2009

INTERNAL CONTROLS SCRIPTING LANGUAGE (ICSL): GRUNDZÜGE EINER INSTRUKTIONSSPRACHE FÜR INTERNE KONTROLLEN

Tom Beiler
Strat Hollis

Markus Böhm
PricewaterhouseCoopers, Germany

INTERNAL CONTROLS SCRIPTING LANGUAGE (ICSL): GRUNDZÜGE EINER INSTRUKTIONSSPRACHE FÜR INTERNE KONTROLLEN

Tom Beiler[1], Markus Böhm[2]

Kurzfassung
Compliance von Unternehmen, also die Erfüllung gesetzlicher und regulatorischer Vorgaben, nimmt in ihrer Bedeutung stetig zu, und damit steigt auch die Bedeutung des Internen Kontrollsystems. Problematisch dabei ist, dass die Vorgaben für die Gestaltung des Internen Kontrollsystems Interpretationsspielräume zulassen, die zu unerwünschten Effizienzverlusten bei der Umsetzung der Vorgaben führen können. Dieser Beitrag stellt in diesem Zusammenhang die Grundzüge einer formalisierten Sprache vor, mit der sich interne Kontrollen präzise beschreiben lassen, so dass Angemessenheit und Wirksamkeit von Kontrollen bei der Gestaltung und Erweiterung des Internen Kontrollsystems besser beurteilt werden können.

1. Einführung

Das Interne Kontrollsystem (IKS) ist ein zentrales Element in Unternehmen, um die Wirtschaftlichkeit und Wirksamkeit der Geschäftstätigkeit zu sichern [3]. Neben dem Schutz des Vermögens der Eigentümer oder Anteilseigner dient das IKS ebenfalls dazu, die Ordnungsmäßigkeit und Verlässlichkeit der Rechnungslegung sowie die Einhaltung von gesetzlichen und regulatorischen Vorgaben sicherzustellen. Diese Aufgabe der Einhaltung von Gesetzen und regulatorischen Vorgaben sowie der intern definierten Regelungen und Richtlinien führt in den Bereich der *Compliance* [9]. Compliance, also das Verhalten in Konformität mit bestimmten Regeln sowie der explizite Nachweis dieser Konformität gegenüber Dritten, hat in den letzten Jahren deutlich an Bedeutung gewonnen. Auch ist die Zahl der Vorgaben, die ein Unternehmen je nach Branche und Gesellschaftsform beachten muss, stark gestiegen, und eine Umkehrung dieser Tendenz ist nicht erkennbar. So gibt es neben dem bekannten Beispiel des Sarbanes-Oxley Act [11, 12] eine Vielzahl weiterer Vorgaben, die heutige Unternehmen gegebenenfalls erfüllen müssen. Eine recht vollständige Übersicht gibt [16], unter anderem zu:

- Basel II
- European 8th Directive
- Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG)
- Health Insurance Portability and Accountability Act (HIPAA)

[1] Strat Hollis, Germany
[2] PricewaterhouseCoopers, Germany

- Payment Card Industry Data Security Standard (PCI-DSS)

Weiterhin ist zu beobachten, dass zunehmend auch industriegetriebene Vorgaben für Unternehmen verbindlich werden, beispielsweise des PCI Data Security Standard [13]. Hier fordern nicht der Gesetzgeber oder eine Regulierungsbehörde, sondern namhafte Vertreter der Kreditkartenindustrie die Einhaltung, begründet über ihre Stellung im Markt.

Zur Schaffung und Erhaltung der Compliance leistet das IKS eines Unternehmens einen entscheidenden Beitrag, und mit der Zunahme der Bedeutung von Compliance steigt auch die Bedeutung von interner Kontrolle, denn zusätzliche Compliance-Vorgaben stellen in der Regel auch neue und erweiterte Anforderungen an das IKS eines Unternehmens. Hierzu werden entsprechend der Zielsetzung der gegebenen Compliance-Aufgabe typischerweise bestimmte Kataloge mit Anforderungen an Unternehmensprozesse und deren Kontrolle assoziiert bzw. Ausschnitte von diesen als geeignet identifiziert. Diese Kataloge bestehen im Wesentlichen aus einer hierarchisch gegliederten Liste von Kontrollzielen (Control Objectives), zusammen mit Beschreibungen zugehöriger Kontrollmechanismen (Control Practices). Diese Mechanismen sollen, bei korrekter Umsetzung und Anwendung, sicherstellen, dass die Kontrollziele erreicht werden. Für den Bereich der Informationstechnologie eines Unternehmens sind [5, 6, 7, 8, 13, 14] solche Kataloge; in Abbildung 1 sind zur Veranschaulichung zwei Beispiele hieraus angegeben.

Für Unternehmen besteht somit im Zuge der Compliance-Herstellung die Aufgabe, ihr IKS entsprechend dieser neuen Anforderungen zu erweitern. Das bedeutet, dass die Kontrollziele und Mechanismen im spezifischen Unternehmenskontext richtig interpretiert, angemessen gestaltet und damit wirksam implementiert werden müssen. In der Praxis ist diese Aufgabe allerdings keineswegs trivial. So liegen im Fall des Sarbanes-Oxley Act Untersuchungen vor, die den Schluss nahe legen, dass die Bewältigung dieser Aufgabe oftmals weder reibungslos vonstatten geht noch als effizient wahrgenommen wird. (Einige Zahlen hierzu sind in Abbildung 2 gegeben). Ein bedingender Faktor hier ist, dass die Anforderungskataloge gleichzeitig für viele Unternehmen Gültigkeit haben sollen und daher zwangsläufig eher allgemein und nur wenig spezifisch gehalten sind. Damit entstehen aber auch erhebliche Interpretationsspielräume bei der Implementierung innerhalb einer

SOX Control 8: Ensure Systems Security, 8.8: Periodic Review of Access Rights. [8]	
Illustrative Control: A control process exists and is followed to periodically review and confirm access rights.	*Illustrative Test of Control*: Inquire whether access controls for financial reporting systems and subsystems are reviewed by management on a periodic basis. Assess the adequacy of how exceptions are reexamined, and if the follow-up occurs in a timely manner.

PCI-DSS Requirement 1: Install and Maintain a Firewall Configuration to Protect Cardholder Data. [13, 14]	
Requirement 1.2: Build a firewall configuration that denies all traffic from "un-trusted" networks and hosts, except for protocols necessary for the cardholder data environment.	*Recommended Audit Procedure*: Select a sample of firewalls/routers 1) between the Internet and the DMZ and 2) between the DMZ and the internal network. The sample should include the choke router at the Internet, the DMZ router and firewall, the DMZ cardholder segment, the perimeter router, and the internal cardholder network segment. Examine firewall and router configurations to verify that inbound and outbound traffic is limited to only protocols that are necessary for the cardholder data environment.

Abbildung 1: Beispiele aus zwei Anforderungskatalogen

konkreten Unternehmensrealität. Das wiederum birgt ein erhebliches Risiko für Defizite in interner Kontrolle von unter Umständen signifikantem Ausmaß, welche im ungünstigsten Fall erst im Rahmen einer Prüfung entdeckt werden. Die entsprechende Folge wäre dann die Feststellung der Non-Compliance, also die Verfehlung des eigentlichen Ziels.

Prüfungen und Kontrollen sind keineswegs eine neue Unternehmenspflicht; Unternehmen besitzen im Gegenteil bereits einen mehr oder weniger reichhaltigen Fundus an Kontrollen und Prüfprozeduren. Ebenso werden in Fällen größerer Compliance & Control Projekte externe Spezialisten und Dienstleister hinzugezogen, die ihr Know How in diesem Bereich einbringen. Dennoch, obwohl viele Unternehmen an dieser Stelle gleichartige Probleme erfahren, und das Wissen zu korrekten Implementierungen grundsätzlich vorhanden ist, hat sich bisher keine allgemein verfügbare Interpretationsmethodik über Unternehmensgrenzen hinweg etabliert. Die gemachten methodischen Erfahrungen werden nicht oder nur kaum mitgeteilt und somit auch nicht synergetisch im Sinne einer erweiterten Best Practice der Interpretation aufbereitet. Dabei hätte eine solche Methodik Vorteile: Im Zuge einer konkreten Implementierung könnten Alternativen für präzisierte Kontrollen von anerkannter Qualität verglichen und in Bausatzart ausgewählt werden oder mindestens als Startpunkt einer Anpassung dienen; eine Prüfung des Designs und eventuell auch der operationalen Effektivität würde vereinfacht. Aus Unternehmenssicht wäre es wünschenswert, wenn sich mit einem solchen Hilfsmittel die Effizienz bei Entwicklung und Betrieb des IKS verbessern ließe.

Eine Voraussetzung zu einer Interpretationsmethodik ist die Möglichkeit der formalen Beschreibung von Kontrollen und Kontrollprozeduren. Von einem solchen Formalismus ist zu fordern, dass zum einen die durchzuführenden Kontrollaktivitäten möglichst genau beschrieben werden können. Dabei muss eine möglichst genaue Orientierung an den Vorgaben unterstützt werden, um das Risiko einer fehlerhaften Interpretation und Durchführung zu minimieren. Zum anderen soll ein solcher Formalismus ausreichend Flexibilität bieten, so dass Kontrollen in jedem Schritt der Präzisierung, beginnend bei allgemeinen Anforderungen und dann mit der entsprechend erforderlichen Expertise, durchführbar sind. So wird der Prozess der Interpretation der Vorgaben im Unternehmenskontext

---

Der Sarbanes-Oxley Act ist eine Compliance-Vorgabe, die für an U.S.-Börsen gelistete Unternehmen zu erfüllen ist. Motiviert unter anderem durch den Enron und WorldCom-Skandal werden CFO und CEO verpflichtet, die Korrektheit der Finanzberichterstattung zu bestätigen, indem sie die korrekte Funktion der internen Kontrollen persönlich zertifizieren. Kann eine korrekte Funktion aufgrund signifikanter Schwachstellen in den internen Kontrollen nicht festgestellt, und somit eine Fehlerhaftigkeit der Finanzberichterstattung nicht ausgeschlossen werden, so ist diese Tatsache mit den Finanzberichten der Börsenöffentlichkeit und Investoren gegenüber zu bekunden.

Die Organisation 'Financial Executives International' (FEI) hat bei einer Umfrage für das Berichtsjahr 2006 festgehalten [4], dass nur 22% von befragten Repräsentanten betroffener Unternehmen den Nutzen als die Kosten überwiegend einschätzten, während 78% befanden, die Kosten würden den Nutzen übersteigen. Andererseits jedoch zeigt ein Report von Lord & Benoit [10], dass Unternehmen, die im Hinblick auf ihre Finanzberichterstattung keine Schwachstellen des Internen Kontrollsystems offenlegen mussten, dafür auch vom Aktienmarkt „deutlich belohnt“ wurden. Diese Beobachtungen lassen zusammengenommen den Schluss zu, dass die unternehmensinterne Realität der Sarbanes-Oxley Compliance negativer wahrgenommen wird als das tatsächliche Ergebnis im Aktienwert vermuten ließe. Ein möglicher Faktor für dieses Missverhältnis zwischen Wahrnehmung und Ergebnis sind die Reibungsverluste durch die Maßnahmen zur Schaffung und Erhaltung der Compliance.

Der FEI-Bericht gibt auch die Größenordnung der Aufwendungen für Sarbanes-Oxley Compliance für das Berichtsjahr 2006 an. Dem Bericht zufolge werden durchschnittlich ca. 18.000 Personenstunden angegeben, und die Kosten, für eine Stichprobe von 200 Unternehmen, belaufen sich im Durchschnitt auf 2,9 Mio. Dollar. Bei diesen Zahlen ist zu beachten, dass sie sich auf Sarbanes-Oxley Vorgaben als ganzes beziehen. Rein auf das IKS bezogen liegen solche Untersuchungen leider nicht vor.

Abbildung 2: Einige Daten zur Praxis des Sarbanes-Oxley Act

nicht durch unangemessen strikte Formerfordernisse behindert, sondern stattdessen sogar unterstützt. Mit der Instruktionssprache für Interne Kontrollen ICSL stellen wir in diesem Beitrag einen solchen Formalismus vor.

1.1. Diskussion verwandter Ansätze und alternativer Kandidaten

Am Markt für Unternehmensdienstleistungen und -produkte werden Unternehmen mit Compliance-Aufgaben durchaus unterstützt. Es gibt, nicht zuletzt motiviert durch den Sarbanes-Oxley Act, eine Reihe von Produkten, die Unternehmen bei der Bewältigung von Compliance-Aufgaben im Bereich der Dokumentation unterstützen können [2]. Diese Produkte sind oftmals jedoch nur wenig mehr als eine Adaption von Dokumenten-Management; Dokumenteninhalte mit den eigentlichen Beschreibungen von Kontrollen werden hingegen meist nicht mehr innerhalb des Produkts detailliert abgebildet. Eine Übersicht hierzu liefert [17]. Zur Lösung der eingangs geschilderten Probleme sind diese Produkte in dieser Form daher nur bedingt geeignet, und aus diesen Produkten sind bisher auch keine Ansätze vergleichbar mit ICSL hervorgegangen, oder sind zumindest unzugänglich geblieben.

Eine weitere Klasse von Werkzeugen verfolgt den Zweck, mittels Programmskripten automatisch Daten aus relevanten Systemen zu extrahieren, zu konsolidieren, und zu analysieren. Ein bekannter Vertreter ist hier die 'Audit Command Language' ACL, von den Alternativen ist Picalo ein freies Werkzeug im Sinne von Open Source [1, 15]. Diese Werkzeuge werden für Kontrollen eingesetzt: routinemäßig im Rahmen von IKS sowie ad hoc, und auch zur Aufdeckung und forensischen Analyse von Betrug in Unternehmen. Des Weiteren gibt es spezialisierte Werkzeuge zur automatischen Überwachung von Hard- und Software-Komponenten, insbesondere um die Konfiguration dieser Komponenten zu erfassen, mit den Soll-Einstellungen zu vergleichen und gegebenenfalls eine Korrektur einzuleiten. Beispiele hierfür sind die Einstellungen von IT-Komponenten zur minimalen Passwortqualität, und die automatische Überprüfung von Firewall-Einstellungen. Diese Werkzeuge sind jedoch nah an IT-Komponenten orientiert, und daher kaum allgemein verwendbar. Davon abgesehen ist die Zielsetzung dieser Werkzeuge eine Automatisierung, und so können in beiden Fällen diese Sprachen nicht die oberste Ebene der persönlichen, manuellen Kontrolle durch die tatsächlich Verantwortlichen innerhalb des Unternehmens vollständig ersetzen, denn sonst würde das Verantwortlichkeitsprinzip verletzt.

Auch wenn eine Automatisierung, wo sie möglich ist, grundsätzlich Vorteile und Unterstützung bietet, sollte dennoch Spielraum für fließende Übergänge von manuellen zu automatisch angereicherten Kontrollen vorhanden sein. Dies ist vor allem dort ein Vorteil, wo eine situative Reaktion auf eine variierende Kontrollumgebung und menschliches Urteilsvermögen gefragt sind. In derartigen Situationen erweisen sich automatische Lösungen als zu statisch und unflexibel. Zudem kann der Betrieb solcher Werkzeuge für Unternehmen einen hohen Kostenfaktor darstellen. Als Konsequenz sei festgehalten, dass eine Instruktionssprache ICSL mit automatischen Kontrollen sinnvoll kooperieren können muss.

2. Die Grundzüge von ICSL

ICSL basiert auf der Beobachtung, dass die einschlägigen Kontrollanforderungskataloge einer Normsprache bereits recht nahe sind. Es treten häufig dieselben Handlungsanweisungen auf, angewendet auf verschiedene Objekte. Folgerichtig besteht ICSL im Kern aus einer Formalisierung der typischen, verwendeten Handlungsanweisungen wie zum Beispiel 'verify', 'identify' und 'obtain' mit einer Festlegung der zugehörigen Objekte. Eine exemplarische Auswahl von Grundinstruktionen ist in Abbildung 3 dargestellt. Hinzu kommen Elemente zur Strukturierung von Prozeduren, nament-

lich zur Gruppierung und Blockbildung, zur bedingten Ausführung, und zur Iteration. Darüber hinaus enthält ICSL die Möglichkeit, untergeordnete, externe und automatische Skripte zu initiieren und auszuwerten. Abbildung 4 zeigt die Strukturierungselemente von ICSL.

Ein weiteres Element von ICSL ist die Behandlung von Instruktionsobjekten als Variablen. Im Gegensatz zu Kontrollbeschreibungen in gebrauchssprachlicher Form werden in ICSL die verwendeten Objekte, insbesondere die in die Kontrolle involvierten Rollen und Personen und die erhobenen und gesammelten Nachweise (beispielsweise Listen, Spreadsheets, Dokumente, E-Mails, digitalisierte Kopien) eineindeutig benannt und referenziert.

In der ICSL-Notation werden die Schlüsselwörter einer Instruktion in Fettschrift und literale Satzteile sowie Variablen in normaler Schriftstärke geschrieben. Variablen haben zudem zur Kennzeichnung ein Dollarzeichen '$' vorangestellt, und wenn sie mit Inhalten belegt sind, werden sie, ähnlich einem Hyperlink, unterstrichen. Es gibt keinen formalen Unterschied zwischen skalaren Daten und Datenlisten. Im Zweifelsfall ist ein Skalar eine Liste mit nur einem Element. Falls zur Verdeutlichung gewünscht, können Variablen, die Listen repräsentieren, in eckige Klammern und weiterhin mit führendem Dollarzeichen zur Variablenkennzeichnung gesetzt werden. Variablennamen, die aus mehreren Wörtern bestehen, können zur Vermeidung von Missverständnissen in runde Klammern gesetzt werden.

Ein ICSL-Skript muss stets mindestens eine 'Conclude'-Instruktion enthalten. Damit wird festgeschrieben, dass die Durchführung einer Kontrolle immer ein Ergebnis 'Pass' oder 'Fail' haben muss, das heißt, die Feststellung eines Sachverhaltes muss auch zu einer Bewertung führen. Eine Ausnahme ist eine alleinstehende 'Verify'-Instruktion: Dies ist der Anfang einer Refinement-Kette, und wir gehen davon aus, dass ein 'Conclude' in diesem Fall implizit enthalten ist.

Verify that $Item(s) **is/are** $Prädikat**.**

Verify ist eine abstrakte Instruktion die feststellt, ob *Prädikat* für *Item* bzw. für alle *Items* wahr ist. Ohne Refinement lässt sie es frei, wie diese Prüfung angemessen durchzuführen ist. Verify bildet sinnvoller weise den Anfang einer Refinement-Kette.

Obtain $Item **from** $Rolle/Person**.**

'Obtain' ist eine Eingabe-Instruktion: Es soll eine Sache oder Information, beschrieben mit *Item,* von einer Person *Person*, oder von einer Person in der angegebenen Rolle *Rolle* erhalten werden. Falls eine Rolle spezifiziert war, steht im weiteren Verlauf der Name der tatsächlichen Person in dieser Variablen.

Identify Element(e) **in** $Item(s) **where** $Prädikat **to** $Ergebnis**.**

'Identify' liefert diejenigen *Elemente* aus *Item(s)*, für die das Prädikat *Prädikat* erfüllt ist. Das Ergebnis bzw. die Liste mit den identifizierten Elementen befindet sich anschließend in *Ergebnis*.

Compare $Item-1 **and** $Item-2 **[on** $Komparator**] to** $Ergebnis**.**

'Compare' vergleicht zwei Dinge *Item-1* und *Item-2* und beschreibt als Ergebnis die festgestellten Unterschiede in *Result*. Optional kann mit *Comparator* eine Vergleichsmethode angegeben werden.

Validate $Item **with** $Check-Liste**.**

'Validate' überprüft *Item* mithilfe der gegebenen Prüfliste *Check-Liste*. Das Ergebnis dieser Validierung steht im weiteren Ablauf in *Check-Liste* zur Verfügung.

Conclude to Pass **if** Bedingung, **or to** Fail **otherwise.**

'Conclude' stellt das Kontrollergebnis 'Pass' oder 'Fail' fest. Hierzu werden die vorherigen Tests in *Bedingung* ausgewertet und mit einem Referenz- oder Schwellwert verglichen.

Assess $Sachverhalt **[with** $Methode**] to** $Assessment**.**

'Assess' liefert eine Einschätzung des gegebenen *Sachverhalts*. Meist ist dies das Kontrollresultat, oder ein Zwischenergebnis. Das Ergebnis steht in *Assessment*. Optional kann mit *Methode* eine Methode, Vorlage oder andere Arbeitshilfe angegeben werden.

Report Ergebnis **with** $Vorlage**.**

'Report' beschreibt das Kontrollergebnis, und möglicherweise weitere Details z.B. statistischer Art, in Form und mithilfe von *Vorlage*. Der Report ist anschließend in *Vorlage* enthalten.

Abbildung 3: Eine Auswahl von ICSL Grundinstruktionen

Block / Gruppierung: {*Instruktionsfolge***}**

Folgen von Instruktionen werden in einem Block gruppiert, um sie gemeinsam als eine strukturelle Einheit handhaben zu können. Dies wird von den weiteren Strukturelementen von ICSL benötigt. Eine solche zusammengehörige Instruktionssequenz wird in geschweifte Klammern gesetzt. Es ist außerdem guter Stil, die Instruktionen einzurücken. Eine einzelne Instruktion bildet bereits eine Folge; dann können die geschweiften Klammern weggelassen werden.

Bedingte Ausführung: if *Bedingung* **then** *Block* **else** *Block*.

ICSL enthält die typische Bedingte Ausführung in der typischen Schreibweise if-then-else.

Iteration: For each $*Item* **in** $*Liste* **[do]** *Block***.**

Mit diesem Operator wird über die Liste *Liste* iteriert, und in jeder Iteration steht in Block das aktuell iterierte Listenelement *Item* zur Verfügung. Die Liste wird Element für Element 'ausgerollt', und der Block auf jedem Element einmal durchgeführt.

Refinement: *Instruktion* **by** *Block***.**

Der Refinement Operator 'By' kann am Ende bzw. anderer geeigneter Stelle einer Grundinstruktion stehen. Der zugehörige Block beschreibt dann genauer, wie die Grundinstruktion auszuführen ist. Die häufigste Grundinstruktion, die so präzisiert wird, ist die 'Verify'-Instruktion.

Aufruf Externer Skripte: Perform *Skript***.**

Mit der Instruktion 'Perform' werden andere Skripte zur Ausführung gebracht. Solche Skripte können wiederum ICSL-Skripte sein, oder auch beliebige andere Skripte, wie z.B. automatische Skripte in ACL.

Abbildung 4: Die Strukturierungselemente von ICSL

Die Reihenfolge eines Skripts kann aufgebrochen werden, wenn zum Beispiel erst während der testenden Phase ein weiterer Eingabebedarf auftritt und ein weiteres bzw. wiederholtes 'Obtain' oder 'Observe' erforderlich wird. Ein solcher Fall tritt ein, wenn erst im Testverlauf auffällt, dass die Eingaben nicht vollständig waren. Ebenso die Reihenfolge betreffend ist die Verfahrensweise, wenn das Skript nicht sinnvoll weitergeführt werden kann, weil etwa auf ein 'Obtain' keine verwertbare Eingabe trotz angemessenen Aufwands erfolgt ist. Dann liegt es im Ermessen des Ausführenden, an geeigneter Stelle wieder aufzusetzen. In der Regel ist diese Stelle dann das 'Conclude' mit einem Ergebnis 'Fail' inklusive geeigneter Begründung.

Weitere Konstrukte von ICSL sind das Refinement zur Unterstützung verschiedener Grade an Präzisierung sowie die Historisierung von tatsächlich durchgeführten Kontrollen. Beide Konstrukte werden im Folgenden detaillierter dargestellt.

2.1. Refinement

ICSL enthält ein Konstrukt zum Refinement. Mit Refinement wird in der Spezifikationstheorie die Konkretisierung einer allgemeineren Spezifikation in eine näher am Zielsystem orientierte Spezifikation bezeichnet. Aus den Freiheitsgraden, die eine höhere Spezifikation bietet, werden konkrete Möglichkeiten im Hinblick auf das Zielsystem gewählt und festgelegt. ICSL enthält dieses Konstrukt, um den Vorgang der Interpretation von einer allgemeinen Kontrollvorgabe, bei der vergleichsweise viel Expertenwissen zur Durchführung benötigt wird, zu einer im Unternehmenskontext konkreteren Anpassung mit weniger Freiheitsgraden, und damit mit weniger Fehlermöglichkeiten zu begleiten. Wir demonstrieren den Refinement-Vorgang anhand von SOX-Cobit 8.8 aus Abbildung 1. Die gegebene Kontrolle lässt sich umschreiben in eine Verify-Instruktion:

A control process exists and is followed to periodically review and confirm access rights.

▼ *Transformation nach ICSL*

Verify that all access rights **are** periodically confirmed.

Bei der Transformation nach ICSL ist der Term 'a control process exists and is followed' wegen Redundanz weggefallen. Des weiteren fließt bei der Transformation das Expertenwissen mit ein,

dass es hier insbesondere darum geht, dass für alle Benutzer relevanter Systeme die Zugriffsrechte regelmäßig bestätigt werden sollen, um Zugriffsrechte, für die keine Autorisierung (mehr) vorliegt, zu entdecken und in der Folge zu bereinigen. Es ist hier übrigens auch zu sehen, dass bereits ein verhältnismäßig einfacher und klarer Satz nicht unbedingt einfach in der Interpretation ist.

Unter Berücksichtigung dieser Hinweise zur Interpretation kann jetzt ein Refinement stattfinden:

Verify that all access rights are periodically confirmed.

▼ *Refinement innerhalb ICSL*

Verify that all access rights are periodically confirmed **by {**
Obtain $List of all users **from** Admin**.**
Sample $List of all users **to** $[Sample]**.**
For each $User **in** $[Sample] **do** {
Obtain $Current Access Rights **from** Access Admin**.**
Obtain $Last Confirmation **from** Access Admin**.**
Identify any Access Right(s) **in** $Current Access Rights **where** that Access Right has:
- neither been confirmed in $Last Confirmation,
- nor has been granted since the last confirmation,
- nor is currently in the process of being revoked,

appending to $Exceptions**.**
}**.**
Conclude on Pass if $Exceptions is **empty, and on** Fail **otherwise**.
}.

Die im ‚illustrative Test of Control' zu SOX-Control 8.8 geforderte Einschätzung der Angemessenheit der Zeitlichkeiten des Aufräumens kann als weitere Kontrolle modelliert werden. Diese Kontrolle würde die Vorgaben für die Zeitlichkeiten erheben, und diese dann mit einer qualifizierten Einschätzung vergleichen. Ob diese zeitlichen Vorgaben auch tatsächlich eingehalten werden, ist hingegen in obiger Beispielkontrolle noch zu integrieren. Ebenso ist die Sample-Methode genauer zu spezifizieren. An dieser Stelle wurde darauf verzichtet. Auch wird implizit vorausgesetzt, dass nur die Zugriffsrechte für ein gegebenes System untersucht werden. Für mehr als ein einzelnes relevantes System wäre diese Kontrolle von einem übergeordneten Skript aus durchzuführen, dass über die Liste der Systeme iteriert, oder eine Liste der relevanten Systeme müsste direkt im Skript berücksichtigt werden.

Grundsätzlich ist beim Refinement zu beachten, dass die Präzisierung entlang einer Abstraktionsachse verläuft, und nicht etwa entlang einer Vervollständigung. Der Refinement-Operator kann nur dazu dienen, vorhandene Teile einer Kontrolle genauer zu fassen, nicht aber, um nicht vorhandene Teile einzufügen. Das Einfügen neuer Teile verändert die Kontrollsemantik, und muss dann auch dementsprechend behandelt werden.

2.2. Historisierung

ICSL bietet zudem eine Historisierung der Ausführung von Kontrollen. Die Historisierung von Kontrolläufen ist hilfreich zur Dokumentation und zur Erbringung des Nachweises, dass Kontrollen tatsächlich und ordnungsgemäß durchgeführt worden sind. Zur Historisierung werden die Instruktionen des Skripts im Zuge der Ausführung in eine historische Version überführt, bei der sprachlich vom Imperativ in eine Vergangenheitsform übergegangen wird, und die geforderten Variablen dann mit Inhalten belegt sind. Mit diesem Mechanismus geht aus der Kontrollausführung das Ausführungsprotokoll bzw. der Kontrollreport hervor. Abbildung 5 zeigt exemplarisch eine erfolgreiche Ausführung (Ergebnis: Pass), und einen fehlgeschlagenen Lauf (Ergebnis: Fail), der Handlungsbe-

Der erfolgreiche Fall: Die Kontrolle hat keine Abweichungen identifiziert.

The following Control has been performed on Feb. 28th, 2008 **with the Result** Pass:

It has successfully been verified that all access rights **are** periodically confirmed **by** {
$List of all users **has been obtained from** Admin A. Smith.
$List of all users **has been sampled to** $[Sample].
For each $User **in** $[Sample], **the following has been performed**: {
$the Current Access Rights **have been obtained from** Access Admin B.Miller**.**
$the Last Confirmation **has been obtained from** Access Admin B.Miller**.**
Any Access Right **in** $the Current Access Rights **have been identified where** that Access Right has:
— neither been confirmed in $Last Confirmation,
— nor has been granted since the last confirmation,
— nor is currently in the process of being revoked,
appending to $Exceptions.
}.
It has been concluded on Pass, **as** no exceptions, and hence no violations of the re-confirmation process have been identified.

Der fehlgeschlagene Fall: Die Kontrolle hat Abweichungen aufgedeckt.

The following Control has been performed on Jan. 31th, 2008 **with the Result** Fail:

It has not been possible to verify that all access rights are periodically confirmed, **for the following reason:** 2 users have been detected where the access rights have never been confirmed over the three years of the existence of the accounts. This exceeds the expected re-confirmation period of 3 months. The preliminary reason given by Admin A. Smith is that one user is a technical user, and hence is being excluded from the re-confirmation process, while the second user somehow has left the company and forgotten to be deleted from the user base. See $Exceptions for detailed information on the user ids identified.

Abbildung 5: Historische Formen der Beispielkontrolle

darf für zwei Benutzerkonten identifiziert hat. Im Beispiel nicht mit aufgeführt ist, dass im fehlgeschlagenen Fall der historische Kontrollauf wie im erfolgreichen Fall zu Dokumentationszwecken ebenfalls enthalten ist.

2.3. Weitere Elemente von ICSL

ICSL enthält weitere Ansätze, die über die beschriebenen Grundzüge hinaus gehen:

Formulare. Manuelle Kontrollen und ihre Ausführungen werden in der Praxis oftmals durch Formulare und Checklisten abgebildet, und ICSL-Skripte haben durchaus eine Äquivalenz mit Formularen, allerdings sind ICSL-Skripte in der Darstellung kompakter. Bei einer Umsetzung von ICSL-Skripten mithilfe interaktiver Webseiten in einem Intranet werden die Übergänge zu Web-Formularen fließend. Ebenso ist aber für ICSL ein Formulargenerator für klassische Papierformulare denkbar.

Rollen und Qualifikationen. Ein ICSL-Script Instruktionen kann verschiedener Schwierigkeitsgrade enthalten. Diesem Umstand kann Rechnung getragen werden, indem die Instruktionen eines Skriptes mit Rollen annotiert werden, zum Beispiel 'Business Analyst', 'Senior Control Analyst', 'Controller', 'Compliance Manager' (Namen können je nach Umgebung variieren). Hierdurch können die zur Verfügung stehenden Kontrollakteure bezüglich der Qualifikation und Funktion effizienter eingesetzt werden, und es kann der Umstand, dass eine wenig präzise Kontrollbeschreibung eine größere Expertise in der Durchführung erfordert, deutlicher widergespiegelt werden.

Delegation. Die Ausführung einzelner Instruktionen in einem Skript kann delegiert werden. Dies ist sinnvoll zur Arbeitsteilung, im speziellen für Iterationen über Listen, die zum Beispiel mehrere Systeme, Ansprechpartner, Abteilungen oder Orte betreffen. Dieser Umstand ist mit der historischen Form einer Kontrolle aufzuzeichnen. Es ist jedoch immer der oder die ursprüngliche Kontrollausführende verantwortlich für die Korrektheit der Durchführung.

Management-System. Mit zunehmender Größe des IKS ist es sinnvoll, die Kontrollen mit einem passenden Management-System zu verwalten. In Bezug auf ICSL sind für ein solches System folgende Möglichkeiten zur Einbettung wünschenswert:

- Der Entwicklungs- und Reifungsprozess von Kontrollen sollte durch einen angemessenen Lebenszyklus und eine Versionierung unterstützt werden.
- Das System sollte die zeitliche Planung der Kontrolldurchführungen und weitere inhaltliche Abhängigkeiten abbilden und umsetzen können.
- Das System sollte die historischen Kontrollen zusammen mit den erbrachten Nachweisen archivieren. So kann der korrekte Betrieb des IKS schnell und übersichtlich dargestellt werden.
- Das System sollte die Ergebnisse der Kontrollen konsolidieren und für ein Management-Reporting aufbereiten.
- Das System sollte eine Wiederverwendung von Nachweisen und Materialien ermöglichen, wenn dies in bezug auf eine Kontrolle möglich und sinnvoll ist. Dies minimiert Störungen durch wiederholtes Beschaffen von Daten bei sonst an den Kontrollen unbeteiligten Datenlieferanten, und minimiert damit Störungen der eigentlichen Geschäftstätigkeiten und Unternehmensabläufe.

3. Abschließende Bemerkungen

In diesem Beitrag wurden die Grundzüge der Skriptsprache Internal Controls Scripting Language ICSL zur Formalisierung Interner Kontrollen vorgestellt. ICSL behält dabei die gewohnten Elemente typischer Kontrollbeschreibungen bei und belegt diese mit einer definierten Semantik. ICSL bietet Variablen zur eindeutigen Bezeichnung und durchgängigen Verwendung der Instruktionsobjekte, sowie die für Skriptsprachen typischen Strukturierungselemente, inklusive des Aufrufs externer Skripte. Letztere Möglichkeit erlaubt die Integration anderer, bereits vorhandener Kontrollen. Diese können auch automatisch sein.

Der Prozess der Überführung allgemeiner Kontrollanforderungen in die konkrete Realität eines gegebenen Unternehmens wird in seinen Schritten der Anpassung und Verfeinerung durch den Refinement-Operator unterstützt. Dabei erlaubt ICSL eine Balance zwischen der Genauigkeit durch eine hohe Formalisierung einerseits, und den Vorteilen menschlichen Einschätzungs- und Reaktionsvermögens in der Kontrollausführung auch in Anwesenheit von Präzisionsdefiziten in der Spezifikation sowie bei geänderter Kontrollumgebung andererseits.

Durch eine Formalisierung interner Kontrollen schafft ICSL eine Voraussetzung zur genaueren Formulierung und Vergleichbarkeit von Kontrollen auch außerhalb eines einzelnen, sehr spezifischen Unternehmenskontextes. Dies kann die Basis sein für einen allgemein verfügbaren Fundus konkreter, interner Kontrollen, zusammen mit der zugehörigen Interpretationsmethodik. Dies ist eine Vorbedingungen für eine Verbesserung der Hilfsmittel zur Entwicklung und Erweiterung eines IKS im Kontext von Compliance.

4. Literaturangaben

[1] ACL SERVICES LTD., WWW.ACL.COM.

[2] AMBERG, M., MOSSANEN, K.., Vorteile und Herausforderungen IT-gestützter Compliance-Erfüllung, Gemeinschaftliche Studie der Universität Erlangen, Lehrstuhl für Wirtschaftinformatik III, und Novell, Inc., 2007.

[3] COMMITTEE OF THE SPONSORING ORGANIZATIONS OF THE TREADWAY COMMISSION (COSO) (Eds.), Internal Control – Integrated Framework, 1992.

[4] FINANCIAL EXECUTIVES INTERNATIONAL (FEI), FEI Survey: Management Drives Sarbanes-Oxley Compliance Costs Down by 23%, But Auditor Fees Virtually Unchanged, Florham Park, NJ, May 2007.

[5] INTERNATIONAL STANDARDIZATION ORGANIZATION, ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements, 2005.

[6] INTERNATIONAL STANDARDIZATION ORGANIZATION, ISO/IEC 27002: Information technology – Security techniques – Code of practice for information security management, 2005.

[7] IT GOVERNANCE INSTITUTE, CobiT (Control Objectives for Information and Related Technology), Revision 4.1, Rolling Meadows, IL, May 2007.

[8] IT GOVERNANCE INSTITUTE, IT Control Objectives for Sarbanes-Oxley: The Role of IT in the Design and Implementation of Internal Control Over Financial Reporting, 2nd Edition, Rolling Meadows, IL, September 2006.

[9] KLOTZ, M., DORN, D.-W., IT-Compliance – Begriff, Umfang und relevante Regelwerke, in: Praxis der Wirtschaftsinformatik, HMD Heft 263, dpunkt.verlag, Oktober 2008.

[10] LORD, R., The Lord & Benoit Report: Do the Benefits exceed the Costs?, Lord & Benoit, LLC., Worcester, MA, May 2008.

[11] MARCHETTI, A.-M., Beyond Sarbanes Oxley Compliance. Effective Enterprise Risk Management, John Wiley & Sons, New Jersey, 2005.

[12] MENZIES, C., REIMER, B., et al., Sarbanes-Oxley Act. Professionelles Management Interner Kontrollen, Schäffer Poeschel Verlag Stuttgart, und PriceWaterhouseCoopers, 2004.

[13] PCI SECURITY STANDARDS COUNCIL, Payment Card Industry (PCI) Data Security Standard, Version 1.1, Released September 2006.

[14] PCI SECURITY STANDARDS COUNCIL, Payment Card Industry (PCI) Data Security Standard: Security Audit Procedures, Version 1.1, Released September 2006.

[15] PICALO.ORG, Picalo Data Analysis and Fraud Detection Toolkit: Picalo Manual, Version 3, published on www.picalo.org, 2007.

[16] TARANTINO, A., Manager's Guide to Compliance. Best Practices and Case Studies, John Wiley & Sons, New Jersey, 2006.

[17] TRIPATHY, R., Sarbanes-Oxley Tools: Why Do They Fail?, Published on InsideSarbanesOxley.com, 2006.