



Konfigurations-, Rollout-, und Change-Management

Best Practice Tipps für Projekte und Lebenszyklen

Stand 26. Sept 2010, 18:00 | Version 1.0

Thomas vom Braucke

© Ceyoniq Technology GmbH, 2010

Impressum

Alle Rechte vorbehalten. Dieses Dokument und die darin beschriebenen Programme sind urheberrechtlich geschützte Erzeugnisse der Ceyoniq Technology GmbH, Bielefeld, Bundesrepublik Deutschland.

Änderungen vorbehalten.

Die in diesem Dokument enthaltenen Informationen begründen keine Gewährleistungs- und Haftungsübernahme seitens Ceyoniq.

Die teilweise oder vollständige Vervielfältigung ist nur mit schriftlicher Genehmigung der Ceyoniq Technology GmbH zulässig.

Alle verwendeten Handelsmarken gehören den jeweiligen Herstellern/Inhabern.

Copyright © 2010 Ceyoniq Technology GmbH

Ausarbeitung: Ceyoniq Technology GmbH

Ergänzend gilt

"Freigabe zur Veröffentlichung der DokVersion 1.0.pdf"

IMPRESSUM	2
1 ZIELGRUPPE, INHALT	5
2 WICHTIGE AUFGABEN, MEILENSTEINE UND RISIKEN	6
2.1 Anforderungen aus ITIL Change Management	6
Übersicht Change Management	6
2.2 nscale Lebenszyklus	6
2.3 Notwendigkeit von Nachvollziehbarkeit	6
2.3.1 Dokumentation oder selbstdokumentierende Systeme?.....	6
2.3.2 Aspekte, die Nachvollziehbarkeit erfordern und ergeben sollen	7
Entwicklung und Weiterentwicklung von Anpassungen, Customizing.....	7
Compliance	7
Weiteres	7
3 GRUNDLAGEN FÜR [NSCALE] PROJEKTE UND LEBENSZYKLEN	8
3.1 Organisatorische Struktur	8
InformationsKnoten einrichten	8
Anforderung einfache Nutzbarkeit	8
3.2 3-stufige Systemlandschaft	8
3.3 Datensicherung	9
3.4 Protokollierungen, Inventarisierungen, Snapshots	9
Protokolle im System.....	9
Inventarisierung	9
VM-Snapshots und System-Images.....	9
4 NSCALE IM CHANGE-MANAGEMENT-PROZESS	10
4.1 Request for Change	10
4.2 Registrierung und Klassifizierung	10
nscale Aufwandsabschätzung	10
nscale Risiko-Klassifizierung	10
4.3 Überwachung und Planung	10
nscale und anderes - Grundsätzliches bei der Planung	11
nscale Datenrelationen	11
4.4 Genehmigung	11
4.5 Ausarbeitung und Test.....	11
Ausarbeitung der geplanten Änderungen	11
Tests im Q-System	11
4.6 Implementierung	11
4.7 Auswertung.....	11

5 MÖGLICHKEITEN ZUR ÜBERTRAGUNG VON ANPASSUNGEN IN NSCALE13

Terminologie in diesem Bereich.....	13
Rollback, Back-Out-Möglichkeiten, Datensicherung der Konfigurationen und Versionen des Systems	13
Möglichkeiten zur Übertragung von Anpassungen.....	13

6 CHECKLISTEN ZU KLASSIFIZIERUNG, VERFAHREN UND PRÜFUNG .. 14

Methoden und Werkzeuge zur Risikoanalyse.....	14
Matrix Risikobewertung nscale Items.....	14
Abhängigkeiten.....	14
Fazit: Projektspezifische Checklisten und Prüfungen	14

7 BEISPIELE UND VORLAGEN FÜR CHANGES 15

7.1 Vorlagen	15
Arbeitsliste Change Implementierung	15
7.2 Beispiele	15
7.2.1 Beispiel mit Rollen.....	15
Rolle F erhält für Dokumentklasse X das Recht zu archivieren.	15
7.2.2 Beispiel mit Attributen.....	15
Attribut in Verträgen ergänzen.....	15
7.2.3 Beispiel mit DrittSystemanbindungen	15
Posteingangsmanagement Exchange	15
7.2.4 Anpassungsfehler und Patches (,Bugfixes')	15
Klasse XY	15
Maske YZ.....	15

8 WICHTIGE ANLAGEN 16

nscale Konfigurations-, Rollout-, ChangeManagement - Listen, Matrizes.xlsx	16
Sicherung und Wiederherstellung von Daten und Konfiguration - [Verfahrens]Leitfaden für nscale-Systeme.pdf	16

9 QUELLEN..... 17

Notizen zur Entwicklung des Dokuments	17
---	----

10 SUPPORT 18

1 ZIELGRUPPE, INHALT

Das Dokument richtet sich an Berater, Projektplaner und IT-Administratoren.

Es enthält Richtlinien, Tipps und Best Practice Beispiele für das Einspielen und die Verwaltung von Customizing-Änderungen und Softwareupdates in nscale Projekten. Dieses Papier soll Handwerkszeug zur technischen Umsetzung des [ITIL-Change-Managements](#)-Ziels „alle Anpassungen an der IT-Infrastruktur kontrolliert und effizient unter Minimierung von Risiken durchzuführen“ liefern.

Fokus ist, was im nscale zur Verfügung steht um Veränderungen geplant, effizient, kostengünstig und mit minimalem Risiko ausführen zu können. Der organisatorische durch das Change Management zu gestaltende Rahmen von Erfassung, Dokumentation, Genehmigung und Überwachung wird hier jedoch nur peripher betrachtet. Es wird z. B. beschrieben, welche Funktionen des nscale für Dokumentation, Risikoklassifizierung usw. genutzt werden können.

Schwerpunkt der Ausgabe-Versionen 1 des Dokuments sind Analyse, Dokumentations- und Update-Techniken, welche für die Ceyoniq Server-Komponente nscale Application Layer und ferner den pipeliner Verwendung finden können.

2 WICHTIGE AUFGABEN, MEILENSTEINE UND RISIKEN

Eine wesentliche Herausforderung ist über eine gute Balance zwischen Absicherung, hinreichender Dokumentation, Durchführung von Tests usw. ein wirtschaftliches und dauerhaft effizientes IT-System zu betreiben und weiter zu entwickeln. Eine gewohnheitsmäßige Reflektion, dass die IT nicht Selbstzweck sondern vielmehr effizienzsteigerndes Werkzeug für die Prozesse und Geschäftsmodelle sein soll, kann zu einem optimierten Ergebnis führen.

2.1 Anforderungen aus ITIL Change Management

Übersicht Change Management

[Quelle: [Change Management \(ITIL\) - Wikipedia 20100614.pdf](#)]

Das Change-Management wird in der ITIL-Literatur im Buch "Service Transition" als eine eigene Prozessdisziplin beschrieben und gehört zu den Prozessen der Umsetzung von geschäftlichen Anforderungen in die IT-Service-Landschaft. Das Ziel des Change-Managements nach ITIL ist es, alle Anpassungen an der IT-Infrastruktur kontrolliert und effizient unter Minimierung von Risiken durchzuführen.

Ein hoher Anteil von kostenintensiven IT-Service-Störungen lässt sich häufig auf schlecht koordinierte oder unzureichend gesteuerte Veränderungen an der IT-Servicelandschaft zurückverfolgen. Diese Störungen können bei der heutigen Verknüpfung der IT mit den führenden Prozessen eines Unternehmens enorme Kosten nach sich ziehen. Dies rechtfertigt Investitionen in Prozesse, in welchen der Bedarf für einen Change und die möglichen negativen Auswirkungen geprüft und die Störungen durch entsprechende Maßnahmen auf ein akzeptables Minimum reduziert werden.

Es ist die Aufgabe des Change-Managements, sicherzustellen, dass standardisierte Methoden und Verfahren zur Durchführung von Veränderungen existieren und effizient genutzt werden.

Change-Management ist dafür verantwortlich, den Change-Prozess zu erstellen und zu verwalten. Der Prozess beinhaltet die Erfassung, Dokumentation, Genehmigung und Überwachung und stellt sicher, dass Veränderungen geplant, effizient, kostengünstig und mit minimalem Risiko ausgeführt werden.

2.2 nscale Lebenszyklus

Wichtige Meilensteine in einem [Software-Lebenszyklus](#) in der Phase [Implementierung](#) und Nutzung sind:

- erstes Rollout, grundsätzliches Upgrade oder Migration eines Altsystems;
- technisch notwendige Updates (Kapazitätserweiterung, Sicherheit, Auslaufen Support für Teilkomponenten im Bereich Soft- oder Hardware, etc.);
- fachlich gewollte Änderungen der Anpassungen;
- Ausfall, Notfall und sonstige nicht geplante Ereignisse.

2.3 Notwendigkeit von Nachvollziehbarkeit

2.3.1 Dokumentation oder selbstdokumentierende Systeme?

Redundante oder nicht nachgezogene Dokumentationen sollten vermieden werden. Oft kommt es sonst zu dem Problem, dass ein Eintrag in einem Techbook nicht mehr dem Stand des Systems entspricht. Da viele Dinge von den Systemen selbst dokumentiert sind, ist es sinnvoll in den ‚statischen‘ Dokumenten

lediglich auf die Systeme zu verweisen. Nicht jedoch jedes Detail nochmals separat aufzuschreiben.
Wichtig ist hierbei, dass der Soll-Zustand ebenfalls dokumentiert bleibt.

Freiraum für das Wesentliche.

2.3.2 Aspekte, die Nachvollziehbarkeit erfordern und ergeben sollen

Diverse Gesichtspunkte fordern Klarheit und Transparenz. Hinreichend dokumentiert sorgen diese für selbiges.

Entwicklung und Weiterentwicklung von Anpassungen, Customizing

- diesbezügliche [OS](#)
- Systemvergleiche

Compliance

- Die Nachweisbarkeit von Regeleinhaltungen erfordert nachvollziehbare Betriebskonzepte.
- Prozesse und deren Veränderung.

Veränderung von Berechtigungen, Publizierungstechniken und Informationsvernetzungen

- durch [ACL](#)-Änderungen oder veränderte Objektklassen oder veränderte Sichtbarkeitsregeln (ggf. auch historisch / Vergleichbarkeit)
s. a. Bsp. IC Revision-Report mit Vorgängervergleich
- Veränderte Auffindbarkeit von Inhalten
z. B. durch Änderung von SQL-Abfragen respektive Trefferlisten
- und Verknüpfung von Daten
=> Veränderungen von Informationen bzw. ECM-Bereitstellungs-Ergebnissen

Weiteres

- Laufender Betrieb
- Support
- Fehlereingrenzung
- Produkt: Updates / Rollbacks
 - o Basissoftware wie nscale, OS, Datenbank, etc.
- Anpassungen: Updates / Rollbacks

3 GRUNDLAGEN FÜR [NSCALE] PROJEKTE UND LEBENSZYKLEN

Dieses Kapitel gibt Hinweise und Anregungen bezüglich allgemeiner Projektvorbereitung, Umgebung und Einstellungen in nscale-Projekten im Hinblick auf das Einspielen und die Verwaltung von Customizing-Änderungen und Softwareupdates. Ziel ist der auch langfristig wirtschaftliche und gesicherte Betrieb der Lösung. Der jeweilige CM-Prozess soll von diesen Grundlagen profitieren.

3.1 Organisatorische Struktur

Es empfiehlt sich systematisches Projekt- und Change-Management z. B. in Anlehnung an [ITIL](#) zu betreiben. Basierend auf der erwarteten Komplexität und Lebensdauer eines Projekts soll eine wirtschaftliche Detaillierung dieser Struktur festgelegt werden.

Vor dem Beginn fachlicher oder technischer Planungen und Umsetzungen sollen die Ablage- und Auffindpunkte für die organisatorischen Informationen bereit stehen. So können sich aus der Situation ergebende Aufgaben, ergänzende Prüfvorschriften oder z. B. zukünftige sinnvolle Änderungen effizient abgelegt werden. Durch die Struktur ist eine mehr oder Minder automatische Nachverfolgung gewährleistet.

Informationsknoten einrichten

Sinnvoll ist oft ein Verzeichnis oder Portaleinstiegspunkt zu den wesentlichen Dokumenten, Listen und Datenbanken zum Projekt. Die Rollen und Verantwortlichkeiten bei der Pflege, Kategorisierung und Organisation der Maßnahmen sollen möglichst klar, verbindlich und eindeutig sein.

Sinnvolle Dokumente und Aufgaben- und Informations-Basen können z. B. sein:

- Checklisten zum allgemeinen Änderungsprozess.
(Nicht jedes Projekt rechtfertigt die Einrichtung einer wirklichen Steuerungsinfrastruktur.)
- CR-Datenbank bzw. Liste zur Änderungsplanung und -verfolgung
(ITIL empfiehlt eine CMDB, [Configuration Management Database](#).)
- Systematiken bzw. Checklisten zur Klassifizierung des Risikos einer Anpassung.
- Funktions- und QS-Checklisten
Übertragung Fachkonzepten, Lastenheften, Pflichtenheften in systematisch abarbeitbare Aufgabenrespektive Checklisten.

[Quelle, Referenz des Autors: [Informationsknoten einrichten](#)]

Anforderung einfache Nutzbarkeit

Nicht nur die Existenz dieser Informations- und Aufgabensammelpunkte ist wichtig. Entscheidende Bedeutung hat auch deren einfache Nutz- und Erreichbarkeit für die Projektbeteiligten.

3.2 3-stufige Systemlandschaft

Ceyoniq empfiehlt i. d. R. eine 3-stufige Systemlandschaft:

- Entwicklungssystem
- Q, QS bzw. Test-System:
Das Q-System sollte regelmäßige inhaltliche Kopie des Produktiv-System sein. So kann getestet werden, welche Auswirkungen auf die Daten die zu testende Änderung hat.

Durch Rückübertragung der Produktiv-Konfiguration nach Implementation zurück in das Q-System können Analyse-Lags vermieden werden.

- Produktivsystem

Einige Kunden haben basierend auf Ihrer Risikoklassifizierung und Größe des Projektteams entschieden Entwicklungs- und Q-System als ein System zu betreiben.

3.3 Datensicherung

Zeigen Sie die kritischen [Cls, also Systemkomponenten](#), beispielsweise in einer Systemskizze auf. Vielleicht nutzen Sie Visio und erstellen Verbinder auf einem Layer Datensicherung um alles diesbezüglich Relevante zu kennzeichnen.

Dokumentieren Sie den Versionsstand Ihres nscale-Systems und der in Relation befindlichen Systeme wie Datenbanken, Hardwaretreiber, Betriebssystem des Servers usw..

Speichern Sie die Setups und Dateien dieser zuvor beschriebenen Versionen in ein gut strukturiertes Verzeichnis. Sichern Sie dieses Verzeichnis. I. d. R. findet man sonst im Notfall nur mühselig die notwendigen Dateiversionen.

Prüfen und testen Sie Ihre Datensicherung vor der Inbetriebnahme Ihres Produktivsystems.

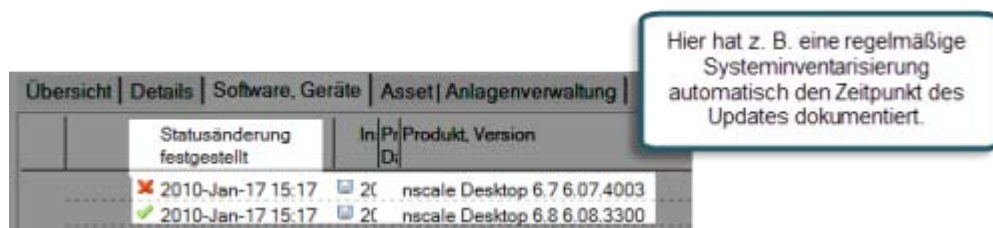
Ein Beispiel für eine diesbezügliche Übergabe und Abnahme finden Sie in [Sicherung und Wiederherstellung von Daten und Konfiguration - \[Verfahrens\]Leitfaden für nscale-Systeme.pdf](#).

3.4 Protokollierungen, Inventarisierungen, Snapshots

Protokolle im System

Nscale selbst besitzt diverse Möglichkeiten Systemzustände und Veränderungen zu protokollieren. Hierauf wird in nachfolgenden Kapiteln eingegangen.

Inventarisierung



Nutzen Sie ggf. System-Inventarisierung um Fehleranalysen durchzuführen oder Updates und Rollbacks usw. zu planen. Auf eine strukturierte administrative Datenbasis aufsetzend können wie in [Beispiel Eingrenzung auf Basis Inventory.pdf](#) gezeigt potentielle Ursachen für ein verändertes Systemverhalten eingegrenzt werden. Diese vom Autor selbst geschriebene Analyse basiert auf Daten die per RPM ([RH-InventoryLinux](#)) oder durch Dritthersteller-Werkzeuge ([RH-InventoryWindows](#)) gesammelt wurden.

VM-Snapshots und System-Images

Falls Sie Ihre Komponenten in virtuellen Maschinen betreiben, so denken Sie auch an die Möglichkeiten von Snapshots. | Achtung: Analysieren Sie in welchen Relation Echtdateien zurückgerollt werden und ob die Beziehungen zu anderen Systemen und Teildatenbeständen gültig bleiben.

4 NSCALE IM CHANGE-MANAGEMENT-PROZESS

Dieser Leitfaden soll die Werkzeuge und Möglichkeiten von nscale, welche in Change Management Prozessen genutzt werden können aufzeigen. In diesem Kapitel deshalb eine schematische Beschreibung von Change-Management-Prozess-Schritten mit Verweisen auf nscale-projektspezifische Werkzeuge.

Die Struktur dieses Kapitels folgt dem im Wikipedia-Artikel [Change Management \(ITIL\)](#) beschriebenen Lebenszyklus eines Änderungsauftrages respektive Changes. Der reale CM-Prozess kann in der jeweiligen Organisation etwas anders definiert sein wie z. B. in [Change Management Ablauf \(20100623\).pdf](#) beschrieben.

4.1 Request for Change

Ein Änderungsantrag wird erstellt. Verwaltung und Verfolgung mittels einer [organisatorischen Projektinfrastruktur](#).

4.2 Registrierung und Klassifizierung

In dieser Phase werden Informationen gesammelt. Aus fachlicher Sicht wird eine Priorität zugewiesen. Bestimmt wird eine Kategorie aus Sicht des Aufwandes für die Umsetzung der Änderung. Eine Risikoabschätzung ist grundsätzlich und für die Einschätzung des Aufwandes in Bezug auf Test-Detaillierung, Roll-Back-Vorbereitungen etc. notwendig.

nscale | Aufwandsabschätzung

Wie beschrieben, ist bei Abschätzung des Aufwandes auch jener für die relativ notwendige Risiko-Minimierung zu berücksichtigen.

nscale | Risiko-Klassifizierung

- Was muss getestet werden?
Mit welchem Aufwand ist es überhaupt möglich alle Auswirkungen zu testen?
Hier kann bereits die QS- und Funktionsliste für diesen Change begonnen werden.
[Zur Formatierung, Formalisierung und Ergänzung von Standardprüfungspunkten in Tabellen nutzt der Autor [excel_utils_&_formatting_public.xls](#).]
- Ist klar definiert welche Änderungen mit dem Update in das System transportiert werden?
Welche Relationen haben die geplanten Änderungen zu anderen Komponenten des Gesamtsystems?
- Welche Fehler können passieren,
welche Komponenten und [Configuration Items](#) (CI) stehen in möglicherweise beeinflusster Relation?
- Checkliste Risiko-Bewertung | s. [Kapitel Checklisten](#)
- Rollbackmöglichkeiten (unter Vermeidung von Daten-Lag's) | s. [Kapitel Checklisten](#)

4.3 Überwachung und Planung

Es erfolgt organisatorische Planung und Terminvorgaben. Insbesondere je nach im vorherigen Schritt erfolgter Kategorisierung und Risikoeinschätzung erfolgt eine weniger oder mehr detaillierte technische Planung. Dies ist dann auch der Beginn der Dokumentation des Soll-Zustandes.

nscale | und anderes – Grundsätzliches bei der Planung

- Welche Fehler können passieren, welche Komponenten und [Configuration Items](#) (CI) stehen in möglicherweise beeinflusster Relation?
- [Systemoptimierung durch Akzeptanz von Fehlern](#) (Beispiel Rechnungstransaktionen)
- [Minimierte Entwicklungs- und Fehlerbehebungsaufwände durch geplantes Kompensationsverhalten](#)

nscale | Datenrelationen

- Möglichst bei der Planung ein Zeitfenster reservieren in dem keine neuen Produktivdaten eingehen. Dabei muss dann ggf. auch die Zeit für ein mögliches Rollback einkalkuliert werden.

4.4 Genehmigung

Es wird entschieden, ob der Change durchgeführt werden soll.

4.5 Ausarbeitung und Test

Ausarbeitung der geplanten Änderungen

Bei der Ausarbeitung, dem Customizing respektive der Programmierung der geplanten Funktionselemente sollten die als ‚[Grundsätzliches bei der Planung](#)‘ aufgeführten Punkte Berücksichtigung finden.

Tests im Q-System

Testweise Implementierung ins Q-System und dort prüfen:

- Die möglichst bereits bei der Risikoklassifizierung und Planung erstellte QS- und Funktions-Liste ggf. noch ergänzen und dann hier abhaken.
- Ggf. auch den Back-Out-Plan prüfen und einen Rollback-Test durchführen.

4.6 Implementierung

Implementierung der Changes im vorgesehen Zeitrahmen. In Bezug auf nscale bedeutet dies das Einspielen der vorbereiteten oder ggf. nur geplanten Änderungen und Anpassungen.

Im Kapitel ‚[Übertragung von Anpassungen in nscale](#)‘ erhalten Sie Hinweise, Tipps und Best Practice-Beispiele wie Sie von einem Entwicklungs- oder Q-System die Änderungen und Anpassungen in ein anderes System übertragen.

Im Rahmen der Implementierung ist zu dokumentieren welche Änderungen vorgenommen wurden. Möglicherweise reicht die Ergänzung der Planungsdokumente. Dies hat den Vorteil, dass alles an einer Stelle bereitgestellt und auch redaktionell gepflegt wird. Hierbei ist allerdings zu beachten, dass jeweils auch der Soll-Zustand in der Dokumentation erhalten bleibt.

4.7 Auswertung

ITIL nennt als letzten Punkt die Auswertung.

Z. B. soll rekapituliert werden

- waren die Methoden zum Konfigurations-Transport gut gewählt und i. o.?
- Wurden neu „erfahrene“ Punkte im Rahmen der ständigen Verbesserung in den Verfahrens- und QS-Listen ergänzt?

- Kosten- / Nutzen i. o.?
- Weiteres siehe „[nscale Change Management \(ITIL\) - Wikipedia 20100614.pdf](#)“

Eine gute Organisationsstruktur unterstützt die fortlaufende Verbesserung: Es ist klar wo entsprechende Aspekte effizient abgelegt werden können.

[Quelle des Autors: [DA Workflow 'Continuous Improvement'.VSD](#)]

5 MÖGLICHKEITEN ZUR ÜBERTRAGUNG VON ANPASSUNGEN IN NSCALE

Dieses Kapitel referenziert Hinweise, Tipps und Best Practice-Beispiele wie Sie von einem Entwicklungs- oder Q-System die Änderungen und Anpassungen in ein anderes System übertragen.

Terminologie in diesem Bereich

Im Sinne von ITIL weicht der Begriff [Configuration Item](#) (CI) vom Begriff der Konfiguration wie man ihn oft im Zusammenhang mit nscale verwendet ab: Ein CI ist in der ITIL ein an den führenden Geschäftsprozessen beteiligtes Betriebsmittel. Dies meint PCs, Peripheriegeräte, Telefone, sämtliche Netzwerkkomponenten, Server, Installationsdokumentationen, Verfahrensanweisungen, IT-Dienstleistungen, Software (Sourcecode, Executables) bis hin zu Werkzeugen.

Dieses Kapitel betrachtet jedoch die verschiedenen Einspiel- und Übertragungsmöglichkeiten von nscale-Anpassungen. Im nachfolgenden sind deshalb mit Konfiguration oder Konfigurationselement Datenzusammenhänge gemeint, die die funktionellen und physischen Merkmale eines Produkts enthalten und festlegen.

Rollback, Back-Out-Möglichkeiten, Datensicherung der Konfigurationen und Versionen des Systems

Die in der Anlage verglichenen Techniken zur Übertragung von nscale Anpassungen, Einstellungen und Produktsoftware-Versionsständen können genutzt werden Checklisten für System-Backups zu erstellen. Weiterhin können die Informationen als Grundlage für Rollback-Vorgehenslisten dienen.

Möglichkeiten zur Übertragung von Anpassungen

Möglichkeiten und Abwägungen zum Transport, Sicherung und Wiederherstellung von Konfigurationselementen und Daten in nscale-Systemen finden Sie in der Anlage „[nscale Konfigurationen,- Rollout,- ChangeManagement - Listen, Matrizes.xlsx](#)“, Tabellenblatt „Cfg-Transport & BackOut“.

6 CHECKLISTEN ZU KLASSIFIZIERUNG, VERFAHREN UND PRÜFUNG

Methoden und Werkzeuge zur Risikoanalyse

Die Tabelle „Matrix Risiken nscale Changes“ in der Anlage „[nscale Konfigurations,- Rollout,- ChangeManagement - Listen, Matrizes.xlsx](#)“ zeigt Vorgehensweisen zur Risiko-Analyse mit nscale-Werkzeugen.

Matrix Risikobewertung nscale Items

Eine Matrix zur allgemeinen Risikobewertung von Changes an nscale-Systemen ist die Tabelle „Matrix Risiken nscale Changes“ in der Anlage „[nscale Konfigurations,- Rollout,- ChangeManagement - Listen, Matrizes.xlsx](#)“. Aufgrund der Vielzahl möglicher Variationen und Kombinationen vom Ineinandergreifen von der Konfigurations- und Anpassungsmöglichkeiten kann es sich nur um Anhaltspunkte handeln.

Abhängigkeiten

Weiterhin können auch die projektspezifischen fachlichen Auswirkungen eine Veränderung der Risikoklassifizierung bedeuten. Somit kann bzw. muss die Matrix in Bezug auf die im Projekt häufiger auftretenden Kombinationen konkretisiert werden.

Das Gesamt-Risiko ist auch abhängig von den Möglichkeiten des Back-Outs. Dies ist abhängig von der Datensicherung und [Rücktransportmöglichkeiten](#) der alten Konfiguration.

Fazit: Projektspezifische Checklisten und Prüfungen

Die diesem Dokument anliegende Arbeitsmappe „[nscale Konfigurations,- Rollout,- ChangeManagement - Listen, Matrizes.xlsx](#)“ kann in einem permanenten Prozess maßnahmen- und bereichsspezifisch ergänzt und verbessert werden. Hinsichtlich Versionierung und Pflege sei auf das Kapitel „[Organisatorische Struktur](#)“ verwiesen.

In jedem Fall sind vom Verantwortlichen die jeweilige Klassifikation und das Verfahren zu verifizieren.

7 BEISPIELE UND VORLAGEN FÜR CHANGES

7.1 Vorlagen

Arbeitsliste Change Implementierung

Eine fallspezifisch zu ergänzende Vorlage für eine Arbeitsliste ist das Tabellenblatt „Vorl. Checkl. Change Implement.“ in „[nscale Konfigurations-, Rollout-, ChangeManagement - Listen, Matrizes.xlsx](#)“.

7.2 Beispiele

Im folgenden Beispiele für häufig geforderte Systemänderungen. Ggf. bietet es sich an, diese projekt- bzw. bereichsspezifisch zu konkretisieren und als Referenz bzw. Vorlage in der Change-Management-[Organisations-Infrastruktur](#) zu kennzeichnen.

7.2.1 Beispiel mit Rollen

Rolle F erhält für Dokumentklasse X das Recht zu archivieren.

7.2.2 Beispiel mit Attributen

Attribut in Verträgen ergänzen

Im Bereich Vertragsmanagement wird ein neues Attribut eingefügt (Datenmodell, Maske, ...)

7.2.3 Beispiel mit DrittSystemanbindungen

Posteingangsmanagement Exchange

Ein neuer Bereich Posteingangsmanagement mit Schnittstelle zu Exchange wird implementiert

7.2.4 Anpassungsfehler und Patches („Bugfixes“)

Klasse XY

Maske YZ

8 WICHTIGE ANLAGEN

[nscale Konfigurations-, Rollout-, ChangeManagement - Listen, Matrices.xlsx](#)

- Matrix Risiko nscale Change
- BackOut-Möglichkeiten

[Sicherung und Wiederherstellung von Daten und Konfiguration - \[Verfahrens\]Leitfaden für nscale-Systeme.pdf](#)

- Leitfaden für Sicherung und Wiederherstellung von nscale-Systemen

9 QUELLEN

Notizen zur Entwicklung des Dokuments

[20100215 nscale Konfigurations-, Rollout-, ChangeManagement, VersionsInventory - Best Practice](#)

[Change Management \(ITIL\) - Wikipedia 20100614.pdf](#)

[Glossar - nscale Rollout, ChangeManagement, VersionsInventory.xlsm](#)

10 SUPPORT

Ihr erster Ansprechpartner für den Support ist der Calldesk der CeyonIQ Technology GmbH. So werden Ihre Anfragen zielgerichtet kanalisiert, schnell beantwortet und sind in späteren Fällen durch andere Mitarbeiter nachvollziehbar.

Tel: +49 521 9318-2222

Fax: +49 521 9318-882222

E-Mail: calldesk@ceyoniq.com

Weitere Informationen zum Thema Support entnehmen Sie bitte direkt unserer Website <http://www.ceyoniq.com/>. Im Bereich Support können Sie Fehlermeldeformulare downloaden.