Inhalts-Verzeichnis

3.3. Konfigurations-Überwachung

3.3.1. Neue CIs und Versionen registrieren
3.3.2. Software entwickeln
3.3.3. Standard CIs
3.3.4. Neue CIs und Versionen von building und releasing
3.3.5. Update von CIs
3.3.6. Lizenz Überwachung
3.3.7. Update und Archivierung der Konfigurations-Reporte von nicht benutzten CIs
3.3.8. Integrität der Konfigurationen beschützen
3.3.9. Update der CMDB

3.4. Configuration Status Nachweise

3.5. Verifikation und Audit

4) Configuration Management Prozess

5) Configuration Management Tools

6) Zusammenfassung

7) Literatur

1) Einführung und Motivation

1.1. Was ist ITIL

1.1.1. Warum wurde die ITIL eingeführt

IT Services sind eine Reihe von verbundenen Funktionen, die durch IT Systeme geliefert werden, um geschäftliche Anforderungen zu unterstützen und müssen vom Kunden als zusammenhängend, selbständig und integriert gesehen werden. Ziel des IT-Service Managements ist es, den Einsatz und die Wirkung der eingesetzten IT-Infrastruktur zu optimieren und zu maximieren.

Die Grundgedanken lauten:

- Kundenorientierung
- Prozessorientierung
- Qualitätsverbesserung [6]

In diesem Bewusstsein wurde die IT Infrastructure Library (ITIL) als internationaler de-facto-Standard für das IT-Service-Management geschaffen.

1.1.2. Aufbau der ITIL

Die IT Infrastructure Library- ITIL- besteht aus Modulen, die helfen, IT Ressourcen besser zu organisieren. Jedes ITIL Modul beschreibt eine einzelne Funktion des IT Service Managements und dessen Umsetzung in die Praxis. Funktionen können entweder nacheinander oder parallel zueinander geplant respektive verbessert werden. Die projektmäßige Umsetzung umfasst grob folgende Phasen(siehe Abbildung .1) [6]

Abb.1: Umsetzung des ITIL Module

Dank dem Ziel der vorbeugenden Planung und Durchführung von Maßnahmen hilft ITIL, die Wahrscheinlichkeit von Ausfällen zu reduzieren, die Ausfallfolgen und deren Dauer zu minimieren und im Katastrophenfall die Informatikdienstleistungen in der erforderlichen (mit den Anwendern vereinbarten) Zeit wieder schneller her- und sicherzustellen. Die meisten Risiken - wie Fehlen oder Inkonsistenz von Ablaufregeln, instabile Hard- und Software, ineffiziente Integrationen, Services, Support sowie fehlendes Wissen -, aber auch die gesamte Sicherheitskomplexität können mit ITIL proaktiv als Risiken definiert und gezielt und strukturiert minimiert werden. [6]

Der Haupt Grund für das Einführen eines CM ist ziemlich einfach: Zeit und Geld zu sparen. Z.B. hat die Erfahrung gezeigt, dass ein Item zu korrigieren ungefähr zehn Mal länger dauert, als es zu kodieren. Deshalb werden die hohen oder niedrigen Kosten in Manpower und Zeit jeder möglichen Absicherung oder Wartung des Programms direkt durch die hohe oder niedrige Qualität das Configuration Management beeinflusst.

1.2. Einführung ins Configuration Managament

1.2.1. Welche Rolle spielt Configuration Management in der ITIL

Das Configuration Management bildet mit der Configuration Management-Database (CMDB) die zentrale Informationsquelle für den IT Service. Mit Hilfe dieser gemeinsamen Datenbasis können die IT Services effektiver durchgeführt und Kosten optimiert werden. Dabei wird durch die Einbettung des Configuration Management in das IT Service Management sichergestellt, dass die notwendigen Schnittstellen zu den übrigen Disziplinen des IT Service Management bestehen und die erforderlichen Informationen allen Prozessbeteiligten zur Verfügung stehen. [4]

Das Configuration Management gewinnt in Unternehmen eine immer größere Bedeutung. Der Einsatz und die Verwaltung von Hardware und Softwarelizenzen wird - auch und gerade unter rechtlichen Aspekten - immer problematischer. Darüber hinaus liefert das Configuration Management wichtige Informationen für die Kosten- und Leistungsverrechnung bzw. das Service Level Management. [4]

1.2.2. Was ist Configuration Management

Die zentrale Rolle des Konfigurationsmanagements liegt darin, Informationen für andere Service Management Prozesse innerhalb eines Unternehmens oder einer Suborganisation zu Verfügung zu stellen. Durch ein Konfigurationsmanagement verfügt ein Unternehmen über:

- Eine genaue Kontrolle der eingesetzten Vermögenswerte in Form von einzelnen, identifizierbaren Configuration Items.
- Eine Grundlage für wirtschaftlich und qualitativ hoch stehende Informatikdienstleistungen. [6]

1.2.3. Zielsetzung

Mit dem Aufbau eines Configuration-Managements wird die folgende Zielsetzung verbunden: [19]

- **Genaue Informationen über CIs und ihre Komponenten zur Verfügung zu stellen**: Diese Informationen stützen alle weiteren Service Management Prozesse, wie Release Management, Change Management, Incident Management, Problem Management, Capacity Management und Planung Contingency

- **Kontrolle wertvoller CIs,** z.B., wenn ein Computer gestohlen wurde, muss dieser ersetzt werden. CM hilft dem IT Management zu wissen, was seine Werte sein sollen, was für ihren Schutz verantwortlich ist, und ob der tatsächliche Warenbestand das offiziell zusammenbringt.

- **Helfen bei der finanziellen Planung und beim Management**: CM liefert eine komplette Liste von CIs. Aus dieser Liste lassen sich einfach die erwarteten Wartungskosten und Lizenzgebühren, Wartungsverträge, Lizenz Erneuerungs-Daten,

CI-Lebensspanne-Verfallsdaten und CI-Wiederbeschaffungskosten herleiten. Indem es diese Informationen zur Verfügung stellt, trägt CM zur finanziellen Planung der IT Direktionen bei.

- **Software Veränderungen sichtbar machen**: Solche Maßnahmen können Untersuchungen durch das IT Management auf bestimmte mögliche Veränderungen (Changes) auslösen, die für Datenschutz, Lizenzmanagement und regelnde Befolgung erforderlich sein könnten.

- **Stützung und Verbesserung des Release Management**: CM Informationen stützen das „roll out" über verteilte Positionen, indem sie Informationen über die Versionen von CIs und über Veränderungen (Changes) zur Verfügung stellen, die in einem Release enthalten sein werden.

- **Sicherheit durch das Steuern der Versionen von CIs**: Dadurch gestaltet es sich schwieriger, dass CIs versehentlich oder böswillig geändert werden können oder fehlerhafte Versionen hinzugefügt werden können.

- **Durchführung der Organisation, Auswirkungsanalyse und Zeitplan: Change sicher, leistungsfähig und effektiv**. Dies verringert die Gefahr der Beeinflussung eines Changes durch die Umgebung.

- **Versehen des Problemmanagements mit Daten auf Tendenzen**: Solche Daten beziehen sich auf Tendenzen in den Problemen, die bestimmte CI Typen beeinflussen, z.B. bestimmte Lieferanten oder Entwicklungsgruppen, für die Verbesserung der IT Services. Diese Informationen über Problemtendenzen stützen die proaktive Vermeidung von Problemen.

2) Grund Konzepte

2.1 Configuration Item (CI)

2.1.1 CI Typen

CIs, die unter die Steuerung von CM fallen, umfassen Hardware, Software und dazugehörige Komponenten. Beispiele umfassen Dienstleistungen, Bediener, Klimata, Netwerk-Komponenten, Arbeitsplätze, mobile Einheiten, Anwendungen, Lizenzen, Telekommunikations-Dienste und Einrichtungen. Auf Grund der unterschiedlichen Typen, werden auch unterschiedliche Informationen zum entsprechenden CI benötigt und unterschiedliche Anforderungen gestellt. [6]

- **Hardware**

Abb.2 veranschaulicht die Ausprägung und den Detaillierungsgrad der verschiedenen Bereiche:

In vielen Unternehmen wird Hardware mit einer eindeutigen Inventarnummer versehen. Dadurch kann ein Gerät oder eine Komponente schnell identifiziert werden. Neben den üblichen Bezeichnungen des Gerätes und der Kategorie (PC, Drucker, Monitor) gehören auch gerätespezifische Informationen dazu. Abb. 3 und Abb.4 sind einige Beispiele für den Bereich Hardware: [6]

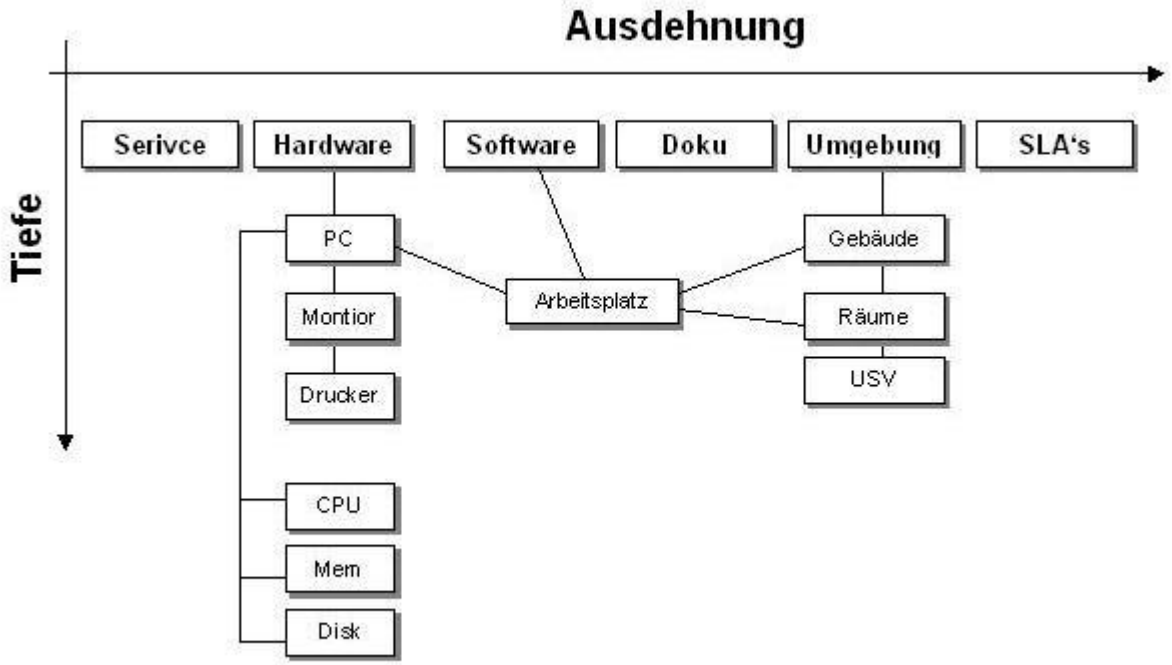

Abb.2: *Detaillierungsgrad und Ausprägung von CIs* [6]

Attribute	*Wert*	*Anzahl*
CPU	SPARC III	2
Takt	750 Mhz	
Cache	512 KB	
RAM	1 GB	
System Bus	128 bits	
RAID Disk Storage	120 GB	4
Disk Storage (Control)	RAID 5	
Internal Tape Drive	DAT, 8mm, DLT	
UltraSCSI Interface		1
Serial / Modem Ports		2
High-Speed Printer Ports		1
Network Adapter	100Mbps with RJ45 Interface	2
CD-ROM (Internal)	32X Speed, read-only	1
Keyboard	Local Language	1
Mouse	3 button	1
Parallel Centronics Ports		1
Disk Drive (3 1/2" Format)	1.44 MB	1
Internal Disk Hot-Pluggable	yes	

Abb.3: *Beispiel für CIs eines Midrange Servers* [6]

Attribut	*Wert*	*Anzahl*
Interface	Centronics parallel	1
Charakter Support	Local Language	
LCD Language Support	Local Language	
Dot Density	180 DPI	
Paper	500 Blatt	
Toner	Kassette xy	

Abb.4: *Beispiel für CIs eines Printers/Druckes* [6]

- **Software**

Da die dazugehörigen Attribute für alle Dateien identisch sind, ist es etwas einfacher diese Softwarekomponenten zu definieren.

Abb. 5. einige Beispiele für die Definition von Configuration Items aus dem Bereich Software:

Attribut	*Wert*
Name	xy
Version	3.2.1
Sprache	Deutsch
Kernel Version	2.4.1
Service Pack	5
Nur Lizenz	Yes
Mit CD-Rom	No

Abb.5: Beispiel für CI eines Betriebssystems ***[6]***

- **Dokumente**

Auch Dokumente, wie beispielsweise Benutzerhandbücher und Systemdokumentationen, müssen als Configuration Items definiert werden. Ähnlich wie bei der Software ist es auch für Dokumentationen einfach, die zugehörigen Attribute festzulegen, da sie für alle Dateien praktisch gleich sind. Abb.6 ein Beispiel für ein Benutzerhandbuch von MS Word: [6]

Attribut	*Wert*
Name	Benutzerhandbuch
Dateityp	MS WORD
Beschreibung	Benutzerhandbuch Anwendung
Dateiname	BH_AdAsp.doc
Status	Freigegeben
Grösse	20.5 KB
Erstellt	01.12.2002
Verändert	01.12.2002
Letzter Zugriff	10.12.2002
Dateiversion	2.5
Sprache	Deutsch
Anwendung	AdAspera
Anwendung Version	2.0

Abb.6: Beispiel CI Benutzerhandbuch ***[6]***

Zu jeder Hard- und Software gehört im Normalfall eine Dokumentation. Folgend ein Beispiel für den Detaillierungsgrad der Configuration Items Dokumentation (sieh Abb.7) [6]

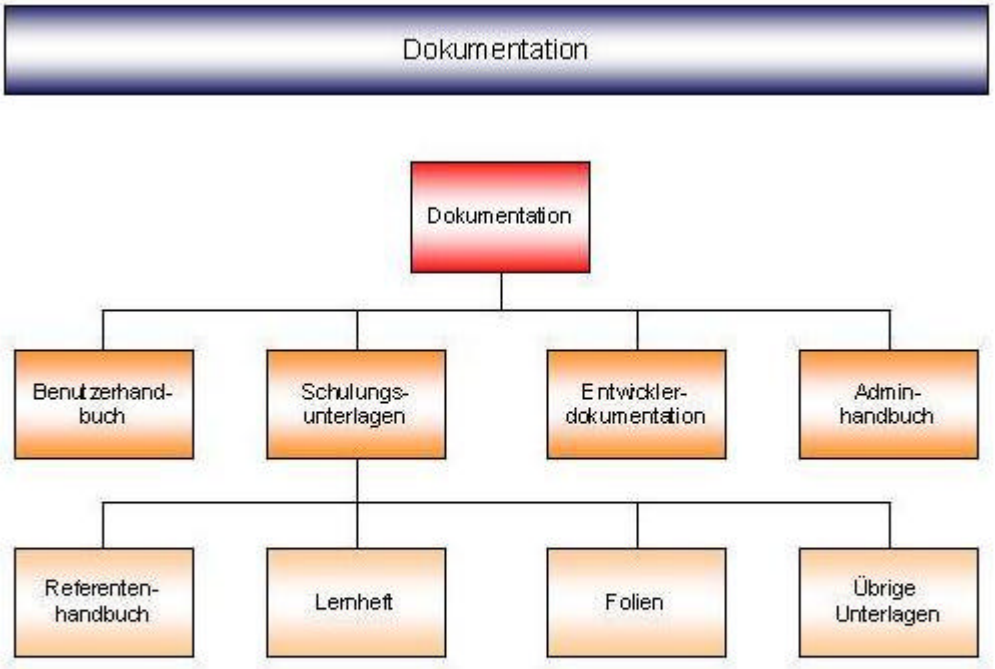

Abb. 7: Konfigurationseinheiten und Detaillierungsgrad Dokumentation [6]

2.1.2. CI Lebenszyklus

Die Lebenszykluszustände für jeden CI Type sollten definiert werden; z.B. ein Application Release kann registriert, akzeptiert, installiert, zurückgenommen werden. Ein Beispiel eines Lebenszyklus für ein Package Application Release wird in Abb.8 gezeigt.[19]

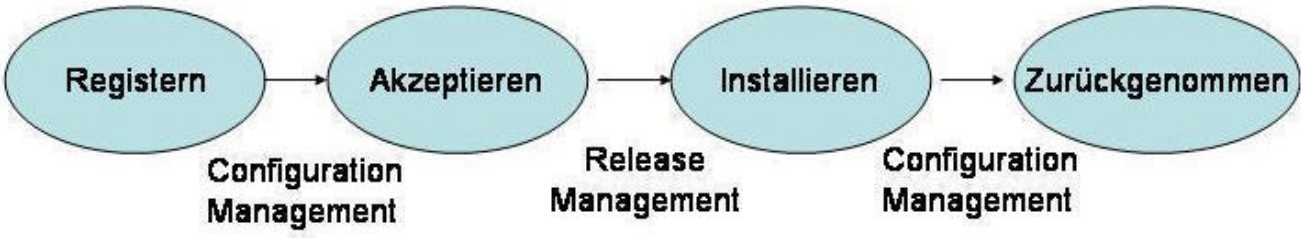

Abb. 8: Anwendungs-Release Lebenszyklus in der IT Infrasturktur

2.1.3. CI Beziehungen

Die Beziehungen zwischen CIs sollten gespeichert werden, um Abhängigkeitsinformationen zur Verfügung zu stellen. Beispiel , siehe bitte Abb.9

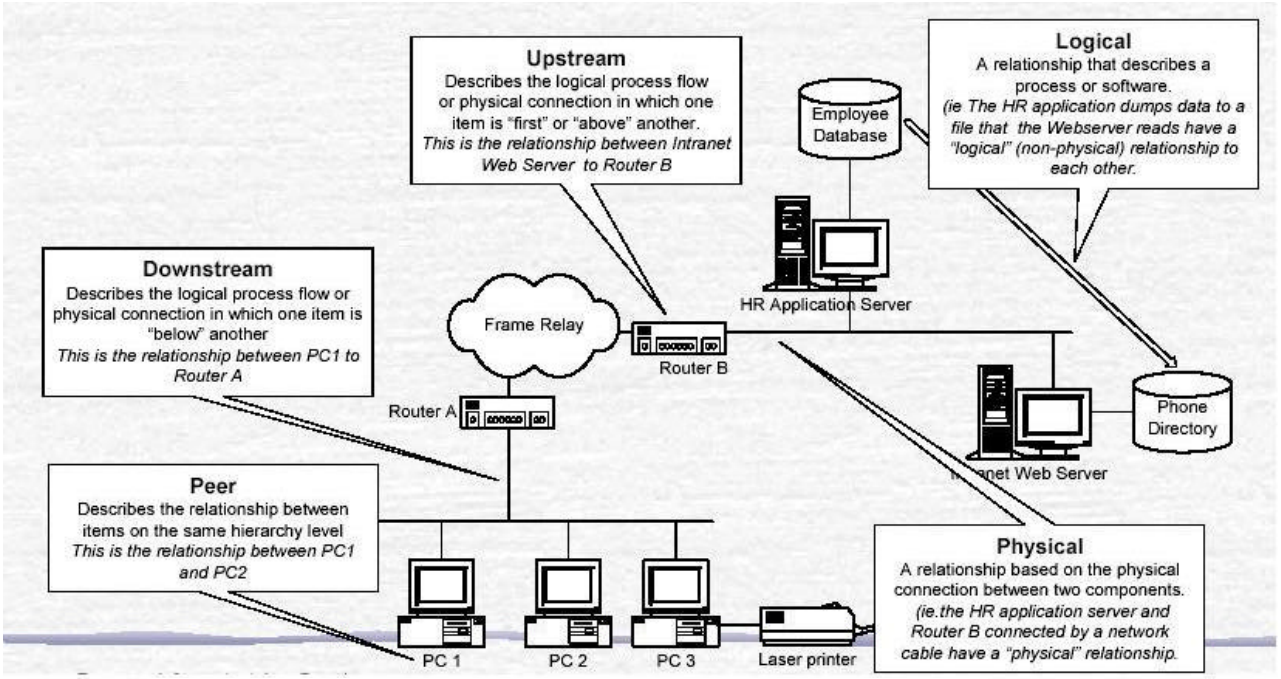

Abb. 9: Item-Beziehungen

Es kann viel mehr Typen von Beziehungen geben, aber alle diese Beziehungen sind im CMDB enthalten - das ist der Haupt-Unterschied zwischen dem, was ist in einer CMDB und was in einem Asset Register ist. Ein Mechanismus wird für die Verbindung von Problem-Protokoll, Incident-Protokoll, bekannten Fehlern und Release-Protokoll mit IT Infrastruktur CIs angefordert. Alle Beziehungen sollten in der CMDB enthalten sein. Alle Changes und Release Aufzeichnungen sollten das beeinflusste CI identifizieren.[19]

2.1. Configuration Management Database (CMDB)

Viele Organisationen benutzen bereits einige Elemente des Configuration Managements und verwenden häufig Spreadsheets, lokale Datenbanken oder Papiersysteme. In den heutigen großen und komplexeren IT Infrastrukturen, erfordert CM die Tools der CMDB. Physikalische und elektronische Bibliotheken zusammen mit der CMDB sind erforderlich, um endgültige Kopien von Software und Komponenten zu erhalten. Die CMDB ist wahrscheinlich, auf einer Datenbank-Technologie zu basieren, die die flexiblen und leistungsfähigen Abfragemöglichkeiten liefert. Auf der folgenden Abbildung sieht man die Beziehung zwischen der CMDB und anderen Prozessen.

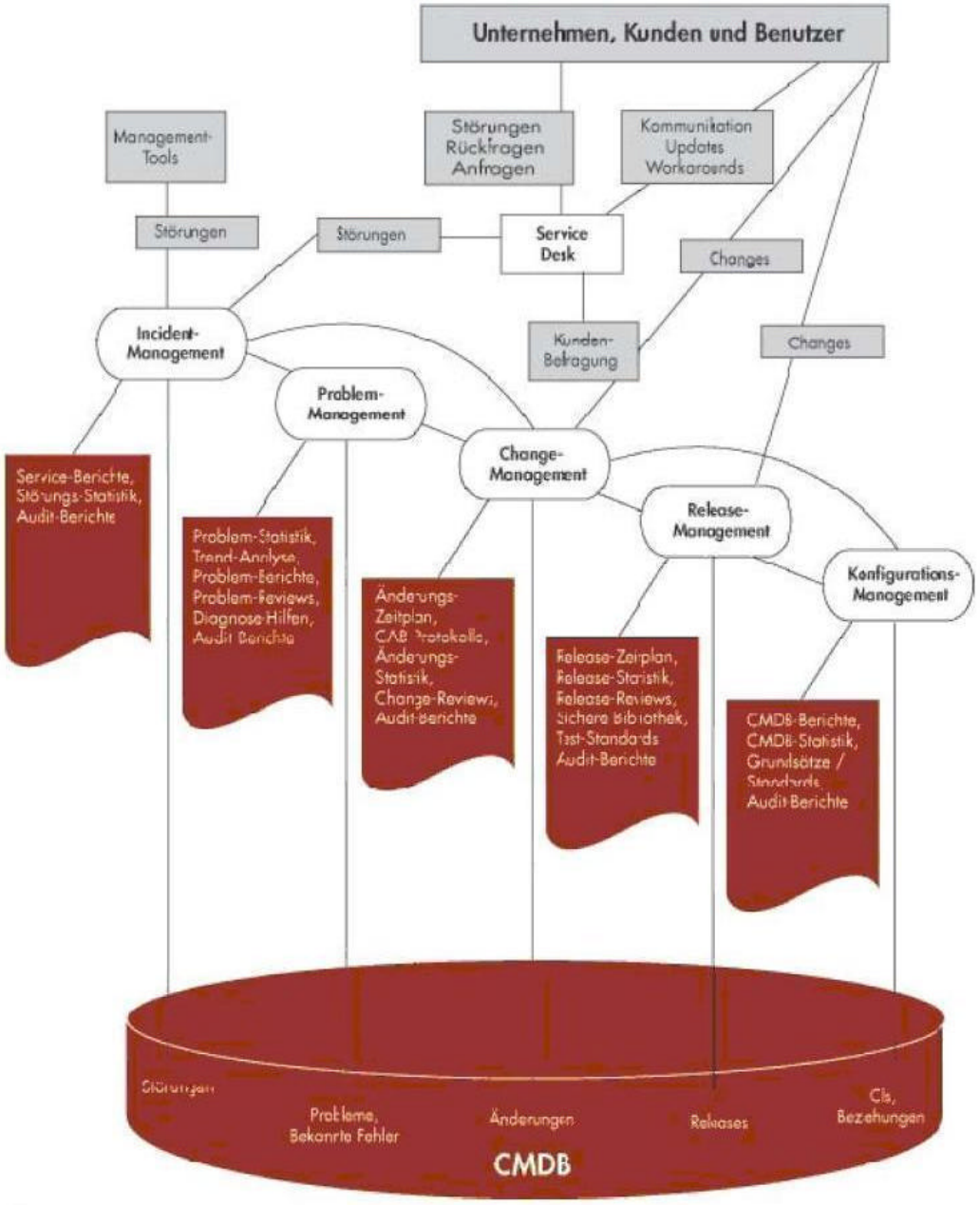

Abb.10: Service support Schnittstelle und Leistung [16]

Einige Beispiele seiner möglichen Nutzung sollen verzeichnen:

- Übernahme von Configuraion items: Übernehmen von Configuration items, welche immer wieder verwendet werden.
- Integration mit Service-Management-Prozessen: Die zentralisierte und Verwandtschafts-CMDB bietet Integration mit Service-Level-Management, Incident/Problem- und Change-Management Prozessen.
- Incident- und Problemmanagement: Netztopologien der verbundenen Konfigurationsitems können graphisch angezeigt werden, und einzelne Verbindungsarten zwischen Konfigurationsitems können geschaffen und definiert werden. Diese Informationen, sowie eine völlig revidierte Geschichte und genau

geschilderte Grundlinienkonfigurationsdaten sind alle für den Service fest vorhanden und stellen die Fähigkeit bereit, Problemkleinteile schnell zu lokalisieren.

- Change-Management: Die CMDB stellt graphisch Netzwerk-Verhältnisse dar und lässt die IT die Auswirkung der planned-/emergency Change auf die Infrastruktur festsetzen, um Service-Unterbrechungen herabzusetzen.
- Service-Level-Management: Service-Level Vereinbarungen können durch Configuration Items definiert werden. Die Konfigurations-Daten erlauben, dass die Geschäftsauswirkung der Stillstandszeit des CI festgestellt wird und lassen folglich alle möglichen Kundendienstvereinbarungen gegen dieses bestimmte Stück Infrastruktur laufen. [18]

Die CMDB sollte die Beziehungen zwischen allen System Komponenten, einschließlich der Incidents, Problemen, bekannten Fehlern, Changes und Releases enthalten. Die CMDB enthält auch Informationen zu Incidents, Problemen, bekannten Fehlern, und Firmendaten über Angestellte, Lieferanten, Positionen und Unternehmenseinheiten. Zusätzlich zur Speicherung von Personalinformationen, wird die CMDB häufig für Service-Level-Management verwendet, um Details von Dienstleistungen zu speichern und sie auf zu Grunde liegende IT Komponenten zu beziehen. Die CMDB ist auch für die Wartung von Lizenzen und von Verträgen zuständig.

2.3. Definitive Software Library (DSL)

In der Definitive Software Library (DSL) sind die definitiven autorisierten Versionen aller Software CIs gespeichert und geschützt. Es ist eine physikalische Bibliothek oder Speicher, in dem die Master Copy der Software-Version abgelegt wird. Dieser logische Speicher kann in Wirklichkeit aus einer oder mehreren physikalischen Software-Bibliotheken bestehen. Die Bibliotheken sollten getrennt von Entwicklung, Test oder live Speicher sein. Die DSL kann einen physikalischen Speicher auch mit Vorlagenkopien der eingekauften Software einschließen, z.B. einen feuerfesten Safe enthalten. Nur autorisierte Software sollte in die DSL aufgenommen werden. [14]

Das Beispiel bietet die Möglichkeit, den Entwicklungsprozess exemplarisch zu verfolgen: Der Entwicklungsprozess läuft nun in wie Abb.11 ab:

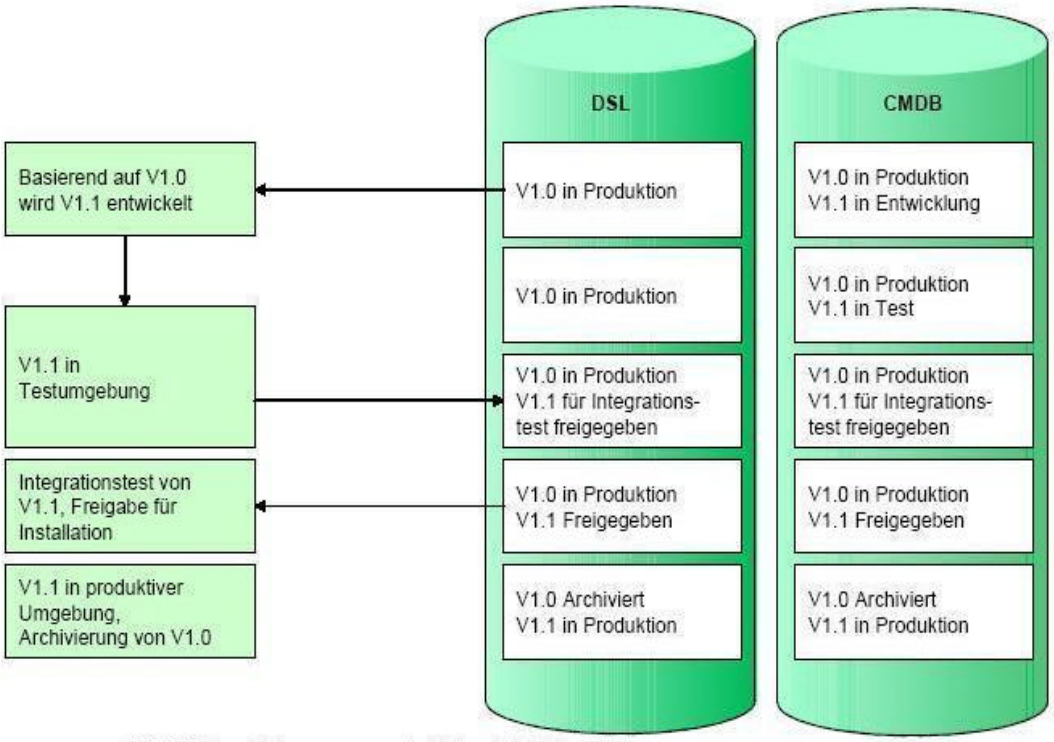

Abb.11: Entwicklungsprozess mit DSL und CMDB [16]

1) - Aufgrund eines Änderungsantrages übernimmt ein Entwickler die Version 1.0 einer Software in seine Entwicklungsumgebung mittels check-out auf

2) - Nach Beendigung der Entwicklungsarbeit wird die Datei der Testumgebung übergeben. Die Konfigurationsmanagement-Datenbank wird hierbei aktualisiert, damit erkennbar wird, dass die Entwicklungsarbeit abgeschlossen ist.

3) - Nach erfolgreichem Test wird die Version 1.1 (da ja Zusätzliches entwickelt wurde) mittels check-in wieder der Definitive Software Library zugeführt. Der Status ändert sich erneut.

4) - Um den Integrationstest durchzuführen, wird die Version 1.1 aus der DSL kopiert.

5) - Nach erfolgreichem Integrationstest der Version 1.1 wird die Version 1.0 archiviert

6) - Die nun abgelöste Version 1.0 wird nicht gelöscht, damit wir im Notfall auf diese Version zurückgreifen können.

2.4. Configuration Baseline

Eine configuration baseline ist die Konfiguration eines Produktes oder des Systems, die zu einer bestimmten Zeit hergestellt werden, welche die Struktur und die Details einer Konfiguration erfasst. Sie dient als Hinweis für weitere Tätigkeiten. Eine Anwendungs- oder Software-baseline liefert die Fähigkeit, eine spezifische Version zu einem späteren Zeitpunkt zu ändern oder umzubauen.

Eine Configuration baseline ist auch ein Snapshot oder eine notierte Position. Obwohl die Position später aktualisiert werden kann, bleibt die configuration baseline als der ursprüngliche Zustand erhalten und kann somit mit der gegenwärtigen Position. Eine configuration baseline sammelt alle relevanten Komponenten in der Bereitschaft für eine Änderung oder einen Release und bietet die Basis für einen Konfigurations-Audit und eine Konfigurations-Regression. Nach einer Änderung z.B., sollte das CM System eine configuration baseline, seinen Inhalt und Komponenten speichern, sich sichern und berichten.

Configuration baselines sollten die dazugehörigen Konfigurations-Komponenten umfassen,

- die Release Reporte (aktuell, vergangen und geplant)
- die Change Reporte (aktuell, vergangen und geplant)
- die Zustände des Systems und seiner Komponenten, wenn eine Änderung akzeptiert ist und wenn sie installiert wird
- die Zustände eines Systems und seiner Komponenten, wenn ein Package Release installiert wird.
- Hardware und Software Standardspezifikationen

3) Configuration Management Aktivitäten

Configuration Management umfasst die Planung, Identifikation, die Aufnahme und die Beziehung von IT Komponenten und Beziehungen.
Die grundlegenden Tätigkeiten von CM sind wie Abb12:

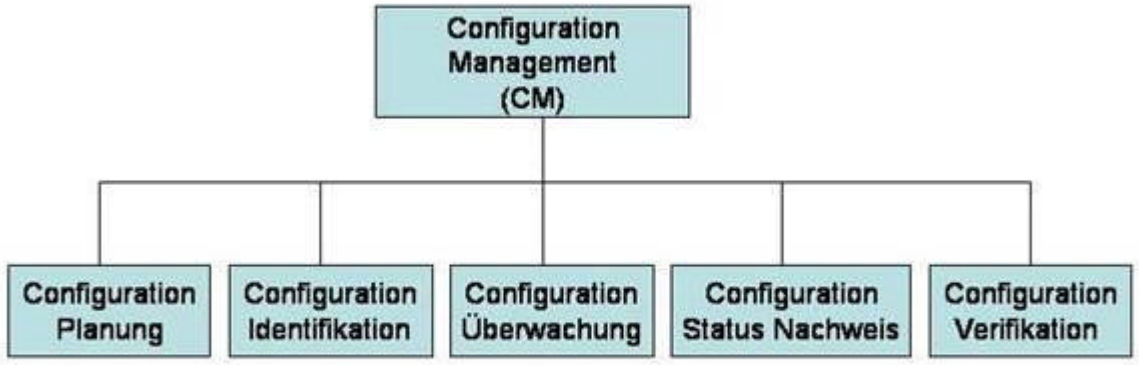

Abb.12: Configuration Management Aktivität

Szenario

Als Szenario planen wir ein Configuration Management für das Extranet der BMW AG. Der Begriff Extranet drückt hierbei die organisatorischen Beziehungen aus: Die Händler stellen eigenverantwortliche Organisationen dar, die mit BMW in einem vertraglich zugesicherten Verhältnis stehen. Damit beschränkt sich der Zugriff für Händler auf ausgewählte Anwendungen, die getrennt vom Unternehmensnetz zur Verfügung gestellt werden. Extranet Management ist eine wichtige Technik des Extranet der BMW AG. Das Extranet Management der BMW AG befasst sich mit allen Tätigkeiten der Informationsabwicklung und, den Netz Kommunikationsdienst zwischen BMW und ihren Händler zu überwachen und zu organisieren [20]. Das Ziel ist, die einwandfreie Funktion des Extranets der BMW AG zu

garantieren, au außergewöhnliche Vorkommnisse sofort zu reagieren und einen möglichen Zusammenbruch zu regeln. Das Extranet Configuration Management ist ein wichtiges Element des Extranet Management. Das Extranet besteht aus vielen verschiedenen in der Fabrik hergestellten Produkten. Die Informationen zu den Produkten, z.B. Parameter und Status, gegenseitig zu verstehen und anzupassen wäre notwendig. Insbesondere ist das Extranet System oft dynamisch. Deswegen sollte das Extranet, durch Wartung oder Erneuerung der Betriebsausrüstung, die Netzteilung regulieren. Netzwerk Configuration Management dokumentiert die komplette zu monitorende LAN- und WAN-Umgebung des Netzwerk Inventars. Diese Informationen sind in anderen Servicebestandteilen absolute Voraussetzung, wie beispielsweise beim Problem Management, um den Fehlerlösungs- und Eskalationsprozess zu unterstützen. Durch die Sicherung und Archivierung der Konfiguration der aktiven Komponenten kann die Wiederherstellung im Crashfall erfolgen.

3.1. CM Planung

Erst wird feststellt, und zwar auf einem hohen Niveau, was für die Gesamtkonfiguration erforderlich ist. Dies umfasst Planung und Definition, deren Zweck, Bereich, Zielsetzungen, politische Linien und Prozeduren und deren organisatorischen und technischen Kontext für das CM.

Configuration Management Planung besteht aus der Zustimmung und dem Definieren folgender Punkte:

- die Strategie, die Politik, der Bereich und die Zielsetzungen des Konfigurationsmanagements
- die Analyse der gegenwärtigen Position des Vermögens und der Konfigurationen
- der organisatorische, technische und managehafte Kontext, mit dem die Configuration-Management-Tätigkeiten eingeführt werden sollen
- den strategische Linien für in Verbindung stehende Prozesse wie Change Management und Release Management
- Interfaces, z.B. zwischen Projekten, Lieferanten, Applikation und Support teams
- die relevanten Prozesse, Prozeduren, Richtlinien, Support tools, Rollen und Verantwortlichkeiten für alle Configuration-Management-Tätigkeiten
- die Position der Speicherbreiche und der Library, die Hardware, Software und Komponenten enthalten.[19]

Die Configuration policy/strategy ist die Zielsetzung und die key success factors (KSFs) von dem, was vom Configuration Management erzielt werden sollte. Die ausführlichen Tätigkeiten und die Betriebsmittel werden angefordert, um die Zielsetzungen zu erreichen und die KSFs in der Strategie können in einem Projektplan dokumentiert werden. Die Meilensteine werden häufig im Configuration management plan zusammengefasst.[19]

Szenario

Für das Extranet der BMW AG sollten wir uns in der CM Planung mit den Aussagen über Strategien, Technologien, Vorgehensweisen, Aktivitäten des BMW Extranet befassen, sowie mit den Ausagen über Aufbau- und Ablauforganisation, Personalbedarf und -qualifikation, Ausbildung, Sachmittel des BMW Extranets, der generellen Beschreibung der Route, Workstations und Server usw. und einer geeigneten Datenbank und der Lokalisation der Datenbank.

3.2. Konfigurations-Identifikation

Der Hauptbaustein jedes möglichen CM ist die Konfigurations-Identifikation. Denn es ist ziemlich einfach: Wenn Sie etwas nicht identifizieren können, können Sie es nicht kontrollieren! Konfigurations-Identifikation ist, die Konfigurations-Strukturen für die ganzen CIs der Infrastruktur, einschließlich ihres Inhabers, ihre Beziehungen und Konfigurations Dokumentation zu selektieren und zu identifizieren. Sie umfasst das Zuteilen der Identifikations- und der Versionsnummern für CIs, das Beschriften jedes CIs und das Eintragen der CIs in die Configuration Management Database (CADM). Siehe bitte Abb.15

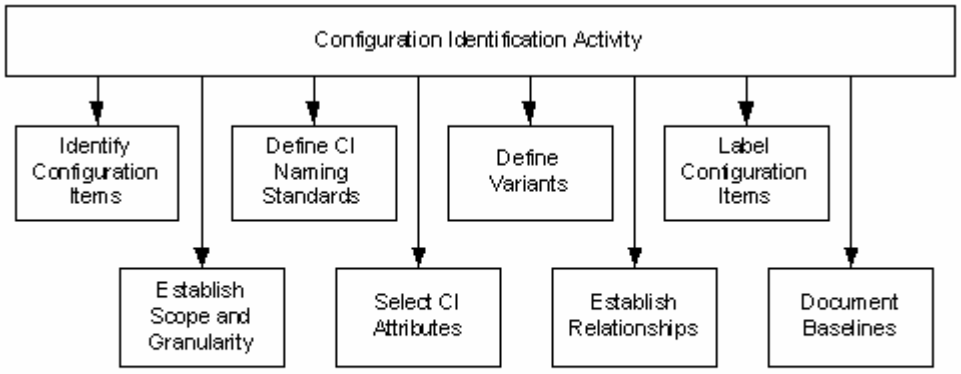

Abb.15: Configuration Identifikation aktivität [1]

3.2.1. Konfigurations-Strukturen und die Selektion von CIs

Konfigurations-Strukturen sollten die Beziehungen und die Position von CIs in jeder Struktur beschreiben. Z.B. eine Konfigurations-Struktur für das BMW Extranet verwendet wie Abb.13 Infrastruktur CIs, Server, Netz und Software CIs. Mehrfache Ansicht , wie Abb.14 durch unterschiedliche Konfigurations-Strukturen verbessert die Stossanalyse, Service Reporting, Change Management und Release Management.[19]

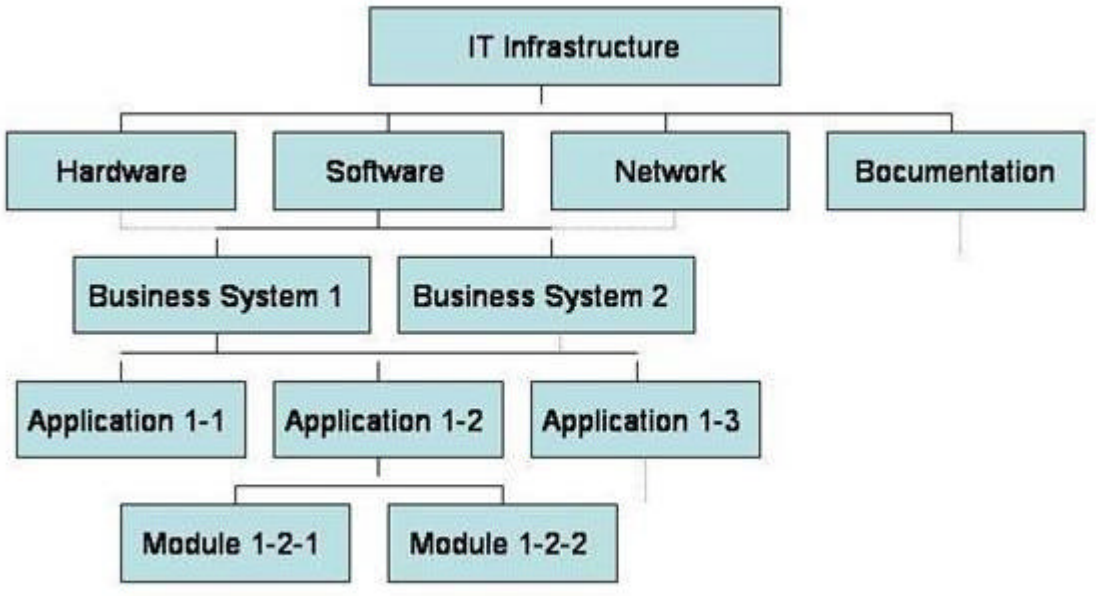

Abb.13 Beispiel: Configuration breakdown Structure-1

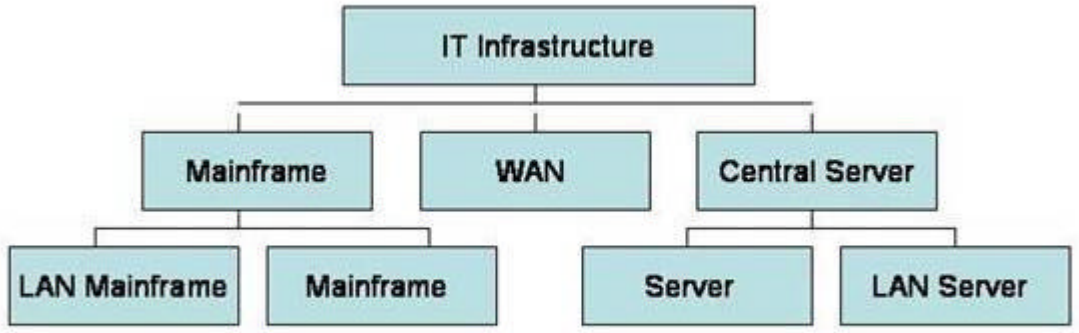

Abb.14 Beispiel: Configuration breakdown Structure-2

CIs sollten selektiert werden, indem man einen Zerlegungsprozess am Top Niveau Item mit einigen Anleitungskriterien für die Selektion von CIs anwendet. Ein CI kann aus einer möglichen Menge unterschiedlicher CIs oder nur aus sich selbst bestehen. Zum Beispiel kann ein Geschäftssystem des BMW Extranet durch viele Applikation benutzt werden.[19]

Das gewählte CI Niveau hängt von den Geschäfts- und Service-Anforderungen ab. Das niedrigste CI Niveau im Voraus ist notwendig, sogar wenn Sie nicht sofort die CMDB unten bis zu diesem Niveau bestücken. Es lohnt sich, Zeit auf dieser Tätigkeit auszugeben und voraus zu schauen. Es kann Reorganisierungs-Kosten der CMDB zukünftig sparen. Jedoch ist es nicht immer einfach das passende Niveau von CIs im Voraus zu wählen. Wenn möglich, sollte man ein CM-Tool verwenden, das nicht übermäßig weitere CIs bis zum untersten Niveau unterteilt. Wenn dies nicht möglich ist, sollte man ein Tool wählen, das die Aufnahme der Eigenschaften einzelner CIs erlaubt.[19]

Wenn Informationen auf niedrigem CI Niveau nicht wertvoll sein würden, z.B. wenn eine Tastatur nicht oft ausgetauscht wird, oder die Organisation es als Verbrauchsmaterial sieht, speichert man diese nicht. Informationen sind nur wertvoll wenn das Change Management, die Kontrolle von Release und Problemen erleichtert, oder die Kontrolle des Vermögens, das unabhängig verschoben, kopiert oder geändert werden kann. Die Organisation sollte planen, das CI Niveau regelmäßig nachzuprüfen. Damit diese Informationen für ein niedriges Niveau noch wertvoll und nützlich sind und für die Behandlung der Änderungen und der Probleme und des Assets Managements reichen, muss das CMDB zu einem genügend niedrigen Niveau gehen[19].

Szenario

Die anfängliche Tätigkeit des Konfigurations-Managements des BMW Extranets ist, einen vollständigen Datensatz der Netzeinrichtung, (z.B. Router, LAN-Switch, WAN-Switch oder ATM-Switch) des Extranets und Software zu selektieren. Das kann manuell oder automatisch erreicht werden. Der Datensatz der Netzeinrichtung sollte möglichst detailliert werden. Folgende Tabelle ist ein Beispiel für einen Datensatz der Netzeinrichtung**.** [15]

Device Data	Device Access Data	Device Configuration Data (Based On Device Type)
Device Name	IP Address	Router Configuration (Memory, Power, etc)
Type	MAC Address	Router Interface (I/F) Types (IP, ATM, FDDI, etc)
Vendor Name	Gateway Address	Router I/F Configuration Data (IP, IP Multicast, LANE, IPX, etc)
Model	Access Authorization	Routing Protocols (BGP, OSPF, etc.)
Revision	Passwords	
Date of Purchase	Dial-up Numbers	
SNMP Managed or Not		
Maintenance Data		

Abb.16: allgemeine Datensatz des Netzeinrichtung Item [15]

3.2.2. Configuration baselines identifizieren

Configuration Baselines sollten durch formale Vereinbarung zu einem bestimmten Zeitpunkt hergestellet werden und als Abfahrtspunkte für die formale Überwachung einer Konfiguration benutzt werden. Einige Baselines, die unterschiedlichen Zuständen im Lebenszyklus eines Baseline Items entsprechen, können zu jeder möglichen gegebenen Zeit bestehen. Z.B. sind die Phasen der Baseline für ein Software Release in Extranet, „aktuell live“, „zuletzt live“ and „next installiert“ [19]

3.2.3. Namenskonvention

Eine standardisierte, sinnvolle Namenskonvention für CIs lässt das Konfigurations-Management einfach zu und stellt sicher, dass nicht unbeabsichtigt Duplikate gebildet werden. Das Komponenten-Name und -Nummer Schema wird durch spezifische automatisierte Tools benutzt, um das Konfigurations-Management zu unterstützen.[7]

Die Namenskonvention sollte Code ergeben, der das System der Komponenten anzeigt, den Typ der Komponenten, die hierarchische Unterteilung des CI und, wo mehrfache Versionen bestehen, einen Version discriminator. Das empfohlene Format für CI Codes ist
sss.ccc[.ccc..](vvv)
sss = System identifizier
ccc = Komponenten Spezifikationssymbol
vvv = Version discriminator [7]

System identifizier: Das System ist der Bestandteil des höchsten Niveaus, das unter dem Konfigurations-Management betrachtet werden kann. Ein einzigartiger Systemname sollte als Präfix jedes CI innerhalb des Systems definiert und verwendet werden.[7]

Komponenten Spezifikationssymbol. Das Spezifikationssymbol wird verwendet, um zwischen Komponenten des gleichen Typs für ein System zu unterscheiden und wird im Allgemeinen eine hierarchische Struktur der Systeme und der Unterbaugruppen reflektieren. Im Format sollten so viele Spezifikationssymbole verwendet werden, wie erforderlich sind.[7]

Version: Die Version kann verwendet werden um die spezifische Versionsnummer zu reflektieren, welche die Komponenten identifiziert. Für kommerzielle Komponenten sollte die Version die Verkäuferversion oder Release Nummer widerspiegeln, z.B. r1.1a , oder v6.2. Die meisten CIs haben die Verkäufer Release version, die als Konfigurationseintragungsattribut anstatt als Name-CI kodiert wird.[7]

Szenario
Namenskonventionen und DNS für die Netzeinrichtung helfen beim Management des Extranets. Die meisten Netzeinrichtungen haben eine oder zwei Schnittstellen, um die Netzeinrichtung zu verwalten. Wir sollten für diese Schnittstelle mit Netzeinrichtungs-Typ, Lokalisation und Schnittstellen-Typ eine Namenskonvention erzeugen. Wir empfehlen auch, einen DHCP Bereich einschließlich, der die Positionen der Benutzer identifiziert und sie dem DNS hinzuzufügen. Dieser kann ein Teil der IP Adresse oder der physischen Position sein. Ein Beispiel könnte lauten: "dhcp-bldg-c21-10", "dhcp-bldg-c21-253", das IP Adressen in Gebäude C identifiziert, zweiter Stock, Kabine 1. Man könnte auch das exakte Subnetz für die Identifikation verwenden. Sobald eine Namenskonvention für die Netzeinrichtung und DHCP erzeugt wurde, benötigt man Tools zum Aufspüren und für den Zugang zum Management.

3.2.3. Labeln von CIs
Alle CIs sollten mit ihrem Konfigurations-Hauptlistennamen beschriftet werden, und Änderungen an den Konfigurations-Komponenten sollten durch Updates an den Aufklebern der CIs reflektiert werden. Physikalische Label sollten an der Hardware und den Komponenten angebracht werden, und in den Quellcode Modulen sollten Anmerkungs-Header benutzt werden, um Software auszuzeichnen.[19]

3.3. Konfigurations-Überwachung

Konfigurations-Überwachung stellt vom Empfang bis zur Beseitigung sicher, dass nur autorisierte und identifizierbare CIs angenommen und notiert werden, sie stellt sicher, dass kein CI addiert, geändert, ersetzt oder entfernt wird, ohne passende steuernde Komponenten. Z.B. fordert eine anerkannte Änderung eine aktualisierte Spezifikation an. Die Aktivitäten unter der Konfigurations-Überwachung sind Folgende:

3.3.1. Neue CIs und Versionen registrieren
Der Prozess der Registrierung fängt mit der Bestellung oder Entwickelung des Items an. Einige Organisationen verwenden ihren eigenen Beschaffungsprozess, um sicherzugehen, dass bestellte CIs addiert werden. Lieferanten tragen möglicherweise auch zur Meldung bei, indem sie die CIs vorher labeln. Auf diese Weise funktioniert die Zuordnung und die Anlieferung von CIs unter der Configuration Management Überwachung.[19]

3.3.2. Software entwickeln

Für die Software ist der Punkt des Empfangs (’receipt’) normalerweise der, an dem die Software zum Einsatz bereit ist. Eine DSL wird empfohlen, in der alle Software CIs und ihre Komponenten in ihrer endgültigen, auf Qualität kontrollierten Zustand enthalten sind. Beim Registrierungs-Prozess sollte man sichergehen, dass Details aller autorisierten Software und Komponenten CIs in die CMDB eingetragen sind, bevor die CIs von der Entwicklungs-Bibliothek in die DSL gebracht werden. Der Status des CIs sollte geändert werden, wenn sie in der DSL angemeldet werden. Idealerweise sollten das Dienstprogramm oder das Support-Tool das CMDB Update automatisch durchführen.[19]

3.3.3. Standard CIs

Das Change Management sollte sicherstellen, dass alle neues CIs als richtig registriert im CMDB autorisiert werden, bevor sie geliefert werden und, dass der Status dieser CIs geändert wird, während sie geliefert, angebracht, geprüft und angenommen werden. Eine Überwachung sollte durchgeführt werden, dass gelieferte CIs autorisiert werden. Der Installationsvorgang sollte nicht beginnen, bis diese Überwachung zufriedenstellend durchgeführt worden ist.[19]

3.3.4. Neue CIs und Versionen von building und releasing

Gute Build und Release Überwachung stellt sicher, dass aktualisierte Versionen von Soft- und Hardware richtig aufgebaut und verteilt werden, um mit dem Release kompatibel zu sein. Configuration Management mit Release Management sollte die Versionen der Software, Hardware und Komponenten als das Resultat des Build und Release Prozesses dokumentieren und berichten. Besondere Vorsicht sollte sollte darauf verwandt werden, sicherzugehen, dass Software nicht während der Kopie- oder Verteilungsprozesse korrumpiert oder geändert wird. [19]

3.3.5. Update von CIs

Der Status von CIs ändert sich, wenn sie von der Anlieferung zum Live-Einsatz kommen. Idealerweise sollte die CMDB bezüglich des Status der CIs- und Release-Änderungen automatisch geupdated werden. Dazugehörige Komponenten, wie Test-Zertifikat und Lizenzen, sollten in eine überwachende Dokument Library abgelegt werden.

Veränderungen an den Attributen von CIs in der CMDB sollten mit einem in Verbindung stehenden RFC aktualisiert werden.

Um sicherzustellen, dass alle IT Infrastruktur items durch das Change Management autorisiert werden, wird ein Report aller autorisierten Changes und Verbesserungen in der CMDB angelegt. Sofern eine Veränderung implementiert ist, sollte die CMDB dahingehend geändert werden, um die Statusänderung von CIs zu reflektieren. [19]

3.3.6. Lizenz Überwachung

Es ist folglich besonders wichtig, unabhängig davon, wer die Implementierung durchführt, von der die CMDB mit Details aktualisiert wird, wer Kopien der Software Nutzungsbedingungen hat. Dies unterstützt die Organisation, ihre legalen Verpflichtungen einzuhalten, und es hilft Revisoren und dem Service Desk, nicht autorisierte Kopien zu überprüfen. [19]

3.3.7. Update und Archivierung der Konfigurations-Reporte von nicht benutzten CIs

Der Abbau und die Kontrolle der Beseitigung von CIs ist häufig aus finanziellen und Sicherheitsgründen wichtig. Es sollte für stillgelegte Geräte oder Software Stellen geben, um die korrekte Disposition der Vermögen der Organisation sicherzustellen und damit die relevanten Berichte aktualisiert werden. Die CMDB sollte aktualisiert, und der Status des CIs zum abschließenden Zustand geändert werden, z.B. 'withdrawn' oder ‚archived'[19]

3.3.8. Integrität der Konfigurationen beschützen

Der Prozess für Beschaffung, Ablage, Abfertigung, den Empfang und Beseitigung von Waren sollte sicherstellen, dass Hardware, Software und Komponenten sicher an ihren Bestimmungsort geliefert werden. Speicherbereiche sollten sicher sein. Überwachung der Anlieferungskomponenten und Waren, die in die Organisation kommen, sollten durchgeführt und dokumentiert werden. Installations-, Umgebung- und elektrische Überwachung sollten von den passenden Leuten vor Anschluss ans Netz geplant und durchgeführt werden. Zugriffsüberwachungen und –Beschränkungen sollten definiert und erzwungen werden, um die Mitarbeiter auf dem korrekten Niveau mit dem Zugang zur Configuration Management Datenbank, der physikalischen Hardware, Software und den Komponenten zu haben. [19]

Das Configuration Management sollte die Integrität der gespeicherten Software CIs, ungeachtet des Mediums oder der Library sicherstellen, durch:

- Die Auswahl eines geeigneten Speichermediums, um Regenerationsstörungen oder –Verschlechterungen zu minimieren,
- Erneuerung der archivierten CIs in einer Frequenz kompatibel mit der Lagerfähigkeit des Mediums.
- Speicherung von Duplikaten in kontrollierten Positionen, um die Gefahr des Verlustes im Falle eines Unfalls zu minimieren. [19]

3.3.9. Update der CMDB

CM sollte den Ursprung jedes nicht registrierten Items verfolgen und das CI registrieren, um ein Problem zu beheben oder eine Löschung einzuleiten. Die nicht registrierten Items sollten vom Management behoben und berichtet werden. Die empfindliche Handhabung kann erforderlich werden, um einen ‚Schwarzmarkt' von nicht registrierten CIs zu vermeiden. (z.B. Software aus dem Internet) [19]

Szenario:

Nach dem Start lässt man über die Prozess Komponenten des Extranets einem Algorithmus laufen, um die Zahl und die Arten der Nodes in seinem Bereich zu überwachen, die Zahl und Arten der Kommunikationsverbindungen festzulegen und, um die Schlüsselinformation der Nodes, wie Kapazität, zu berichten. Für einige Prozess Komponenten des Extranets müssen solche Informationen per Hand gefunden werden und eingegeben werden. Zusätzlich muss die Route „Kosten " bekannt gegeben werden.

Die Konfigurations-Vollständigkeits-Überprüfung sollte die gesamte Konfiguration des Extranets, seine Kompliziertheit und Übereinstimmungen, sowie mögliche Probleme auswerten. Z.B. verwenden wir das Netsys configuration validation tool. Dieses Tool gibt alle Netzeinrichtungs Konfigurationen ein und erzeugt einen Konfigurations-Report, der gegenwärtige Probleme wie doppelte IP Adressen, Protokoll Fehlanpassungen und Widersprüche identifiziert. Das Tool berichtet über alle möglichen Konnektivitäts- oder Protokoll-Probleme, gibt aber nicht Standard-Konfigurationen für die Auswertung auf jedem

Gerät vor. Man kann Configuration Standards manuell wiederholen oder einen Skript erzeugen, um Unterschiede zu dieser Standard Konfiguration zu dokumentieren.

Das Extranet der BMW AG ändert sich selbstverständlich über die Zeit hinweg. Prozess-Komponenten empfangen automatisch eine Mitteilung von einem anderen Node in ihrem Bereich über eine Veränderung (Change). In einigen Fällen muss die Prozess-Komponente die Veränderung automatisch entdecken, wie etwa eine Störung in einem Node. In anderen Fällen muss die Information mittels manueller Intervention gegeben weden. Wenn die Prozess-Komponente selbst geändert wurde, muss der Change an hierarchisch höhere Prozess-Komponenten geschickt werden.

3.4. Configuration Status-Nachweise

Die Identifikations-Prozesse, welche berichten und die Change-Überwachung, die oben beschrieben werden, können sehr große Datenmengen erzeugen. Es wäre jedoch recht sinnlos, wenn diese Daten nicht analysiert würden. Status-Nachweise sind die Nachrichtenbeschaffung und Berichte aller gegenwärtigen und historischen Daten, die jedes CI während seines Lebenszyklus betroffen haben. Dieses ermöglicht Veränderungen der CIs nachweisbar zu machen, etwa spürt man den Status eines CIs auf, während es von einem Zustand in einen anderen wechselt, z.B. “being tested“ (Testphase), “live“ (implementiert) oder „withdrawn“ (zurückgezogen).

Ein CI besteht aus einer Menge von unterschiedlichen Zuständen. Mögliche Zustände umfassen:

- *geplant* - das CI wird für die Zukunft geplant, existiert jedoch noch nicht. Ein funktionsfähiges Konzeptdokument beschreibt die erwartete Leistung des CI;
- *In-Entwicklung* - das CI existiert, jedoch ist es noch nicht in seinem endgültigen Zustand. Die Konfiguration des CIs entwickelt sich noch. Eine Spezifikation der funktionellen Konstruktion könnte das CI beschreiben;
- *geprüft* - die Entwicklung des CIs gilt als komplett, und alle Tests sind durchgeführt worden. Ein kompletter Test-Bericht und Designspezifikation sollten vorhanden sein, um das CI zu beschreiben. Das CI muss zur Implementierung mit der lokalen baseline Prozedur übereinstimmen.
- *implementiert* - ein CI ist implementiert worden;
- *unaktiviert* - das CI ist fertig, aber ist noch nicht in den Service gesetzt worden oder wird als Teil eines Unfall-Backup-Plans gehalten. Obwohl dieser Zustand ursprünglich für Hardware CIs gedacht ist, kann dieser Status in gleicher Weise für ältere Versionen von Software zutreffen, die als "backup" vorliegen und die noch nicht archiviert worden sind.
- *archiviert* - ein CI, das außer Betrieb ist, dessen Reference aber noch existiert. [7]

Szenario: Aktuelle Netzeinrichtung, Verbindung und Endbenutzerlagerinformationen - Status-Nachweise
Aktuelle Netzeinrichtung, Verbindung und Endbenutzerlagerbestandinformationen ermöglichen es, den Netzlagerbestand und Ressourcen, Problem-Auswirkungen und Netz-Change-Auswirkungen aufzuspüren. Die Erkennung der Beziehung zu den Benutzeranforderungen, mit Hilfe des Netzlagerbestandes und der Ressourcen, stellt sicher, dass die Netzeinrichtung aktiv benutzt werden kann, die für ein Audit benötigten Informationen zur Verfügung gestellt werden können und die Netzeinrichtungs-Ressourcen dem Management berichtet werden. Endbenutzer-Beziehungs-Daten liefern Informationen, um die Gefahr bei Veränderungen und deren Auswirkung zu definieren, sowie die Fähigkeit zu schneller Problem-Überprüfung und -Behebung. Netzeinrichtungen, Verbindungen und Endbenutzerlagerbestand Datenbanken werden gewöhnlich durch viele führende Dienstleistungsorganisationen entwickelt. Der führende Entwickler von Netzlagerbestands-Software ist die Corporation. Die Datenbank kann Tabellen für Dinge wie Netzeinrichtung, Verbindungen und Daten des Kunden user/server enthalten, damit man, falls eine Netzeinrichtung nicht mehr benutzt wird, oder Netz-Changes auftreten, die Endbenutzerauswirkung leicht verstehen kann.

3.5. Verifikation und Audit

Die Konfigurations-Verifikation und das Audit bestehen aus der physikalischen Verifikation von allen CIs. Audits werden eingeleitet, um sicherzugehen, dass das in der CMDB notierte CIs auch wirklich in der physikalischen Umgebung vorhanden sind. Sie werden auch benutzt, um den Inhalt der DSL zu überprüfen oder irgendwelcher anderer sicherer Dokument-Libraries, welche die Organisation haben könnte. Zusätzlich helfen sie, zu überprüfen, dass nur autorisierte Softwarebausteine und Verfahren in der Umgebung vorkommen. Ebenfalls sollte überprüft werden, dass passendes RFCs für alle anerkannten CIs eingeordnet wurden. Solche Überprüfungen stellen sicher, dass die CMDB aktuell bleibt und folglich eine zuverlässige Informations-Quelle für die Organisation ist.

Wenn nicht autorisierte items in der Konfigurations-Umgebung gefunden werden, sollte dies dem betreffenden Manager mitgeteilt werden. Korrektur-Maßnahmen werden angefordert und, falls notwendig, sollte eine Schulung, gemacht werden. Bei Entdeckung illegaler Software z.B., sollten Maßnahmen ergriffen werden, um die Software zu entfernen, und eine Schulung zur Verfügung gestellt werden, um bei den Benutzern Verständnis aufkommen zu lassen, warum der Gebrauch illegaler Software ist schädlich ist und was die Konsequenzen für das Individuum und die Firma sein könnten. Der Prozess der Konfigurations-Verifikation und des Audits wird in der Abb.17 gezeigt:

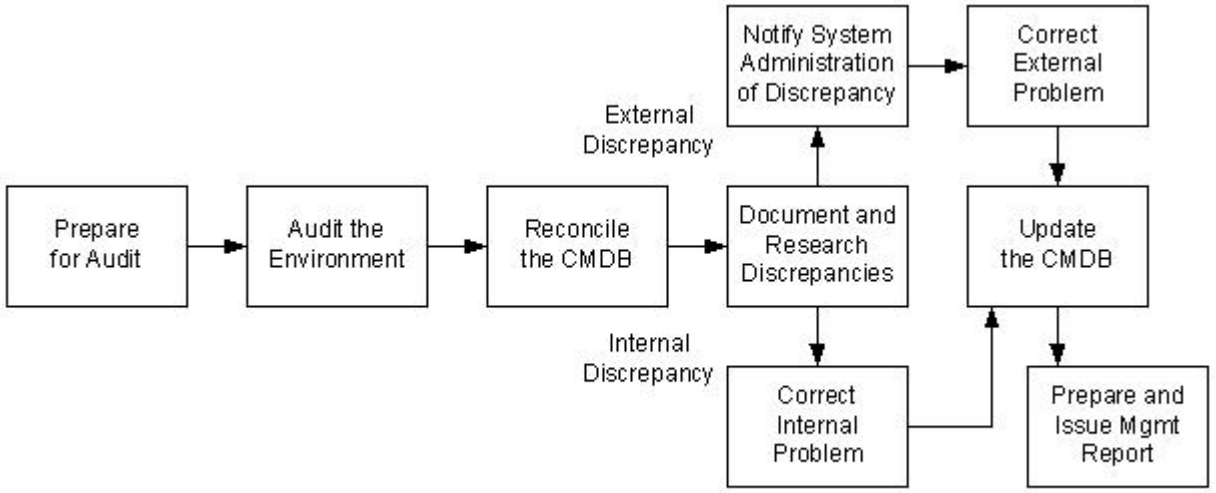

Abb.17: Configuration Audit Prozess [1]

Um ein Audit vorzubereiten, muss der Configuration Manager Konfigurationspersonal-Verantwortlichkeiten, wie die Überprüfung der Desktops, Server, Drucker, Kabel und so weiter übertragen und legt je nach Umgebung die Methode der Ausführung des Audits fest. Komponenten, die keinen Barcode enthalten, sollten nach dem tatsächlichen Audit dokumentiert und Nachforschungen angestellt werden. Der Abgleich („Reconciling“) der Daten in der CMDB umfasst, die Resultate des Audits mit dem tatsächlichen CMDB Inhalt zu vergleichen. Um menschliche Fehler zu vermeiden, sollte ein automatisierter Prozess für diese Aufgabe verwendet werden. Ein Report sollte entwickelt werden, der alle möglichen Diskrepanzen verzeichnet. Die Konfigurations-Mitarbeiter sind für die Untersuchung jeder Diskrepanz verantwortlich, um festzustellen ob das Problem aus einem Fehler des Configuration Managements oder aus einer externen Quelle resultiert. Interne Probleme sollten behoben werden, indem die Mitarbeiter die Aufgaben überprüfen, um festzustellen, ob der Prozess umstrukturiert werden muss, um weitere Probleme zu verhindern. Bei Bedarf sollte eine Schulung geleitet werden, um diese Aufgabe zu verstärken. Externe Probleme können das Resultat der Benutzer sein, die nicht autorisierte Software downloaden oder nicht genehmigte Hardware verwenden. Diese Fälle sollten der System Administration berichtet werden, und diese Gruppe sollte das Problem korrigieren und die Resultate wiederum dem Configuration Management berichten. Die abschließende Aufgabe des Konfigurationsmanagement Mitarbeiters ist, einen Bericht vorzubereiten, der die Audit Resultate, alle Diskrepanzen und die ergriffenen Korrektur-Maßnahmen dokumentiert. Berichte sollten den Configuration-Managern zur Verfügung gestellt werden, damit sie alle mögliche Probleme berücksichtigen können, die in Bezug zu den nicht autorisierten Gebrauch von items und den ergriffenen Korrektur-Maßnahmen stehen.[20]

Szenario:
Netzeinrichtungs-, Protokoll- und Medien-Audits sind Leistungsindikatoren für Übereinstimmungen in den Software-Versionen, Hardware-Netzeinrichtungen, Modulen, Protokollen, sowie in den Medien und in der Namenskonvention des Extranets. Die Audits sollten alle Nicht-Standard-Issues, welche Konfigurations-Updates erfordern oder die Issue verbessern , zuerst identifizieren. Die gesamten Prozesse müssen ausgewertet werden, um festzustellen, wie die suboptimalen oder Nicht-Standard-Entwicklungen verhindern werden können. Cisco RME ist ein Konfigurations-Instrument, das über Module und Software-Versionen revidieren und berichten kann.

4) Configuration Management Prozess

Wie ein Configuration Management Prozessmodul aussieht, und wie es in unserem Szenario aussehen könnte, kann man auf den folgenden Diagrammen (Abb18 und 19) sehen.

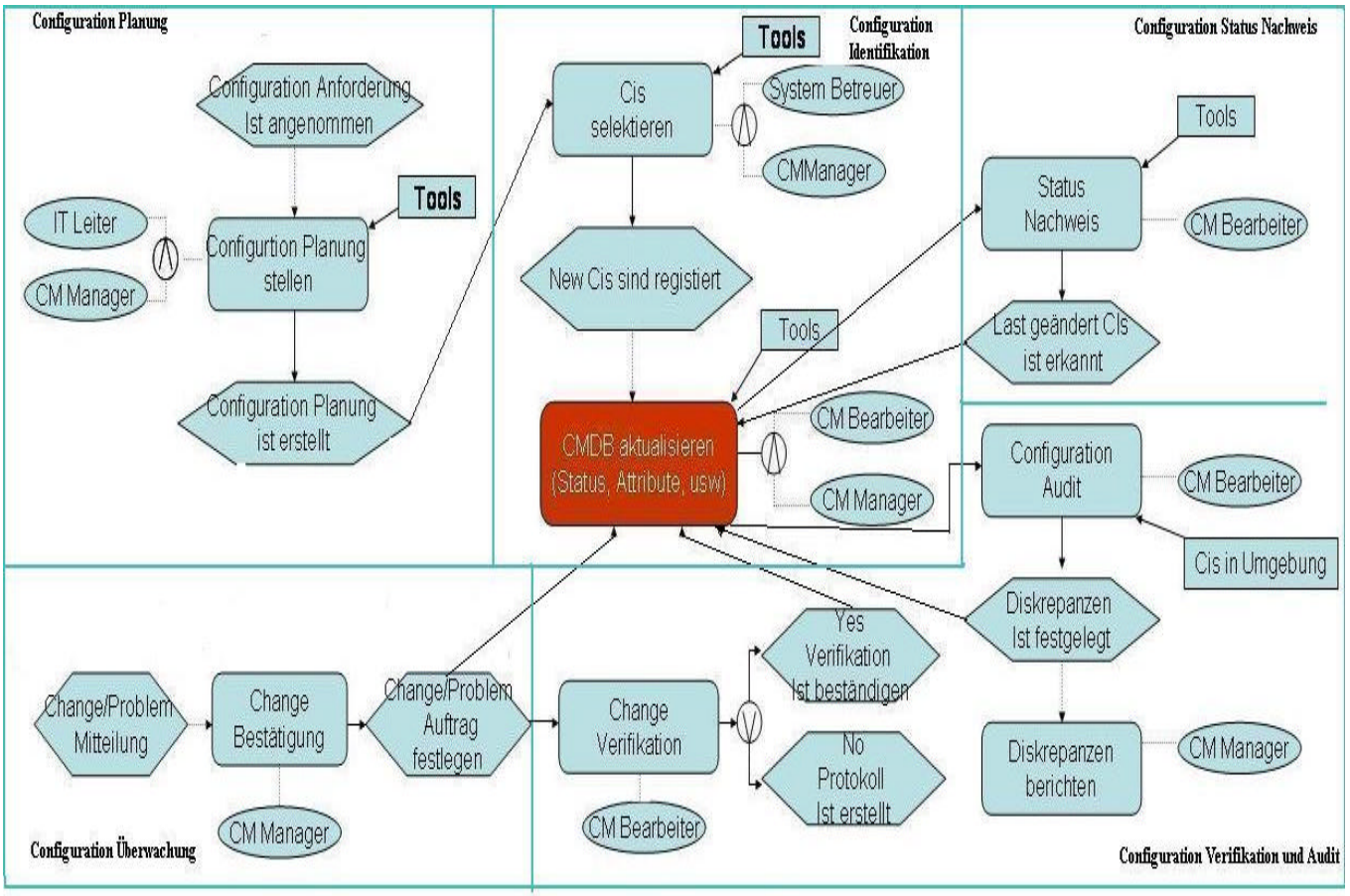

Abb.18. Configuration Prozessmodul

Abb19 ist Prozessmodul für BMW Extranet

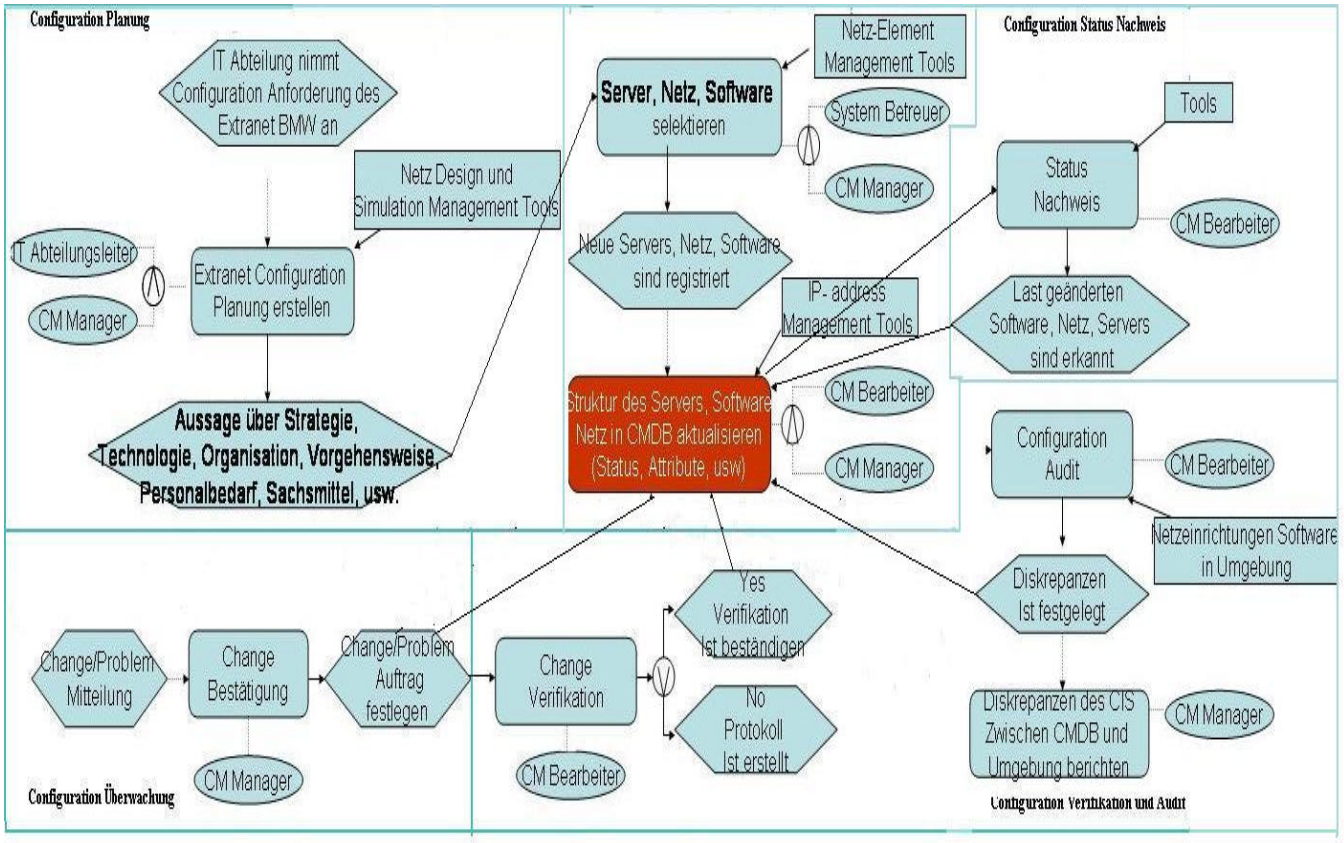

Abb.19: Configuration Prozessmodul für BWM Extranet

5) Configuration management Tools

Wenn man ein proaktives Herangehen an das Configuration Management erwägt, steht das IT Management vor der Entscheidung, eine Menge vorhandener Werkzeuge zu integrieren, oder eigene Tools aufzubauen. Wegen der hohen Schwierigkeit, eigene Configuration Tools aufzubauen, gilt dies nur für sehr große Unternehmen. Für das Extranet der BMW AG entschließen wir uns, die vorhandenen proaktiven Configuration Management Tools in die Umgebung zu integrieren. Es gibt auf dem heutigen Markt vier Kategorien zur Auswahl. [15]

Netz-Element Management Tools:
Helfen uns, Die Netzeinrichtungen von spezifischen Lieferanten zu konfigurieren und zu überwachen (CiscoWorks2000,Nortel Optivity, usw.)

Netz Design und Simulation Management Tools:
Helfen uns bei der Planung der Netztopologie, der Kapazität und der Verhaltenvorhersage (OpNet Modeler, Compuware Vantage Predictor und dergleichen)

IP- address Management Tools:
Hilfe beim Management der Adressen, zur Vermeidung von Duplikaten in einer Umgebung (Lucent VitalQIP, Cisco Network and Service Registrar, usw.)

Multi-Lieferanten Konfiguration 'point Suite' Tools:
Adressieren einen sehr spezifischen Aspekt des Configuration Managements im Zusammenhang mit Einfluss auf eine andere FCAPS Disziplin, wie Sicherheit. Zum Beispiel (AlterPoint, Gold Wire, usw. oben verzeichnet)

Integration dieser Tools in die IT Umgebung kann einen relativ hohe Proaktivität des Konfiguration Managements bewirken, aber es erfordert mehr als die gelegentliche Einbeziehung von IT Personal. [15]

6) Zusammenfassung

Das Configuration Management stellt gesicherte und aktuelle Informationen über die zur Leistungserstellung verwendeten Konfigurationselemente zur Verfügung. Dies geschieht durch das Identifizieren, Kontrollieren, Pflegen und Verifizieren der Versionen aller Konfigurationselemente (Configuration Items - CI). Jedes Betriebsmittel und die daraus resultierenden IT-Services bilden ein Konfigurationselement. Ein CI hat eine Kategorie, Attribute, Relationen, einen Status und bekommt eine eindeutige Referenznummer. Ein grundlegendes Werkzeug zur Erfassung und Präsentation dieser CIs und ihrer Beziehungen zueinander, bildet die Configuration Management Database (CMDB).[6]

7) Literaturverzeichnis

1. *Operations Architecture GuideMicrosoft Solution for Systems Architecture: Internet Data center*, 2002 http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/idc/oag/oagc03.asp

2. microTool GmbH: *Version and Configuration Management with Case /4/0.* MicroTool GmbH, 1998.

3. Cisco System: *Cisco-Configuration Management: Best Practices White Paper.* Cisco System, 1992.

4. ITIL: *Configuration Management nach ITIL*. http://www.kess-dv.de/StandardsUndMethoden/ITIL/ServiceSupport/ConfigurationM/body_configurationm.html

5. E.D.J. Plante: *The Process Documentation Supporting the Department of National Defence IM Configurationand Change Management Framework.* © Her majesty the Queen in right of Canada as represented by the Minister of National Defence, 2001. http://www.itilworld.com/europe/case-studies.htm

6. Rest Roland Lenz: *Konfiguration Management.* 18.Juni.2003, http://www.2cool4u.ch

7. *NZSIT105 - CONFIGURATION MANAGEMENT.* June 1994 http://www.gcsb.govt.nz/nzsit/105/105index.htm

8. *Configuration Management Database.* Exagon Consulting & Solutions GmbH, Copyrights@1994-2003, http://www.exagon.de.

9. Thomas Salvador: *Konfiguration Management.* Fernuniversitaet-Gesamthochschule-Hagen, Juli 1997.

10. Zhuo Wang Information Net GmbH: *IT Service Management.* Copyrights 2003, http://www.aspire-tech.com/aspire/v2/tongxun/577/853.html

11. Zhuo Tianzhu, Liu Wei: *Konfiguration Management*. Nov. 2003 http://media.ccidnet.com/media/ccu/585/04301.htm

12. ITIL Syllabus: IT Service Management (Service Support)Configuration Management, http://www.dv-werk.de/ger/itil/syllabus/com.html

13. Theo de Wit: IT Service Management Improve Business Value with ITIL. Quint Wellington RedWood, 2002, http://www.quintacademy.com

14. Ken Stainsby: *Practical Application of ITIL Best Practices.* http://www.nashco.ca

15. Dennis Drogseth: Implementing Network Configuration Management. Network world, http://www.nwfusion.com

16. Axept AG, http://www.axept.ch

17. ITIL Syllabus: IT Service Management (Service Support) Configuration Management http://www.dv-werk.de/ger/itil/syllabus/com.html

18. Configuration Management. Infra Corporation, @2003, http://www.infra.com.au/Solutions/ConfigurationMgnt.asp

19. *Service Support* Configuration Management

20 Sailer, M., *Klassifizierung und Bewertung von VPN–Lösungen für die Neuausrichtung der europaweiten Extranetstrategie der BMW AG*, August, 2002