- Other executable parts, such as REXX execs or CLISTs,
- Parameter files,
- Procedures and JCL,
- Job scheduling definitions.

Should any of these processes fail or have problems, then the organization is in danger of non-compliance, adverse audit findings, and receiving bad publicity. For corporations, a failure can also mean losing customers and stock valuation.

Availability of Experienced Staff

Reliable change implementation is often dependent upon aging mainframe systems support personnel that are on the verge of retirement. The retiring staff will take with them their experience, judgment, knowledge, and in-depth familiarity with the systems. Newer and younger replacement staff lack this extensive experience that helps to avoid change implementation errors.

The inevitable staff turnover will make it more vital than ever that an automated process be in place for safety and system integrity. Processes should be able to track every step, change and event from start to finish of any product or maintenance install, as well as application changes. Tracking and documentation should be sufficiently detailed to permit it to be used by a new person to guide the installation of subsequent releases.

Key: Automated Change Management

Most installations adhere to some form of “best practices” when implementing changes. However, the term “best practices” is applied to a wide variety of methodologies, many of which actually carry significant exposures. Frameworks such as ITIL and COBIT prescribe IT governance and management methodologies, but they do not provide any concrete tools that ensure compliance with these methodologies.

In particular, paper-based processes, and their electronic equivalents, gather approvals based on the proposed changes documented in a change request. However, such processes and products lack the system interfaces to ensure that all requested changes are completed and no unauthorized changes are made. As a result, approvals are based on what a technician says he or she will do as opposed to what is actually done. Such processes do not enforce compliance and security, and do not ensure that erroneous or unauthorized changes will not occur.

An effective change management system will automatically prevent unauthorized changes, track and back up authorized changes, and provide for coordinated back-out of

unsuccessful changes. Only an automated system with real-time change interception capabilities can ensure that only the changes that have been authorized are actually made. Similarly, only an automated full-spectrum process with comprehensive tracking and automated backups of changes can provide full documentation of the changes made, as well as a fast-track reversal of changes and quick business continuation in the event of a problem.

> "Employees who feel they cannot rely on a company's technology may use manual processes to compensate for IT weaknesses. Not only are such manual processes labor-intensive and inefficient, but they are inherently riskier than automated processes due to irreducible human error."[2]
>
> Deloitte
> *Sarbanes-Oxley Section 404: 10 Threats to Compliance*

Key: Independence from Specific Change Tools

A successful change authorization and control process must ensure that change implementation rules are followed without exception and without loopholes. It must prevent any change that has not been authorized.

A change management facility that can only control changes made using a specific tool opens loopholes and exposes the entire process to uncertainty. Systems programmers must use a wide variety of tools and utilities to make their changes. It is vital that the change management facility is able to intercept changes regardless of the tool used to make the changes.

Key: Management of System Commands

Many types of systems level changes can be staged by changing a file and activated later by a system command. In such situations, both the file change and the command used to activate the change must be managed. Change approvers must be able to control both the content of the change (the file change) and the timing of the change (the system command). Accordingly, the change management system must be able to intercept the system command as well as the file change.

Key: Comprehensive Change Tracking

In today's dynamic systems environment, successive changes can be made in a very short period of time. Timely problem determination and resolution requires the ability to identify every change made as well as when and how it was made. Furthermore, the level of assurance of reliable system operation required by auditors can require that every change be recorded and associated with an approved change request.

Given the wide variety of tools that systems programmers must use to implement changes, it is not possible to capture all changes or produce a proper audit trail if the change tracking facility can only track changes made using a specific tool. A change tracking process for the systems programming environment must be able to track every change regardless of how the change has been made. It must also be able to associate each change with the associated approved change request.

Key: Automatic Change Backups

Daily or weekly backup jobs are not sufficient to provide comprehensive protection in a modern dynamic systems environment. Multiple changes can be made in a very short

period of time, and it must be possible to restore to any prior change level. This capability requires that a tracked resource must be backed up immediately each time it is changed. Backups also enable comparison of different change levels to identify specific changes within the resource.

Key: Coordinated Change Back-out

Inevitably, some changes either do not achieve the desired results or cause unanticipated problems. In such cases, it is often necessary to back out the change to restore the system to a prior operational state.

A proper back-out of a change requires that the set of resources associated with the change can be identified and that backups of the resources before the change are available. The use of change tracking to associate changes with change requests provides the ability to identify the set of associated resources that must be restored. Automated change backups allow for quick restoration of the prior version of the resources that were changed.

Key: Ease of Use

An ever-increasing rate of change, coupled with staffing constraints, dictates that a change management process must minimize the demands it makes on the time of the people charged with implementing changes. A process that is difficult or time-consuming to use inevitably leads to attempts to bypass the process.

To achieve maximum productivity, the process must automate as much of its operation as possible. It must also make it easy for change requesters to specify the changes to be made with a minimum of data entry. Change approvers must be able to quickly determine what they are approving. Problem solvers must be able to identify in a minimal amount of time what changes have been made to the system, and must have quick access to the change backups that allow an unsuccessful change to be backed out.

Key: Self-Auditing Change Management System

The change management system must internally audit and report on changes made to its definitions that could affect its functionality. Changes to the definitions could signal an attempt to manipulate the system for malicious purposes, and must be rigorously investigated.

Key: Compliance

In recent years, many binding laws and guidelines have been passed that include prohibitive fines, and in some cases even incarceration, for non-compliance.

The laws include:

- 2002 SOX – Sarbanes Oxley
- 2010 SOC-2 - Service Organization Control Type 2
- 2018 GDPR – General Data Protection Regulation
- 2019 COBIT 5 - Control Objectives for Information and Related Technology
- 2022 ISO 27001
- 2022 SOX Revised
- 2025 DORA – Digital Operational Resilience Act
- 2025 CORA – Corporate Accountability Act

As security threats continue to evolve, inevitably more regulations and standards will be introduced to compel organizations to maintain robust controls. In such an environment, it is vital for organizations to ensure that they continue to have a solid compliance structure.

Conclusion

Effective systems management requires that all system changes can be identified, and can be backed out if necessary. To ensure system reliability and audit compliance, changes must be controlled so that unintended and unauthorized changes are not introduced to the system. Comprehensive change tracking and change control require real-time interception of changes. Staffing challenges dictate that a systems change management tool must be easy to use and make minimal demands on the time of the people that use it. The wide variety of tools that systems programmers must use mandates that change tracking and change control must not depend on the use of a specific tool or method to make changes and must be able to support repeated dynamic changes.

About Action Software International

Action Software International is a division of Mazda Computer Corporation.

Located in Toronto, Canada, Mazda Computer Corporation has been producing superior systems and network management software since 1980. The Company's products are widely deployed within Global 2000 companies, as well as numerous government and institutional sites.

Mazda Computer Corporation's mission is to provide easy to use high performance systems management solutions to the IBM z/OS system user community, based on highly functional products and exceptional customer service.

Visit www.actionsoftware.com for more information.

References

[1] The Institute of Internal Auditors, "SARBANES-OXLEY SECTION 404: A Guide for Management by Internal Controls Practitioner." Last modified 2008. http://www.theiia.org/download.cfm?file=31866.

[2] Deloitte Development LLC, "Sarbanes-Oxley Section 404: 10 Threats to Compliance." Last modified 2004. http://www.deloitte.com/assets/Dcom-Shared Assets/Documents/us_assur_TenThreatsSep2004.pdf

Action Software International
20 Valleywood Drive, Suite 107
Markham, Ontario L3R 6G1
Canada

Tel: (905) 470-7113
Fax: (905) 470-6507
https://www.actionsoftware.com/