Heterogenes Softwarekonfigurationsmanagement mit UNIX*

Matthias Kranz
mskranz@acm.org
ASDIS Software AG, Berlin

19. Februar 2003

Zusammenfassung

Wie stellen IT-Verantwortliche in Banken sicher, dass am 01.01.2002 um 00:01 Uhr auf allen Geldautomaten der Euro das Sagen und die DM ausgedient hat? Wie erreicht man, dass Tausende von SAP-Anwendern am Montag Morgen auf einen SAP-Client zugreifen, der auf die am Wochenende neu installierte Version der Datenbank und des Anwendungsservers abgestimmt ist?

Softwarekonfigurationsmanagement ist ein feststehender Begriff in großen bzw. komplexen Netzen. Wenn Applikationen, Konfigurationen oder Daten zu einem bestimmten Zeitpunkt und an mehreren Orten nachvollziehbar installiert werden müssen, ist die Administration auf die Unterstützung durch ein entsprechendes Managementwerkzeug angewiesen.

Das Paper gibt einen Überblick über die allgemeinen und speziellen Anforderungen an ein Managementtool und bietet eine Zusammenfassung aktueller Technologien. Es wird dargestellt, welche unterschiedlichen Objekte und Zustände im Softwaremanagement berücksichtigt werden müssen. Dazu zählen die zu adressierenden Benutzer, Systeme und Systemgruppen ebenso wie die Verteilobjekte und ihre Zielplattformen, die abstrakte Zusammenfassung von Konfigurationen zu Profilen, die Berücksichtigung von Abhängigkeiten zwischen Objekten und deren Parametrisierung. Außerdem fordern die unterschiedlichen Betriebssystemplattformen in einem heterogenen Netz eine Unterstützung verschiedenster Paketformate wie RPM, DEB, PKG, LPP etc. unter GNU/Linux und UNIX sowie MSI, InstallShield und andere unter Microsoft Windows. Bei der eigentlichen Datenübertragung werden die alternativen Ansätze Push und Pull, Broad- und Multicast sowie Bandbreitenmanagement vorgestellt. Abgeschlossen wird die technische Betrachtung durch die zu berücksichtigenden Sicherheitsaspekte.

In der Zusammenfassung werden, ergänzend zu einer Bewertung der vorgestellten Aspekte, Aufwände und Nutzen einer Einführung bzw. Anwendung von Softwarekonfigurationsmanagementwerkzeugen gegeneinander abgewogen.

1 Einleitung

Die allgemeinen Anforderungen an Werkzeuge zum Softwarekonfigurationsmanagement lassen sich anschaulich mit Ausdrucksmitteln objektorientierter Konzepte be-

*Die im Text verwendeten Markennamen der jeweiligen Firmen unterliegen leider im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz.

schreiben. Auf der einen Seite gibt es die zu adressierenden Objekte, also die potentiellen Ziele einer Softwareverteilung. Dazu gehören Systeme, Systemgruppen, Benutzer und Benutzergruppen. Demgegenüber stehen Verteilobjekte (z.B. ein Softwarepaket) und Gruppen von Verteilobjekten. Die genannten Objekte können unterschiedliche Zustände annehmen, Attribute und Methoden besitzen. Der Zustand einer Verteileinheit kann beispielsweise Entwicklung, Integration, Abnahme oder Produktion sein. Als Attribut z.B. sind die Größe, die Versions- und Revisionsnummer, unterstützte Zielplattformen und Abhängigkeiten denkbar. Die Methoden ergeben sich aus den Aktionen, die auf eine Verteileinheit anwendbar sind, etwa die Installation, Deinstallation, Aktivierung und Konfiguration.

Ein Softwaremanagement Tool stellt also im Wesentlichen ein Framework dar, dass die genannten Objekte und ihre Zustände verwaltet und gleichzeitig Funktionalität bereitstellt, um diese zu verändern. Viele Werkzeuge bilden außerdem Funktionen aus anderen IT-Management-Bereichen ab, um die Kernprozesse einfacher zu gestalten. Dazu zählen unter anderem Inventar- und Lizenzmanagement. Durch eine integrierte Inventarisierung von Hard- und Software entsteht die Möglichkeit, auf Basis von Abfragen und Bedingungen statische oder dynamische Zielgruppen zu definieren. Ergänzend entstehen neue Attribute, die nicht nur Abhängigkeiten zwischen Verteilobjekten sondern auch zwischen Verteilobjekt und Zielplattform erlauben. Mit Hilfe eines Lizenzmanagements kann zudem verhindert werden, dass Lizenzpotentiale ungenutzt bleiben oder möglicherweise überreizt werden.

1.1 ITIL

Heute lässt sich Softwareverteilung beziehungsweise Softwarekonfigurationsmanagement in mittleren und großen Netzen nicht mehr ohne definierte Prozesse durchführen. Eine Definition solcher Prozesse findet sich in der IT Infrastructure Library (ITIL), dem Industriestandard für nachhaltiges IT Servicemanagement. Hauptsächlich sind die folgenden drei Bereiche betroffen: Durch das Release Management wird sicher gestellt, dass in Produktivumgebungen ausschließlich Software eingesetzt wird, die einen entsprechenden Test- und Autorisierungsprozess durchlaufen hat. Zur Minimierung von Risiken und Kosten ist ein weiterer Kernprozess, das Change Management, notwendig. In ihm werden alle Änderungen an Software und Konfigurationen verwaltet, überprüft und gesteuert. Und schließlich wird durch das Configuration Management noch der Bereich des Asset Management, also die IT-Vermögensverwaltung, abgedeckt.

1.2 Schnittstellen

Es ist klar, dass kein Produkt - auch kein Framework - in der Lage ist, sämtliche Anforderungen aus ITIL abdecken zu können. Um so mehr gewinnt das Vorhandensein von flexiblen Schnittstellen zur Steuerung und zum Informationsabgleich bei Werkzeugen zum Softwaremanagement an Bedeutung. Ein Command Line Interface stellt also unter UNIX ein Mindest-Kriterium dar, um beispielsweise periodische Prozesse durch Skriptsteuerung automatisieren zu können. Zur Einbindung in größere Prozesse müssen außerdem ein Application Programming Interface und Schnittstellen zum etwaigen Im- und Export von benötigten Daten bereit stehen. Häufig bietet das Lightweight Directory Access Protocol (LDAP) in heterogenen Netzwerken mit GNU/Linux, UNIX, Microsoft und Novell einen kleinsten gemeinsamen Nenner zur zusätzlichen Anbindung an einen Verzeichnisdienst.

2 Stand der Technik

Im Folgenden sollen einige Technologien und Eigenschaften vorgestellt werden, die von vielen aktuellen Produkten umgesetzt werden.

Softwaremanagement-Werkzeuge stellen Funktionalität zur automatisierten Verteilung und Installation bzw. Deinstallation von Software bereit. Dabei finden die unterschiedlichsten Verfahren Anwendung. Bei der Datenübertragung reicht das Spektrum vom Mounten von Laufwerken, über das File Transfer Protocol (FTP) und ähnliche Technologien bis hin zum Hypertext Transfer Protocol (HTTP). Dabei sind die jeweils inhärenten Schwierigkeiten und Einschränkungen zu beachten. So stellt HTTP zwar durch seine vermeintlich einfache Administrierbarkeit - gerade im Bereich Sicherheit - eine populäre Technologie dar, doch ist das Protokoll selbst eben ursprünglich nicht für die Übertragung sehr großer Datenmengen konzipiert worden. Noch immer können einige Produkte nicht auf Verbindungsab- oder unterbrüche reagieren, sondern müssen die gesamten Daten erneut senden. Zur Vermeidung redundanter Datenübertragungen wird im einfachsten Fall eine Speicherung der übermittelten Software am nächsten System durchgeführt und dieser Zustand quittiert. Um einen weiteren Auftrag mit dem gleichen Paket durchzuführen, müssen nun nur die Auftragsdaten (in aller Regel wenige kB) gesendet werden. Eine weitere eingesetzte Technologie ist das Byte Level Differencing. In diesem Fall werden ähnlich einem Protokoll wie rsync nur die Daten erneut übertragen, die sich geändert haben.

Noch eine Ebene tiefer werden neben Unicast- auch Multicast-Verbindungen unterstützt. Letzteres wird allerdings noch recht wenig in der Praxis eingesetzt - häufig ist mangelnde Unterstützung in der Infrastruktur der Grund. Produkte, die nur Point-to-Point-Verbindungen unterstützen, lösen Last- und Bandbreitenprobleme durch die Möglichkeit, mehrstufige Depot-Server einzusetzen - mit dem Nachteil eines verlängerten Kommunikationswegs und der Verwendung weiterer Hardwareressourcen. Vorteil dieser asynchronen Verteilung ist in aller Regel die Robustheit und Stabilität gegenüber Netzwerkproblemen.

2.1 Push und Pull

Push oder Pull - das wird häufig als Philosophiefrage verstanden. Dahinter verbirgt sich die Frage, ob der Versand von Aufträgen und Verteileinheiten von einer zentralen bzw. in einer Baumstruktur höher gelegenen Instanz an die Endsysteme erfolgt, oder ob die Endsysteme die Kommunikation aufbauen und die Daten herunterladen. Beide Varianten haben eine Reihe von Vor- und Nachteilen und wünschenswerterweise bringt ein Werkzeug beide Techniken zur freien Auswahl mit.

Ein wesentlicher Vorteil beim Push ist, dass der Server jederzeit die Kontrolle und Übersicht behält. Er kann also bei Bedarf Verbindungen auf- und abbauen, z.B. im Falle von Last- oder Bandbreitensteuerung. Allerdings ist dann auf dem Endsystem eine Server-Instanz für das verwendete Datenübertragungsprotokoll (z.B. SFTP oder HTTPS) und eine Zugangsberechtigung notwendig. Etwaige Benutzer und Passwörter müssen also sicher transportiert und verwaltet werden können.

Pull bietet umgekehrt die Möglichkeit, aus Sicht des Benutzers am Endsystem eine Verbindung zum günstigsten Zeitpunkt zu starten. In diesem Fall müssen sich die Clients beim Server authentifizieren - das Problem der Berechtigungen wird also „dezentralisiert“, aber nicht unbedingt weniger komplex. Ungünstig ist allerdings, wenn Clients in großen Netzen ständig ihre nächstgelegene Server-Instanz nach Aufträgen anfragen.

Beide Formen sind natürlich auch als Mischbetrieb möglich, d.h., dass Aufträge vom Server zum Client, der eigentliche Datenfransfer aber vom Client initiiert wird.

2.1.1 Autopull

Beim Autopull ist die Idee, dass Endsysteme automatisch Verteileinheiten beziehen. Die Auswahl dieser Software wird üblicherweise auf bestimmte Bereiche eingeschränkt. Produkte, die die Subskription von sogenannten Channels erlauben, stellen typischerweise einen Channel für Security-Updates zur Verfügung. Das Endsystem erfragt durch diesen Mechanismus regelmäßig alle im Channel erhältlichen - und nicht angewendeten - Patches.

3 Paketierung

Unter UNIX und GNU/Linux sind einige unterschiedliche Paketformate bekannt. Die Mehrzahl der Pakete ist allerdings traditionell so konzipiert, dass eine Installation ohne Benutzerinteraktion vorgesehen ist. Im Gegensatz zu Microsoft Windows wird hier seltener ein mehrstufiger Dialog und damit eine grafische Oberfläche vorausgesetzt. Diese Tatsache bietet häufig die Möglichkeit, Skriptwrapper für eine automatisierte Installation oder Deinstallation zu entwickeln.

Für die Plattform Microsoft Windows sind einige bekannte Werkzeuge zur Erstellung von Paketen zur unbedienten Installation verfügbar. Zum Beispiel vergleichen sie den Zustand eines Testrechners vor und nach der Installation und erstellen aus der Differenz ein Paket. Dieses Verfahren stößt natürlich dann an seine Grenzen, wenn der Testrechner z.B. schon Bibliotheken besitzt, die eigentlich während der Installation kopiert werden müssen. Eine andere Methode zeichnet alle während der manuellen Abarbeitung durchgeführten Aktionen (z.B. Mausklicks und Tastatureingaben) auf und spielt diese am Zielsystem dann ab. In diesem Fall neigen einige Werkzeuge dazu, Knöpfe oder Eingabefelder bei unterschiedlicher Lokalisation des Test- und des Zielrechners nicht mehr zu erkennen. Die bekannten InstallShield-Routinen erlauben die Aufzeichnung aller Eingaben und Klicks in eine Antwortdatei. In diesem Fall treten zwar keine Verwechslungen von Knöpfen auf, da die Methode ja in der Installationsroutine vorgesehen ist. Aber unter Umständen taucht auf dem Zielsystem ein Dialog auf, der auf dem Testsystem gar nicht abgefragt wurde, z.B. wenn nicht genügend Platz auf dem Ziellaufwerk vorhanden ist. Kurzum: Die Paketierung ist ein Spezialgebiet, das von der eigentlichen Softwareverteilung getrennt zu betrachten ist. Ein Softwaremanagement-Werkzeug sollte daher wenigstens die Möglichkeit bieten, andere Paketformate zu akzeptieren, falls ein integriertes Paketierungs-Tool fehlschlägt.

4 Abhängigkeiten

Software hat oft die leidige Eigenschaft, andere Software vorauszusetzen oder aber nur ohne Vorhandensein einer bestimmten Software sauber zu funktionieren. Viele Paketformate unter UNIX und GNU/Linux verwalten diese Information direkt im Paket selbst, d.h., ein Paket definiert Voraussetzungen, die erfüllt sein müssen. Da gerade unter GNU/Linux eine Vielzahl verschiedener Software mit gleicher Funktionalität vorhanden ist, müssen bei der Abbildung von etwaigen Abhängigkeiten sogenannte Meta-Pakete berücksichtigt werden. Diese definieren eine abstrakte Funktionalität eines Pa-

kets. So könnte z.B. ein bestimmtes Mailtool das Vorhandensein eines Mail Transfer Agents voraussetzen, ohne von einem bestimmten Produkt abhängig zu sein.

Desweiteren ergeben sich in der Praxis häufig systemübergreifende Abhängigkeiten. Eine Software könnte beispielsweise auf einem Gerät einen Datenbank- und auf einem anderen einen Web-Applikations-Server voraussetzen. Die Installation wird durch eine spezielle Client Software auf Hunderten von Desktops komplettiert. Während des Rollouts oder des Updates solch einer Software entstehen nun Abhängigkeiten in Form von Reihenfolgen und Funktionsfähigkeiten. Der Web-Applikations-Server kann erst installiert werden, wenn die neue Datenbank ausgerollt ist. Und die neuen Clients dürfen erst dann auf den Arbeitsplätzen der Endanwender installiert werden, wenn die vorherigen zwei Schritte erfolgreich durchgeführt wurden. Im Fehlerfall muss ein kompletter Rollback erfolgen und alle Teilschritte zurückgenommen werden. Diese Abhängigkeiten werden auch mit dem Begriff Multi-Tier-Abhängigkeiten bezeichnet.

Abhängigkeiten treten also innerhalb eines Management-Werkzeugs auf unterschiedlichen Ebenen und in unterschiedlicher Komplexität auf. Eine vollständige Abbildung ist nicht trivial und wird zur Zeit nur von wenigen Produkten geleistet.

5 Installation

Die Installation von Software ist natürlich der eigentliche Kernprozeß eines Softwaremanagement-Systems. Auf vielen Plattformen ist heute entsprechende Funktionalität zur Verwaltung von Software verhanden, die dann durch ein Tool genutzt werden kann. So definiert das Red Hat Package Management (RPM) nicht nur ein Paketformat, sondern sorgt für die komplette Verwaltung aller Zustände und stellt entsprechende Operationen für die Installation und Deinstallation zur Verfügung. Ähnliche Methoden sind auf fast jeder Betriebssystemplattform vorhanden. Um eine Software auf beliebig viele Endsysteme verteilen zu können, ist also zunächst eine geeignete Paketierung durchzuführen. Anschließend muss die Software termingerecht transferiert und dann eine entsprechende Aktion aufgerufen werden. In den meisten Fällen wird von der Softwaremanagement-Lösung dazu ein Prozeß auf dem Endsystem installiert, der mit Administrationsprivilegien läuft. Dieser wartet auf einkommende Aufträge und führt sie dann aus. Ähnlich wie in modernen Dateisystemen kann der gesamte Installationsprozeß durch eine Art Journaling zusätzlich stabilisiert werden. In diesem Fall wird eine Aktion - auch im Falle unerwarteter Fehler der Plattform selbst - sicher durchgeführt.

6 Sicherheit

Im Bereich Sicherheit sind zum Einen die unberechtigte Verteilung und Installation und zum Anderen die Integrität der Verteileinheiten zwei der wesentlichsten Kriterien. Ergänzend kommt natürlich die Verschlüsselung des eigentlichen Datentransports dazu. Hier stellt die Verwendung vom Secure Sockets Layer (SSL) einen gemeinsamen Nenner dar. Die Integrität der Daten wird zumeist über Zertifikate sicher gestelllt und eine Anbindung an eine Public Key Infrastructure (PKI) ist in aller Regel möglich. Es muss schließlich verhindert werden, dass auf der einen Seite kein Endsystem (oder Benutzer) unberechtigerweise in den Besitz von Software oder Daten gelangt und dass umgekehrt natürlich auch kein Angreifer manipulierte Aufträge oder Verteileinheiten senden kann.

Gerade in diesem Bereich stehen die Produkte erst am Anfang einer Lösung. Die Nachfrage wächst allerdings stark, da immer mehr IT-Bereiche nicht nur Softwaremanagement für die eigene Firma, sondern als Dienstleister für mehrere Kunden erbringen.

7 Zusammenfassung

In welchen Umfeldern lohnt es sich nun, über die Einführung einer Softwaremanagentlösung nachzudenken? Diese Frage läßt sich sicherlich nicht ohne weiteres pauschal beantworten, auch wenn einige Produkthersteller gerne sogenannten Return-of-Invest-Rechner einsetzen, die aus zwei oder drei Eingabeparametern ein vielversprechendes Ergebnis zaubern.

Grundsätzlich bietet sich ein Werkzeug zum Softwaremanagement in allen Bereichen an, in denen eine hohe Standardisierung der Zielplattform vorgegeben ist oder aber zukünftig erreicht werden soll. Außerdem kann die Anzahl der zu managenden Clients (Server und Desktops) und deren geographische Verteilung einen wesentlichen Ausschlag geben. Wenn zusätzlich die Menge der zu verteilenden Software- und/oder Konfigurationsdatenpakete und deren Verteilungsfrequenz entsprechend groß ist, spricht alles für eine Einführung. Die Frage, ob und wenn ja in welcher Zeit sich die Investition amortisiert, sollte von Fall zu Fall individuell berechnet werden. Dazu kann beispielsweise die Analyse eines bisher hauptsächlich manuell durchgeführten Rollouts einer Standardsoftware dienen. Der ermittelten Gesamtzeit und den damit verbundenen Kosten können dann Abschätzungen (oder natürlich konkrete Erfahrungswerte) für die Paketierung und den automatisierten Rollout gegenübergestellt werden. Zu beachten ist dabei selbstverständlich, dass Aufwände bei der Erstpaketierung bei späteren Versionen (Updates und Upgrades) häufig erheblich gesenkt werden können. Schwerer in Zahlen und Kosten zu erfassen ist dagegen das Potential in den Bereichen Qualitätssicherung und Security. Wieviele Fehler bei manuellen Installationen und Updates entstehen bzw. welche tatsächlichen Kosten durch verspätet eingespielte Security-Patches entstehen, ist nur durch langwierige und detaillierte Analysen möglich.

Integrierte Funktionalität zur Inventarisierung und evtl. zum Metering von Soft- und Hardwarekomponenten birgt zudem weiteres Potential zur mittel- und langfristigen Kostensenkung durch die Identifizierung wenig oder gar nicht genutzter Assets. Durch die automatische Installation im Hintergrund wird außerdem die Verfügbarkeit von Servern oder Desktops erheblich erhöht.

Die Auswahl eines konkreten Werkzeugs sollte über einen einfachen Feature- und Funktionsvergleich hinausgehen. Oberflächlich betrachtet bieten vielen Produkte eine ähnliche Funktionalität, verhalten sich dann aber in der Operationalisierung vollkommen unterschiedlich. Durch die Einführung einer unterstützenden Software wird schließlich ein sehr wichtiges Glied der gesamten IT-Management-Kette zumindest in großen Teilen automatisiert. Zuverlässigkeit und langfristige Investitionssicherheit sollten an dieser Stelle durchaus die eine oder andere fehlende Eigenschaft aufwiegen. Für den hier betrachteten Fall des Managements von heterogenen Netzen ist auf jeden Fall ein generisches Paketformat und eine Möglichkeit der Definition abstrakter, systemübergreifender Abhängigkeiten von Vorteil.

Prognosen von Analysten und Beratungsfirmen sagen übrigens voraus, dass sich der UNIX-Server-Markt im Laufe der nächsten vier bis sechs Jahre fast vollständig konsolidieren wird und GNU/Linux den gesamten heutigen Marktanteil proprietärer UNIXe vereinnahmen wird. Nimmt man die Prognosen für den wachsenden Anteil

von GNU/Linux-Desktop-Lösungen mit ins Kalkül, dürften die Produkthersteller gezwungen sein, über weitere integrative Funktionserweiterungen nachzudenken, die den speziellen Anforderungen dieser Systeme gerecht wird.

Literatur

[1] Dr. F. Ziegler: *Prozesse steuern die Servicequalität - Desktopmanagement nach dem ITIL-Standard*, NetworkWorld Germany, 2002

[2] M. Kranz: *Verteilung in komplexen heterogenen Umgebungen*, LANLine, 2/2003

[3] Aberdeen Group: *Software Management Solutions from Linux Suppliers: A Competitive Analysis*, Aberdeen Group, 2002

[4] OGC: *ITIL Online*, http://www.itil.co.uk, 2003