5.3 Genehmigung, Planung und Umsetzung von Changes

> **Kurz erklärt**
>
> Ein Change sollte nicht nur technisch durchgeführt werden.
>
> Vor der Umsetzung müssen Ziel, Risiko, Auswirkungen, Verantwortlichkeiten, Freigaben, Kommunikation, Tests und Rückfallmöglichkeiten geklärt sein.
>
> Gute Change-Planung sorgt dafür, dass Änderungen kontrolliert, nachvollziehbar und mit möglichst geringem Risiko für laufende Services umgesetzt werden.

Warum Genehmigung und Planung wichtig sind

Viele Störungen entstehen nicht durch die Änderung selbst, sondern durch unklare Vorbereitung.

Typische Ursachen:

- Auswirkungen wurden nicht vollständig geprüft.
- Abhängigkeiten waren unbekannt.
- Benutzer wurden nicht informiert.
- Tests waren unzureichend.
- Rollback war nicht möglich.
- Zuständigkeiten waren unklar.
- mehrere Changes kollidierten zeitlich.
- Dokumentation wurde nicht aktualisiert.
- Service Desk wusste nichts vom Change.

Change Enablement soll solche Risiken reduzieren.

Grundprinzip

Ein Change sollte nur dann umgesetzt werden, wenn mindestens klar ist:

- was geändert wird,
- warum es geändert wird,
- welcher Service betroffen ist,
- welche Risiken bestehen,
- wer zuständig ist,
- wann die Umsetzung erfolgt,

- wie getestet wird,
- wie zurückgerollt werden kann,
- wer informiert werden muss,
- und woran Erfolg erkannt wird.

Je höher Risiko und Auswirkung, desto gründlicher müssen Planung und Genehmigung sein.

Typischer Ablauf

```
Änderungsbedarf erkennen
        ↓
Change Request erstellen
        ↓
Risiko und Auswirkung bewerten
        ↓
Change-Typ bestimmen
        ↓
Genehmigung oder Vorabgenehmigung prüfen
        ↓
Umsetzung planen
        ↓
Kommunikation vorbereiten
        ↓
Change durchführen
        ↓
Ergebnis testen
        ↓
Dokumentation aktualisieren
        ↓
Change abschließen oder nachbereiten
```

Dieser Ablauf ist ein Praxisbeispiel.

Die konkrete Umsetzung hängt von Organisation, Risiko und Change-Modell ab.

Change Request erstellen

Ein Change Request beschreibt die gewünschte Änderung.

Typische Inhalte:

- Titel,
- Beschreibung,
- Grund der Änderung,
- erwarteter Nutzen,
- betroffener Service,
- betroffene Systeme oder Configuration Items,
- Change-Typ,
- Risiko,
- Auswirkungen,
- geplanter Zeitpunkt,
- verantwortliche Person oder Gruppe,
- Umsetzungsplan,
- Testplan,
- Rollback oder Backout,
- Kommunikationsbedarf,
- Genehmigungsbedarf,
- Abhängigkeiten,
- Erfolgskriterien.

Ein Change Request sollte so formuliert sein, dass eine andere Person die Änderung bewerten kann.

Gute und schlechte Change-Beschreibung

Ungeeignet:

> Update Server.

Besser:

> Update des produktiven Webservers APP-WEB-01 von Version 2.4.57 auf 2.4.60, um bekannte Sicherheitslücken zu schließen. Betroffen ist der interne Service „Mitarbeiterportal“. Umsetzung im Wartungsfenster am 14.08.2026 von 20:00 bis 21:00 Uhr. Rollback durch Wiederherstellung der vorherigen Paketversion und Konfiguration möglich.

Eine gute Beschreibung macht Ziel, Umfang und Risiko sichtbar.

Genehmigung

Genehmigung bedeutet, dass eine berechtigte Person oder Gruppe der Umsetzung zustimmt.

Dabei wird geprüft:

- Ist der Change sinnvoll?
- Ist der Nutzen nachvollziehbar?
- Sind Risiken bekannt?
- Sind Auswirkungen bewertet?
- Ist der Zeitpunkt geeignet?
- Sind Tests ausreichend?
- Ist ein Rollback möglich?
- Sind Stakeholder informiert?
- Sind Ressourcen verfügbar?
- Gibt es Konflikte mit anderen Changes?

Genehmigung ist keine reine Formalität.

Sie soll eine bewusste Entscheidung ermöglichen.

Change Authority

Die Change Authority ist die Person oder Gruppe, die eine Änderung bewertet oder genehmigt.

Mögliche Change Authorities:

- Service Owner,
- Change Manager,
- technisches Fachteam,
- Produktteam,
- Informationssicherheit,
- Management,
- Change Advisory Board,
- Emergency Change Authority.

Die passende Change Authority hängt ab von:

- Risiko,
- Auswirkung,
- Change-Typ,
- betroffenen Services,
- Sicherheitsbezug,
- Kosten,
- und Dringlichkeit.

Nicht jeder Change benötigt dieselbe Genehmigungsstufe.

Genehmigung nach Risiko

Risiko	mögliche Genehmigung
niedrig	vorab genehmigter Standardprozess
mittel	Team Lead oder Service Owner
hoch	Change Manager, Service Owner oder CAB
sehr hoch	Management, Informationssicherheit, CAB oder spezielle Change Authority
akut kritisch	Emergency Change Authority

Diese Tabelle ist ein Beispiel.

Die Organisation muss ihre Genehmigungswege selbst festlegen.

Change Advisory Board

Ein Change Advisory Board kann bei umfangreichen oder risikoreichen Changes unterstützen.

Mögliche Aufgaben:

- Risiken bewerten,
- Abhängigkeiten erkennen,
- Change-Kollisionen vermeiden,
- Fachbereiche einbinden,
- Kommunikationsbedarf prüfen,
- Rollback bewerten,
- Zeitplanung abstimmen,
- Entscheidung vorbereiten.

Ein CAB sollte nicht jede kleine Änderung ausbremsen.

Es ist besonders dann sinnvoll, wenn mehrere Perspektiven benötigt werden.

Emergency-Genehmigung

Bei Emergency Changes muss oft schneller entschieden werden.

Trotzdem sollten mindestens geklärt werden:

- Warum ist die Änderung dringend?
- Welcher Schaden droht?
- Welche Services sind betroffen?
- Welche Risiken entstehen durch Umsetzung?
- Welche Risiken entstehen durch Nicht-Handeln?
- Wer trifft die Entscheidung?

- Wie wird dokumentiert?
- Wie wird nachträglich geprüft?

Auch ein Emergency Change braucht Kontrolle.

Nur der Genehmigungsweg ist beschleunigt.

Planung der Umsetzung

Ein Umsetzungsplan beschreibt, wie der Change praktisch durchgeführt wird.

Er sollte enthalten:

- Vorbereitungsschritte,
- Reihenfolge der Umsetzung,
- beteiligte Personen,
- benötigte Zugänge,
- benötigte Werkzeuge,
- Startzeit,
- erwartete Dauer,
- Prüfschritte,
- Abbruchkriterien,
- Rollback-Schritte,
- Kommunikationspunkte,
- Dokumentationshinweise.

Je kritischer der Change, desto genauer sollte der Plan sein.

Schrittfolge definieren

Eine gute Schrittfolge vermeidet Unsicherheit während der Umsetzung.

Beispiel:

1. aktuelles Backup prüfen
2. Konfiguration exportieren
3. Benutzerinformation bestätigen
4. Dienst in Wartungsmodus setzen
5. Update installieren
6. Dienst starten
7. Funktionstest durchführen
8. Monitoring prüfen
9. Fachbereichstest bestätigen
10. Wartungsmodus beenden
11. Abschlussmeldung senden

12. Dokumentation aktualisieren

Wichtig ist, dass jeder Schritt ein klares Ergebnis besitzt.

Vorbereitung

Vor der Umsetzung sollte geprüft werden:

- Liegt die Genehmigung vor?
- Ist der Zeitpunkt bestätigt?
- Sind alle Beteiligten verfügbar?
- Ist der Service Desk informiert?
- Sind Backups oder Konfigurationssicherungen vorhanden?
- Sind Zugangsdaten und Berechtigungen verfügbar?
- Sind Installationsdateien oder Pakete geprüft?
- Ist der Rollback vorbereitet?
- Ist Monitoring aktiv?
- Sind Kommunikationsvorlagen bereit?

Eine gute Vorbereitung verhindert Hektik während des Change-Fensters.

Wartungsfenster

Ein Wartungsfenster ist ein geplanter Zeitraum für Änderungen.

Zu berücksichtigen sind:

- Servicezeiten,
- Geschäftsprozesse,
- Benutzergruppen,
- internationale Zeitzonen,
- Produktionszeiten,
- Backupfenster,
- Lieferantenverfügbarkeit,
- Bereitschaftsdienst,
- Rollbackdauer,
- Kommunikationsbedarf.

Ein Wartungsfenster sollte nicht nur technisch bequem sein.

Es muss zur Nutzung des Services passen.

Kommunikation vor dem Change

Betroffene Personen sollten rechtzeitig informiert werden.

Je nach Change können beteiligt sein:

- Benutzer,
- Service Desk,
- Fachbereiche,
- Service Owner,
- Management,
- Lieferanten,
- Informationssicherheit,
- Bereitschaftsdienst.

Eine Vorabinformation kann enthalten:

- was geändert wird,
- warum es geändert wird,
- wann es passiert,
- welche Auswirkungen erwartet werden,
- welche Services betroffen sind,
- ob eine Unterbrechung entsteht,
- was Benutzer beachten müssen,
- wo Statusinformationen verfügbar sind,
- wann eine Abschlussmeldung erfolgt.

Beispiel einer Benutzerinformation

> Am 14.08.2026 zwischen 20:00 und 21:00 Uhr wird das Mitarbeiterportal aktualisiert. In diesem Zeitraum kann der Zugriff kurzzeitig unterbrochen sein. Bitte speichern Sie offene Eingaben vorher. Nach Abschluss informieren wir über die erfolgreiche Umsetzung.

Die Kommunikation sollte verständlich und nicht unnötig technisch sein.

Service Desk informieren

Der Service Desk muss vor wichtigen Changes wissen:

- welcher Service betroffen ist,
- wann der Change stattfindet,
- welche Symptome normal sind,
- welche Meldungen zu erwarten sind,
- welche Benutzer betroffen sein können,
- welche Statusmeldung freigegeben ist,
- wann eskaliert werden muss,

- wer Ansprechpartner ist.

Sonst entstehen unnötige Incidents und Rückfragen.

Tests planen

Ein Testplan beschreibt, wie geprüft wird, ob der Change erfolgreich war.

Mögliche Tests:

- technischer Funktionstest,
- Anmeldung,
- Datenzugriff,
- Schnittstellenprüfung,
- Druck oder Export,
- Performanceprüfung,
- Sicherheitsprüfung,
- Monitoringprüfung,
- Fachbereichstest,
- Smoke Test.

Der Test muss zum Risiko des Changes passen.

Bei kritischen Services reicht ein oberflächlicher Test oft nicht aus.

Smoke Test

Ein Smoke Test ist eine kurze Grundprüfung nach der Umsetzung.

Beispiele:

- Service startet,
- Login funktioniert,
- Startseite lädt,
- wichtige Funktion ist erreichbar,
- keine kritischen Fehlermeldungen im Monitoring,
- Schnittstelle antwortet.

Ein Smoke Test ersetzt keine vollständige Abnahme.

Er hilft aber, grobe Fehler sofort zu erkennen.

Rollback und Backout planen

Rollback oder Backout bedeutet, den Change zurückzunehmen oder in einen sicheren Zustand zu bringen.

Zu klären ist:

- Wie wird der vorherige Zustand wiederhergestellt?
- Welche Sicherungen werden benötigt?
- Wie lange dauert der Rollback?
- Welche Daten können betroffen sein?
- Wer entscheidet über Rollback?
- Wann muss abgebrochen werden?
- Welche Kommunikation erfolgt bei Rollback?
- Wird ein Incident erstellt?
- Welche Folgearbeiten sind notwendig?

Ein Rollback muss vor der Umsetzung geplant werden, nicht erst während der Störung.

Rollback ist nicht immer möglich

Manche Changes lassen sich nicht einfach zurückrollen.

Beispiele:

- Datenbankmigration,
- Datenlöschung,
- Änderung von Datenformaten,
- externe Schnittstellenumstellung,
- Cloud-Ressourcenänderung,
- Massenänderung von Berechtigungen.

Wenn kein einfacher Rollback möglich ist, müssen Risiko, Testtiefe und Kommunikationsplanung besonders sorgfältig sein.

Abbruchkriterien

Abbruchkriterien legen fest, wann ein Change gestoppt oder zurückgerollt wird.

Beispiele:

- Installation überschreitet geplante Dauer,
- kritischer Test schlägt fehl,
- Monitoring zeigt schwere Fehler,
- Datenbankmigration bricht ab,
- Rollback-Zeitfenster wird knapp,
- Fachbereich bestätigt Kernfunktion nicht,

- Sicherheitsprüfung schlägt fehl.

Abbruchkriterien verhindern, dass ein Change trotz deutlicher Warnzeichen weitergeführt wird.

Durchführung

Während der Durchführung sollte konzentriert und nachvollziehbar gearbeitet werden.

Wichtig:

- nur freigegebene Schritte durchführen,
- Änderungen dokumentieren,
- Abweichungen festhalten,
- nicht mehrere ungeplante Änderungen gleichzeitig durchführen,
- Monitoring beobachten,
- Kommunikationspunkte einhalten,
- bei Problemen rechtzeitig eskalieren,
- Abbruchkriterien beachten.

Ungeeignet:

> Während des Updates ändern wir zusätzlich noch ein paar andere Einstellungen.

Besser:

> Nur der freigegebene Change wird umgesetzt. Zusätzliche Auffälligkeiten werden dokumentiert und separat bewertet.

Umgang mit Abweichungen

Während eines Changes können unerwartete Situationen auftreten.

Dann sollte geklärt werden:

- Ist die Abweichung kritisch?
- Ist sie innerhalb der Freigabe abgedeckt?
- Erhöht sie das Risiko?
- Muss eine Entscheidung eingeholt werden?
- Muss der Change abgebrochen werden?
- Muss ein Incident erstellt werden?
- Muss die Kommunikation angepasst werden?

Nicht jede Abweichung darf spontan gelöst werden.

Sonst entsteht ein ungeplanter Change im Change.

Monitoring während des Changes

Monitoring sollte aktiv beobachtet werden.

Zu prüfen sind:

- Verfügbarkeit,
- Antwortzeiten,
- Fehlerraten,
- CPU,
- Arbeitsspeicher,
- Speicherplatz,
- Datenbankstatus,
- Schnittstellen,
- Logs,
- Sicherheitsmeldungen,
- Benutzeranmeldungen.

Monitoring zeigt nicht alles.

Es ergänzt technische und fachliche Tests.

Kommunikation während des Changes

Bei längeren oder kritischen Changes sollten Zwischenstände kommuniziert werden.

Beispiele:

- Change gestartet,
- Wartungsfenster läuft,
- technische Umsetzung abgeschlossen,
- Tests laufen,
- Verzögerung aufgetreten,
- Rollback eingeleitet,
- Change erfolgreich abgeschlossen.

Kommunikation verhindert Unsicherheit.

Sie muss aber sachlich und abgestimmt bleiben.

Erfolg prüfen

Nach der Umsetzung sollte geprüft werden:

- Wurde die Änderung vollständig umgesetzt?
- Funktioniert der betroffene Service?
- Funktionieren abhängige Services?
- Sind Kernfunktionen nutzbar?
- Sind Benutzer oder Fachbereich arbeitsfähig?
- Gibt es neue Incidents?
- Ist Monitoring unauffällig?
- Sind Logs unauffällig?
- Wurde der erwartete Nutzen erreicht?
- Muss Dokumentation aktualisiert werden?

Ein Change ist nicht automatisch erfolgreich, nur weil die technische Tätigkeit abgeschlossen ist.

Fachliche Abnahme

Bei wichtigen Services kann eine fachliche Abnahme erforderlich sein.

Beispiele:

- Fachbereich testet Kernfunktion,
- Key User bestätigt Arbeitsablauf,
- Produktteam prüft Schnittstelle,
- Service Owner bestätigt Nutzbarkeit,
- Informationssicherheit bestätigt Schutzanforderung.

Technische Erreichbarkeit allein genügt nicht immer.

Wichtig ist, ob der Service aus Sicht der Benutzer funktioniert.

Abschlussmeldung

Nach erfolgreicher Umsetzung sollte eine Abschlussmeldung erfolgen.

Sie kann enthalten:

- Change wurde abgeschlossen,
- betroffener Service ist wieder verfügbar,
- bekannte Einschränkungen,
- nächste Schritte,
- Ansprechpartner bei Problemen.

Beispiel:

> Das Update des Mitarbeiterportals wurde erfolgreich abgeschlossen. Der Service ist wieder verfügbar. Bitte melden Sie Auffälligkeiten über das Serviceportal unter Angabe des Services „Mitarbeiterportal“.

Dokumentation aktualisieren

Nach dem Change müssen relevante Informationen aktualisiert werden.

Mögliche Inhalte:

- Change Record,
- CMDB,
- Servicekatalog,
- Knowledge Base,
- Runbooks,
- Betriebshandbuch,
- Monitoringdokumentation,
- Known Errors,
- Problem Records,
- Installationsdokumentation,
- Sicherheitsdokumentation.

Ein Change ist unvollständig, wenn die technische Umgebung geändert wurde, die Dokumentation aber den alten Zustand zeigt.

CMDB aktualisieren

Wenn Configuration Items betroffen sind, sollte die CMDB aktualisiert werden.

Beispiele:

- neue Version,
- neue Beziehung,
- geänderte IP-Adresse,
- geänderter Standort,
- neuer Owner,
- ausgetauschte Hardware,
- neuer Lieferant,
- geänderte Schnittstelle.

Fehlende CMDB-Aktualisierung kann spätere Incidents, Problems und Changes erschweren.

Knowledge Base aktualisieren

Nach einem Change können Knowledge-Artikel falsch oder veraltet sein.

Zu prüfen ist:

- Stimmen Menüpfade noch?
- Haben sich Screenshots geändert?
- Gilt ein Workaround noch?
- Muss ein Known Error geschlossen werden?
- Müssen Service-Desk-Anleitungen angepasst werden?
- Müssen Benutzerhinweise aktualisiert werden?
- Gibt es neue Fehlermeldungen?
- Gibt es neue Eskalationswege?

Wissen muss zum aktuellen Servicezustand passen.

Problem Record aktualisieren

Wenn ein Change eine dauerhafte Lösung für ein Problem war, muss der Problem Record aktualisiert werden.

Zu prüfen ist:

- wurde die Ursache beseitigt?
- funktioniert der Workaround noch oder wird er entfernt?
- ist der Known Error noch gültig?
- sind zugehörige Incidents zurückgegangen?
- muss die Wirksamkeit noch beobachtet werden?
- gibt es verbleibende Risiken?
- kann das Problem geschlossen werden?

Ein Problem sollte nicht automatisch geschlossen werden, nur weil ein Change umgesetzt wurde.

Failed Change

Ein Change gilt als fehlgeschlagen, wenn:

- die Änderung nicht umgesetzt werden konnte,
- ein Rollback notwendig war,
- neue Incidents entstanden,
- Servicequalität schlechter wurde,
- Erfolgskriterien nicht erreicht wurden,
- unerwartete Risiken auftraten,
- oder der Nutzen nicht erreicht wurde.

Ein Failed Change sollte nachvollziehbar ausgewertet werden.

Ziel ist Verbesserung, nicht Schuldzuweisung.

Post Implementation Review

Ein Post Implementation Review prüft nach der Umsetzung, wie der Change verlaufen ist.

Mögliche Fragen:

- Wurde der Change wie geplant durchgeführt?
- Waren Risiko und Auswirkung korrekt bewertet?
- Waren Tests ausreichend?
- War Kommunikation rechtzeitig und verständlich?
- Gab es unerwartete Incidents?
- Hat der Rollback funktioniert oder wäre er realistisch gewesen?
- Wurde der erwartete Nutzen erreicht?
- Was sollte beim nächsten Mal verbessert werden?

Nicht jeder kleine Standard Change benötigt ein ausführliches Review.

Bei risikoreichen oder fehlgeschlagenen Changes ist es besonders wichtig.

Praxisbeispiel: Anwendungsupdate

Change

Produktive Fachanwendung wird aktualisiert.

Planung

- betroffener Service wird dokumentiert,
- Wartungsfenster wird festgelegt,
- Fachbereich wird informiert,
- Backup wird geprüft,
- Testsystem wird verwendet,
- Rollback wird vorbereitet,
- Smoke Test wird definiert.

Umsetzung

Update wird im Wartungsfenster durchgeführt.

Prüfung

- Anwendung startet,
- Benutzeranmeldung funktioniert,
- Kernprozess wird vom Fachbereich getestet,

- Monitoring ist unauffällig.

Abschluss

Dokumentation und Knowledge-Artikel werden aktualisiert.

Praxisbeispiel: Firewall-Regel

Change

Neue Verbindung zwischen Anwendung und Partner-API wird freigeschaltet.

Planung

- Quell- und Zielsysteme prüfen,
- Port und Protokoll dokumentieren,
- Sicherheitsfreigabe einholen,
- betroffene Services identifizieren,
- Testfall definieren,
- Rollback durch Entfernen der Regel beschreiben.

Umsetzung

Firewall-Regel wird im genehmigten Zeitfenster gesetzt.

Prüfung

Schnittstelle überträgt Testdaten korrekt.

Logs zeigen keine unerwarteten Verbindungen.

Praxisbeispiel: Rollback notwendig

Change

Datenbanktreiber einer Anwendung wird aktualisiert.

Problem

Nach dem Update schlägt der Export von Berichten fehl.

Entscheidung

Abbruchkriterium ist erfüllt.

Rollback wird eingeleitet.

Nachbereitung

- Incident wird verknüpft,
- Change als fehlgeschlagen dokumentiert,
- Ursache wird untersucht,
- Testplan wird erweitert,
- neuer Change wird vorbereitet.

Typische Fehler

Fehler 1

Change wird ohne klaren Plan umgesetzt.

Fehler 2

Genehmigung wird als reine Formalität behandelt.

Fehler 3

Service Desk wird nicht informiert.

Fehler 4

Benutzer erfahren erst durch eine Störung vom Change.

Fehler 5

Rollback ist nicht vorbereitet.

Fehler 6

Tests prüfen nur Technik, nicht Benutzer-Outcome.

Fehler 7

Abweichungen werden spontan und undokumentiert behoben.

Fehler 8

Mehrere zusätzliche Änderungen werden nebenbei durchgeführt.

Fehler 9

Change wird abgeschlossen, obwohl Erfolgskriterien nicht geprüft wurden.

Fehler 10

Dokumentation, CMDB und Knowledge Base bleiben veraltet.

Fehler 11

Problem Record wird nach dauerhafter Lösung nicht aktualisiert.

Fehler 12

Failed Change wird nicht ausgewertet.

Checkliste Genehmigung

- [ ] Change Request ist vollständig
- [ ] Nutzen ist nachvollziehbar
- [ ] Risiko ist bewertet
- [ ] Auswirkungen sind beschrieben
- [ ] betroffene Services sind bekannt
- [ ] betroffene CIs sind bekannt
- [ ] Rollback ist geprüft
- [ ] Tests sind beschrieben
- [ ] Kommunikationsbedarf ist bewertet
- [ ] Change Authority ist passend
- [ ] Genehmigung ist dokumentiert

Checkliste Planung

- [ ] Umsetzungszeitpunkt festgelegt
- [ ] Wartungsfenster bestätigt
- [ ] beteiligte Personen verfügbar
- [ ] Service Desk informiert
- [ ] Benutzerkommunikation vorbereitet

- [ ] technische Voraussetzungen geprüft
- [ ] Backup oder Sicherung geprüft
- [ ] Schrittfolge dokumentiert
- [ ] Testplan vorhanden
- [ ] Abbruchkriterien definiert
- [ ] Rollback oder Backout vorbereitet
- [ ] Lieferanten eingebunden, falls erforderlich

Checkliste Umsetzung

- [ ] Start des Changes dokumentiert
- [ ] nur freigegebene Schritte durchgeführt
- [ ] Abweichungen dokumentiert
- [ ] Monitoring beobachtet
- [ ] Kommunikationspunkte eingehalten
- [ ] Tests nach Umsetzung durchgeführt
- [ ] Fachbereich oder Service Owner beteiligt, falls nötig
- [ ] Abbruchkriterien geprüft
- [ ] Rollback durchgeführt, falls erforderlich
- [ ] Ergebnis dokumentiert

Checkliste Abschluss

- [ ] Erfolgskriterien geprüft
- [ ] Service wieder nutzbar
- [ ] abhängige Services geprüft
- [ ] Abschlussmeldung erfolgt
- [ ] Change Record aktualisiert
- [ ] CMDB aktualisiert
- [ ] Knowledge Base aktualisiert
- [ ] Runbooks aktualisiert
- [ ] Problem Record aktualisiert, falls relevant
- [ ] Incidents nach Change geprüft
- [ ] Review durchgeführt, falls erforderlich
- [ ] Lessons Learned dokumentiert

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker setzen Changes häufig technisch um.

Dabei reicht es nicht, nur den Befehl oder die Konfiguration zu kennen.

Wichtig ist auch:

- Auswirkungen zu verstehen,
- Risiken zu erkennen,
- Vorbereitungen zu treffen,
- Benutzer und Service Desk mitzudenken,
- Änderungen nachvollziehbar zu dokumentieren,
- Tests sinnvoll durchzuführen,
- Rollback realistisch zu planen,
- und aus Abweichungen zu lernen.

Gute Change-Umsetzung verbindet technisches Können mit sauberer Planung.

Zusammenfassung

> Change Request prüfen
> ↓
> Genehmigung und Change Authority klären
> ↓
> Umsetzungsschritte planen
> ↓
> Tests, Kommunikation und Rollback vorbereiten
> ↓
> Change im vorgesehenen Zeitfenster durchführen
> ↓
> Ergebnis technisch und fachlich prüfen
> ↓
> Abschluss kommunizieren
> ↓
> Dokumentation, CMDB und Knowledge aktualisieren
> ↓
> Failed Changes oder Abweichungen auswerten

Merksätze

>

> Ein Change ist mehr als die technische Änderung.

> Genehmigung bedeutet bewusste Risikoentscheidung.

> Gute Planung reduziert Incidents nach Changes.

> Rollback wird vor der Umsetzung geplant, nicht während der Störung.

> Der Service Desk muss wichtige Changes kennen.

> Ein Change ist erst abgeschlossen, wenn Ergebnis und Dokumentation geprüft sind.

Verwandte Seiten

- 5.1 Change Enablement – Ziele, Begriffe und Abgrenzung
- 5.2 Change-Typen und Risikobewertung
- 5.4 Release Management und Deployment Management
- 5.5 Change-Erfolg messen und Continual Improvement
- Incident Management
- Problem Management
- Knowledge Management
- Service Configuration Management

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Change Enablement
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Problem Management
- PeopleCert – ITIL Practice Guide: Knowledge Management
- PeopleCert – ITIL Practice Guide: Service Configuration Management

- ITIL Foundation – Version 5

Einordnung

Die dargestellten:

- Genehmigungsschritte,
- Planungsinhalte,
- Kommunikationshinweise,
- Rollback-Fragen,
- Testbeispiele,
- Checklisten,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Genehmigungsmatrix,
- CAB-Struktur,
- Change-Request-Vorlage,
- Rollback-Vorlage,
- Kommunikationspflicht,
- oder konkrete Testtiefe

für alle Organisationen vor.

Die konkrete Umsetzung muss an:

- Services,
- Risiken,
- Service Levels,
- Organisation,
- Sicherheitsanforderungen,
- Change-Modell,
- Lieferanten,
- Datenqualität,
- und verfügbare Fähigkeiten

angepasst werden.

Behandelter Framework-Stand: ITIL Version 5
Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance
Fachlicher Stand: August 2026
