

AAT Configuration Mgmt.

Auftrag

Erwerbe Dir Wissen zum Thema Configuration Management.

Anleitung

Configuration Management wurde „erfunden“ um ein Problem in IT-Projekten zu lösen, vermutlich ist Dir das schon selbst passiert: in einem Team arbeiten mehrere Menschen zusammen und schreiben jeweils Software-Module; diese Module bestehen den Modultest und werden nachher integriert; plötzlich funktioniert nichts mehr. Gründe? Mangelnde Schnittstellendefinition (Interfaces), aber viel zu oft auch verschiedene IDEs (was ist eine IDE?) oder zumindest verschiedene Versionen derselben.

Suche Dir verschiedene Definitionen von Configuration Management (einige davon sind ziemlich aufgeblasen komplex). Vorsicht: mittlerweile gibts das auch für Unternehmen. Fokussiere auf IT-Projekte, versuche aber auch auf allgemeine Projekte zu generalisieren. Welche Aspekte von Configuration Management gibt es?

Ziele

Zum Schluss musst Du

- (A) eine Definition des Begriffes geben können
- (B) erklären können, warum Configuration Management in einem IT-Projekt wichtig/sinnvoll ist.
- (B) den Unterschied zwischen Configuration Management für ein Projekt bzw ein Unternehmen beschreiben können.

Aufwand: ca 2 Unterrichtseinheiten

zusätzlich könntest Du

- (E) für ein vorhandenes (zB ITP_LA) oder ein zukünftiges (siehe dazu auch Thema „Projekthandbuch“) Projekt das Configuration Management festlegen (Partnerarbeit möglich)

Material

siehe Moodle (im Thema);
jedenfalls auch andere Ressourcen aus dem Internet

Ergebnis

eine übersichtliche Zusammenfassung Deines Wissens zum Thema, zB durch MindMap, Graf-iz, ...;

Configuration Management

1. Configuration Management – Definition

Configuration Management (CM) ist ein Prozess, der in der IT-Branche verwendet wird, um **Veränderungen** an den verschiedenen **Komponenten** eines Systems oder Produkts **zu verfolgen**, zu **dokumentieren**, zu **kontrollieren** und zu **organisieren**. Die Komponenten können Software, Hardware, Dokumente oder andere Dinge sein, die zu einem bestimmten Zeitpunkt in der Entwicklung oder im Betrieb des Produkts verwendet werden. CM ermöglicht eine **sorgfältige Verwaltung** von **Änderungen** an diesen **Komponenten**, um **sicherzustellen**, dass das **Endprodukt** den **Anforderungen entspricht** und **konsistent ist**.

Aspekte:

- **Konfigurationsidentifikation:** Identifizierung aller Komponenten des Produkts, einschließlich ihrer Versionen und Beziehungen zueinander.
- **Konfigurationskontrolle:** Überwachung und Verwaltung von Änderungen an den verschiedenen Komponenten, um sicherzustellen, dass sie geplant und genehmigt werden, und dass keine unerwünschten Auswirkungen auf das Gesamtsystem entstehen.
- **Konfigurationsstatusverfolgung:** Erfassung und Berichterstattung über den Status aller Komponenten und Änderungen im Verlauf des Projekts.
- **Konfigurationsprüfung und -audit:** Prüfung der Konfiguration auf ihre Vollständigkeit und Korrektheit, einschließlich der Überprüfung der Dokumentation und anderer relevanten Materialien.
- **Konfigurationsmanagementplanung:** Definition von Prozessen und Verfahren für die Durchführung von Konfigurationsmanagement im Projekt oder Unternehmen.

Wichtigkeit von Configuration Management

Configuration Management ist in einem IT-Projekt wichtig, weil es eine **genaue Verfolgung der Änderungen** an den verschiedenen Komponenten ermöglicht. Dadurch kann das Projektteam **schnell auf Probleme und Fehler reagieren**, die während des Entwicklungsprozesses auftreten können. Durch die Überwachung und Verwaltung von Änderungen an den Komponenten kann auch sichergestellt werden, dass das Endprodukt den Anforderungen entspricht und dass alle Komponenten in der richtigen Version verwendet werden.

Für **Unternehmen** ist Configuration Management wichtig, um sicherzustellen, dass die **IT-Infrastruktur konsistent und kontrolliert** ist. Durch die **Verfolgung von Änderungen** an der **IT-Infrastruktur** können **Unternehmen** auch schnell auf **Sicherheitsbedrohungen reagieren** und potenzielle Ausfallzeiten minimieren. Configuration Management hilft auch dabei, die **Kosten** für die **IT-Infrastruktur** zu **reduzieren**, da Unternehmen genau wissen, welche Komponenten vorhanden sind und wie sie verwendet werden.

Unterschied zwischen Configuration Management für ein Projekt bzw. ein Unternehmen:

Während **Configuration Management** für ein **IT-Projekt** sich auf die **Verwaltung** der **Komponenten** eines **bestimmten Produkts** konzentriert, geht Configuration Management für ein **Unternehmen** darüber hinaus und bezieht sich auf die **Verwaltung** der **gesamten IT-Infrastruktur** eines Unternehmens. Die **Ziele und Prozesse** sind **ähnlich**, aber Configuration Management für **Unternehmen** hat einen **breiteren Umfang** und erfordert eine **umfassendere Überwachung** der IT-Infrastruktur. Configuration Management für ein **Projekt** ist in der Regel auf einen **begrenzten Zeitraum** beschränkt, während Configuration Management für ein **Unternehmen** eine **kontinuierliche Aktivität** ist, die dazu beitragen soll, dass die IT-Infrastruktur des Unternehmens auf dem neuesten Stand bleibt und den Geschäftsanforderungen entspricht.

Ein weiterer Unterschied besteht darin, dass Configuration Management für ein **Unternehmen** oft von einer **zentralen Stelle oder Abteilung** durchgeführt wird, während Configuration Management für ein **Projekt** in der Regel vom **Projektteam selbst** verwaltet wird. Unternehmen benötigen möglicherweise auch spezielle Tools und Systeme, um Configuration Management durchzuführen, während in einem Projekt einfachere Tools und Prozesse verwendet werden können.

Insgesamt ist Configuration Management ein wichtiger Aspekt bei der Entwicklung und Verwaltung von IT-Produkten und -Systemen. Es ermöglicht eine sorgfältige Überwachung und Kontrolle von Änderungen an den verschiedenen Komponenten und stellt sicher, dass das Endprodukt den Anforderungen entspricht und konsistent ist.

Configuration Management in IT-Projekten:

Configuration Management (CM) ist ein **wichtiger Bestandteil der Softwareentwicklung**, der sich mit der Verwaltung von Softwarekomponenten und der Kontrolle von Änderungen an diesen Komponenten befasst. CM wird hierbei verwendet, um sicherzustellen, dass die verschiedenen **Komponenten** eines **Systems** miteinander **kompatibel** sind und dass die **Software** in einer **konsistenten Weise entwickelt** und **bereitgestellt** wird. Im Folgenden werden einige wichtige Aspekte von Configuration Management in IT-Projekten mit Fokus auf die Softwareentwicklung erörtert:

- **Versionskontrolle:**

Eine der wichtigsten Funktionen von CM in der Softwareentwicklung ist die Versionskontrolle. Dies umfasst die Verwaltung von Quellcode, Dokumentation und anderen Artefakten im Zusammenhang mit der Softwareentwicklung. Versionskontrollsysteme (VCS) ermöglichen es Entwicklern, Änderungen an der Software zu verfolgen, zu dokumentieren und zu überprüfen, bevor sie in das Hauptentwicklungszweig integriert werden. VCS helfen auch, Konflikte zu vermeiden, wenn mehrere Entwickler gleichzeitig an der gleichen Codebasis arbeiten.

- **Konfigurationsmanagementplan:** Ein Konfigurationsmanagementplan (CMP) ist ein wichtiger Bestandteil von CM in der Softwareentwicklung. Der CMP definiert die CM-Verfahren und -Praktiken, die im Projekt verwendet werden, und beschreibt die Rollen und Verantwortlichkeiten der beteiligten Personen. Der CMP umfasst auch die Definition von Konfigurationsobjekten (z.B. Quellcode, Binärdateien, Dokumentation) und deren Eigenschaften (z.B. Version, Status, Freigabedatum).
- **Build-Management:** Ein weiterer wichtiger Aspekt von CM in der Softwareentwicklung ist das Build-Management. Das Build-Management umfasst die Erstellung von ausführbaren Anwendungen aus dem Quellcode und anderen Artefakten. Ein Build-Management-System automatisiert diesen Prozess und stellt sicher, dass die Software in einer konsistenten Weise gebaut und bereitgestellt wird.
- **Test-Management:** CM ist auch wichtig für das Test-Management in der Softwareentwicklung. Testfälle und Testdaten müssen verwaltet und aktualisiert werden, um sicherzustellen, dass die Softwarequalität auf einem hohen Niveau bleibt. CM hilft auch, Änderungen an der Software zu verfolgen und zu dokumentieren, die während des Testprozesses vorgenommen werden.
- **Freigabemanagement:** Das Freigabemanagement umfasst die Verwaltung der Freigabe von Softwareversionen. CM hilft bei der Verfolgung von Änderungen an der Software und stellt sicher, dass jede Freigabe konsistent ist und den Anforderungen entspricht. Freigabemanagement umfasst auch die Dokumentation von Freigaben und die Verwaltung von Problemen und Fehlern.

Zusammenfassend ist Configuration Management in IT-Projekten mit Fokus auf die Softwareentwicklung von großer Bedeutung, um sicherzustellen, dass die Software in einer konsistenten Weise entwickelt, getestet und bereitgestellt wird. CM umfasst die Verwaltung von Quellcode, Dokumentation, Testfällen und Freigaben, um sicherzustellen, dass alle Komponenten eines Systems miteinander kompatibel sind und dass Änderungen kontrolliert und dokumentiert werden. Durch die Implementierung von CM in IT-Projekten können Unternehmen Zeit und Kosten sparen und das Risiko von Fehlern und Inkompatibilitäten reduzieren.

Darüber hinaus hilft **CM** Entwicklern, die **Arbeitseffizienz** und **Produktivität** zu **steigern**. Zum Beispiel können Entwickler dank der Versionskontrolle schnell zwischen verschiedenen Codeversionen wechseln und Konflikte zwischen verschiedenen Versionen vermeiden. Durch das Build-Management wird sichergestellt, dass jede neue Version der Software konsistent und korrekt gebaut wird. Das Test-Management ermöglicht es Entwicklern, Fehler schnell zu identifizieren und zu beheben, bevor sie in die Produktion gelangen.

Es ist wichtig zu betonen, dass **Configuration Management für IT-Projekte nicht dasselbe ist wie Configuration Management für Unternehmen**. Während Configuration Management für Unternehmen sich mit der **Verwaltung von Geschäftsprozessen** und **-ressourcen** befasst, konzentriert sich CM für IT-Projekte ausschließlich auf die Verwaltung von **Softwarekomponenten** und deren **Integration**. Configuration Management für Unternehmen

umfasst oft auch die Verwaltung von Hardware, Betriebssystemen und anderen nicht-Software-Komponenten, die für den Betrieb von IT-Systemen erforderlich sind.

- ➔ Zusammenfassend ist Configuration Management ein wesentlicher Bestandteil der Softwareentwicklung in IT-Projekten. Es hilft bei der Verwaltung von Softwarekomponenten, deren Integration und Freigabe sowie bei der Verwaltung von Tests und Fehlerbehebungen. Configuration Management kann dazu beitragen, die Effizienz und Produktivität von Entwicklern zu steigern und das Risiko von Inkompatibilitäten und Fehlern zu reduzieren.

Configuration Management – Interfaces, IDEs, ...

Configuration Management bezieht sich auch auf die **Verwaltung** von **Schnittstellen** (Interfaces) und verschiedenen **Entwicklungsumgebungen** (Integrated Development Environments, IDEs).

Eine der Herausforderungen bei der Entwicklung von Software in Teams besteht darin, dass verschiedene Entwickler möglicherweise unterschiedliche IDEs oder sogar unterschiedliche Versionen derselben IDE verwenden. Dies kann dazu führen, dass **verschiedene Versionen** von **Bibliotheken, Frameworks** oder **anderen Komponenten** verwendet werden, was zu **Inkompatibilitäten** und **Fehlern** führen kann.

Um diese Herausforderungen zu bewältigen, beinhaltet **Configuration Management** die **Verwaltung von Abhängigkeiten zwischen verschiedenen Softwarekomponenten, einschließlich Schnittstellen**. Eine Möglichkeit, dies zu erreichen, besteht darin, eine **einheitliche Entwicklungsumgebung** für das **gesamte Team** zu verwenden. Ein weiterer Ansatz besteht darin, sicherzustellen, dass alle Entwickler **dieselbe Version** der IDE und der verwendeten Bibliotheken und Frameworks verwenden.

Zusätzlich zur Verwaltung von Schnittstellen und IDEs umfasst Configuration Management auch die **Verwaltung von Konfigurationsdateien**, die für die Softwareentwicklung benötigt werden. Konfigurationsdateien können verschiedene **Informationen** wie **Umgebungsvariablen, Verbindungsparameter für Datenbanken** oder **Konfigurationen für andere Softwarekomponenten** enthalten. Eine effektive Verwaltung dieser Konfigurationsdateien kann dazu beitragen, Fehler zu vermeiden und die Bereitstellung von Software in verschiedenen Umgebungen zu erleichtern.

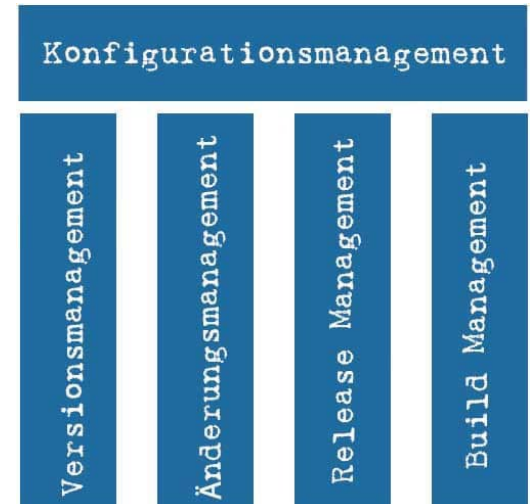
Zusammenfassend kann Configuration Management dazu beitragen, **Inkompatibilitäten** zwischen verschiedenen **Entwicklungsumgebungen** und **Versionen von Komponenten** zu **vermeiden**. Die **Verwaltung von Schnittstellen und Konfigurationsdateien ist ein wichtiger Bestandteil** des Configuration Managements, um sicherzustellen, dass die Software korrekt funktioniert und in verschiedenen Umgebungen bereitgestellt werden kann.

Das vier Säulen-Modell im Konfigurationsmanagement

Häufig wird Konfigurationsmanagement synonym mit *Versionsmanagement* verwendet, das sich aber nicht um zusammengehörende Arbeitsergebnisse, sondern auf die Verwaltung einzelner Dateien konzentriert. In manchen Publikationen wird das Konfigurationsmanagement auch als ein vier Säulen-Modell präsentiert:

- Versionsmanagement
- Änderungsmanagement
- Release Management
- Build Management

Die vier Säulen adressieren die Dokumentation sämtlicher Änderungen bei der Entwicklung von Software und Systemen mit dem Ziel, Anpassungen, Korrekturen und Erweiterungen kontinuierlich zu kontrollieren und steuern.



Begriffe im Konfigurationsmanagement

Wichtige Begriffe im Konfigurationsmanagement sind

- das **Atom** als **Element ohne weitere Teile**,
- die **Baseline** als Bezeichnung einer **Version einer Konfiguration**,
- die **Basiskonfiguration** als Bezeichnung der **zuerst gebildeten Konfiguration**,
- das **Konfigurationselement** als **Bestandteil einer Konfiguration**,
- das **Release** als Konfiguration, die an **Kunden geliefert** und entsprechend vermarktet wird,
- die **Revision** als Mittel den Status eines Konfigurationselements zu kennzeichnen und
- die **Zielkonfiguration** als gewünschtes Ergebnis einer Produkt- oder Softwareentwicklung.

Welche Antworten liefert Konfigurationsmanagement?

Durch das Konfigurationsmanagement können Organisationen bspw. folgende Fragen beantworten:

- Wenn eine Komponente geändert wird, welche Elemente sind betroffen?
- Welche Unterschiede gibt es zwischen zwei Konfigurationen?
- In welcher Konfiguration ist ein identifizierter Fehler zuerst aufgetreten und welche Auswirkung hat dies auf nachfolgende Konfigurationen?

CMDB

Eine CMDB (**Configuration Management Database**) ist eine **Datenbank** mit allen relevanten **Informationen zur IT-Infrastruktur**. In der CMDB werden u.a. die Objekte (**CIs - Configuration Items**) mit **Informationen** erfasst, die Beziehungen **zwischen** den **Objekten** dargestellt (Standort, Integration, einer Person zugeordnet / übergeben), mit **Dokumenten verknüpft**, der **Status** des **Objekts** (betriebsbereit / defekt / in Reparatur) und die verantwortlichen Personen dokumentiert. Da diese Datenbank bereits alle vorhandenen Informationen enthält und durch die Konfigurationsmanagementprozesse kontinuierlich gepflegt wird, sind wichtige Managementsysteme wie ITSM oder ISMS daran angebunden.

Die CMDB **setzt** die **Komponenten** zueinander in **Beziehung** und bietet so einen **Überblick** über die **Zusammenhänge** und **Abhängigkeiten** aller Komponenten über den gesamten Lebenszyklus hinweg.

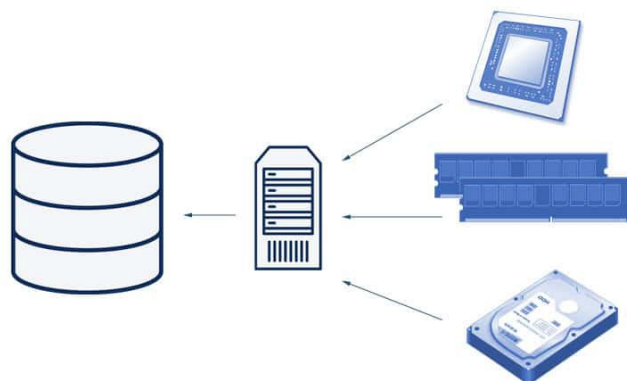
Der Vorteil: Auch wenn die einzelnen Geräte räumlich weit auseinander liegen und die dazugehörigen Daten aus ganz unterschiedlichen Quellen kommen, können mit einer CMDB alle Informationen an einem Ort verwaltet werden.

Configuration Items als Datensätze einer CMDB

Im Configuration Management ist das CI (so die gängige Abkürzung) das Atom, die kleinste unteilbare Einheit. Dabei darf das CI nicht mit einem realen Gerät verwechselt werden. Es handelt sich hierbei um die Abbildung des Gerätes innerhalb der CMDB. Im Deutschen ist auch oft die Rede von einem Konfigurationselement oder Konfigurationsobjekt. In i-doit bezeichnen wir die CIs einfach als "Objekte".

Der Begriff "Configuration" ist hier etwas irreführend. Es handelt sich nicht um wirkliche Konfigurationen von realen Geräten. Vielmehr steht der Begriff für die Darstellung der gegenseitigen Abhängigkeiten der gespeicherten Objekte.

Aus der Sicht der CMDB ist ein Configuration Item (CI) erst einmal ein Objekt. Dies kann alles sein, was zum Betrieb und der Aufrechterhaltung der IT- und Geschäftsprozesse notwendig ist. Der Arbeitsplatz-PC, der Drucker oder das Telefon sind in der Regel als CIs hinterlegt. Auch mit den Geräten in Zusammenhang stehende Objekte werden in einer CMDB als CI hinterlegt. Dies betrifft Rechnungen, Wartungsverträge oder Betriebshandbücher. Schließlich werden auch Server, Gebäude oder Personen als Items in der CMDB angelegt. Diesen CIs können verschiedene Eigenschaften zugewiesen werden.



Ein klassisches Anwendungsbeispiel dazu:

Ein Server wurde als Configuration Item angelegt. Diesem CI werden nun verschiedene technische Eigenschaften wie CPU, RAM und Storage, aber auch logische Eigenschaften wie zuständige Personen, Standort und Software zugeordnet. Das Item "Server" wurde also durch uns mit Informationen aus verschiedenen Quellen konfiguriert und wurde so zum Configuration Item.

Der Vorteil dieser Vorgehensweise ist die Vielfalt an Informationen, die wir zentral speichern können. So ist es uns nicht nur möglich, Informationen einfach nur zu erfassen, sondern auch Beziehungen und Abhängigkeiten zwischen den jeweiligen CIs darzustellen.

IT-Dokumentation und CMDB – gibt es Unterschiede?

Der größte Unterschied zwischen CMDB und IT-Dokumentation liegt in den Informationsquellen, der Datenhaltung und der Darstellung des Ist- und Soll-Zustands der IT-Infrastruktur.

Die IT-Dokumentation befasst sich ausschließlich mit dem IST-Zustand der Infrastruktur; die Dokumentation und Darstellung von Beziehungen und Abhängigkeiten ist sehr beschränkt. Hier liegt die Stärke der CMDB, die diese Zusammenhänge punktgenau abbilden kann. Sie verknüpft sowohl physische als auch logische Verbindungen zwischen den CIs und ermöglicht so einen vollständigen Überblick über die IT-Infrastruktur. Darüber hinaus kann Sie auch (Soll-) Zustände von Objekten darstellen, die sich, physisch gesehen, noch gar nicht im Unternehmen befinden.

So kann ein CI z.B. auch den Zustand "In Lieferung" oder "Bestellt" einnehmen, um die Ankunft im Unternehmen anzukündigen. Nicht selten handelt es sich hierbei um wichtige Informationen, um das Einbinden der Geräte in die IT-Infrastruktur fristgerecht umzusetzen. Diese Zustände sind Teil des Lebenszyklus eines jeden CIs und sollten für das Lifecycle Management unbedingt dokumentiert werden. Die CMDB übernimmt diese Arbeit für Sie.