

# Configuration Management

Versionsverwaltung und Best Practices für Nagios

Marcus Fleige

<marcus.fleige@de.rhenus.com>

Nagios-Workshop 2008, Oldenburg

10.06.2008

## Marcus Fleige

- "localhost"
- beschäftigt bei der Rhenus Assets & Services
- (am Standort Holzwickede (Bang Boom Bang!))
- 4 Jahre Nagios

## Nagios bei der Rhenus

- 750 Hosts
- 124 Host Groups
- 7000 Services
- 6500 Service Escalations
- Einfach - Distributed - Einfach -> Distributed

# Agenda - Worum gehts?

- 1 Good to know - Hintergrundwissen
  - Was ist Configuration Management?
    - Offizieller Ansatz
    - "Pragmatischer Ansatz"
  - Was ist eine Versionsverwaltung?
  - Softwarelösungen
  - Subversion - Features und Benefits
  - Wie funktioniert Subversion?
  - SVN und Nagios?
  - Vokabular
- 2 Subversion in der Praxis
  - SVN-Repository anlegen
  - SVN Daily Use
- 3 Gedanken zur Nagios-Konfiguration
  - Konfiguration strukturieren
    - Grundsätzliches
    - Vorüberlegungen
  - Konfiguration entschlacken: Ballast auslagern
  - Best Practices
    - Vererbung von gruppierten Services
    - Distributed Monitoring: Konfiguration verteilen mit SVN
- 4 Fazit
  - Zusammenfassung
  - F & A

## 1 Good to know - Hintergrundwissen

- Was ist Configuration Management?
  - Offizieller Ansatz
  - "Pragmatischer Ansatz"
- Was ist eine Versionsverwaltung?
- Softwarelösungen
- Subversion - Features und Benefits
- Wie funktioniert Subversion?
- SVN und Nagios?
- Vokabular

## 2 Subversion in der Praxis

- SVN-Repository anlegen
- SVN Daily Use

## 3 Gedanken zur Nagios-Konfiguration

- Konfiguration strukturieren
  - Grundsätzliches
  - Vorüberlegungen
- Konfiguration entschlacken: Ballast auslagern
- Best Practices
  - Vererbung von gruppierten Services
  - Distributed Monitoring: Konfiguration verteilen mit SVN

## 4 Fazit

- Zusammenfassung
- F & A

# Aus dem Leben eines Nagios-Admins

Kommt das jemandem bekannt vor?

- “Hast du die Datei *auch* gerade editiert?”
- “Ich war das nicht! Das muß jemand anders geändert haben!”
- “Verdammt, das habe ich letzte Woche erst angepasst! Wie war das noch?”
- “Duuuu... Ich hab aus Versehen den Ordner da gelöscht...”

# Was ist Configuration Management?

## ISO 10007:2003-Definition

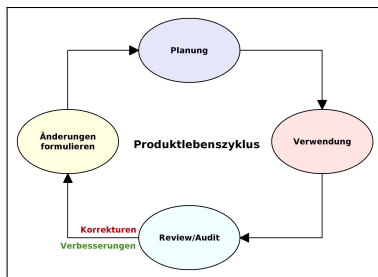
*CM ist eine Managementtätigkeit, die die technische und administrative Leitung des gesamten Produktlebenszyklus, der Konfigurationseinheiten des Produkts und der produktkonfigurations- bezogenen Angaben übernimmt. CM dokumentiert die Produktkonfiguration. CM sorgt für Kennzeichnung und Rückverfolgbarkeit des Produkts und hält den erreichten Stand seiner physischen und funktionellen Anforderungen sowie den Zugang zu genauen Informationen in allen Phasen seines Lebenszyklus bereit[...].*

...WTF?!

# “Pragmatischer Ansatz”

## Fakten:

- CM ist ein Managementbereich der ITIL.
- CM wird in der ISO 10007:2003 definiert.
- CM beschäftigt sich mit der Standardisierung von “Produkt-Lebensläufen”.



- CM ist zu abstrakt und hat direkt erstmal nichts mit der Praxis zu tun.
- Für die Praxis hat es einen Teilbereich: Versionsverwaltung.

# Was ist eine Versionsverwaltung?

- Version Control System (VCS) oder Source Code Management (SCM)
- Abbildung der CM-Grundgedanken auf eine unterstützende *Softwarelösung*

## Aufgaben:

- **Änderungen protokollieren** - wer hat wann was geändert?
- **Revisionssicherheit gewährleisten** - alte Stände wiederherstellen
- **Release-Stände archivieren** - als stabil markierte Stände vorhalten
- **Zugriffe koordinieren** - Berechtigungen für Bereiche vergeben
- **Kollisionen behandeln** - Unterschiedliche Versionen einer Datei
- **(Entwicklungszweige vorhalten** - unwichtig ;-))

# Softwarelösungen

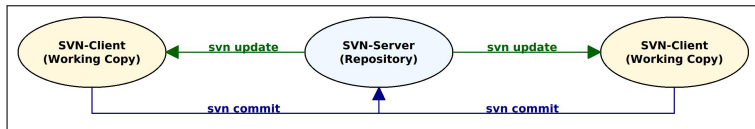
## Beispiele für VCS-Lösungen

- BitKeeper - kommerziell
- M\$ SourceSafe - kommerziell
- Revision Control System - OpenSource
- GIT - OpenSource
- Concurrent Versions System - OpenSource
- **Subversion** - OpenSource
- uvm.

# Subversion - Features und Benefits

- kostenlos, GPL-kompatible Lizenzen (Subversion License, Sleepycat License)
- große Support- und Entwicklungscommunity
- stabil und erprobt
- verschiedene Clients/GUIs für alle Betriebssysteme (z.B. esvn, kdesvn, NautilusSVN, TortoiseSVN)
- Plugins für Entwicklungsumgebungen (z.B. NetBeans, Eclipse, TYPO3 und M\$ Visual Studio)
- diverse Konverter für andere VCS

# Wie funktioniert Subversion?



# Warum lohnt sich SVN für die Nagioskonfiguration?

- **Client-Server-System** - Bearbeitung erfolgt auf lokalem System
- **Desaster Recovery** - Datei gelöscht? Konfig-Ordner weg? Kein Problem!
- **Dokumentation** - Änderungen werden protokolliert und zugeordnet.
- **Fehlersuche** - Vergleiche zwischen aktueller und älteren Versionen - z.B. bei Performanceeinbrüchen
- **Konflikte lösen** - Gleichzeitige Bearbeitung möglich

# Vokabular

- **Repository** - Zentraler Aufbewahrungsort eines Datendepots
- **Checkout** - Auslesen eines Repositories
- **svn update** - Befehl zum Aktualisieren der Arbeitskopie
- **svn commit** - Änderungen in ein Repository übertragen
- **Revision** - Eine bestimmte Version des Datenbestandes im Repository

- 1 Good to know - Hintergrundwissen
  - Was ist Configuration Management?
    - Offizieller Ansatz
    - "Pragmatischer Ansatz"
  - Was ist eine Versionsverwaltung?
  - Softwarelösungen
  - Subversion - Features und Benefits
  - Wie funktioniert Subversion?
  - SVN und Nagios?
  - Vokabular
- 2 Subversion in der Praxis
  - SVN-Repository anlegen
  - SVN Daily Use
- 3 Gedanken zur Nagios-Konfiguration
  - Konfiguration strukturieren
    - Grundsätzliches
    - Vorüberlegungen
  - Konfiguration entschlacken: Ballast auslagern
  - Best Practices
    - Vererbung von gruppierten Services
    - Distributed Monitoring: Konfiguration verteilen mit SVN
- 4 Fazit
  - Zusammenfassung
  - F & A

# SVN-Repository anlegen

```
$ svnadmin create /var/svn/repos/myproject
$
```

```
$ svn import /tmp/myproject file:///var/svn/repos/myproject -m "Initial Import."
Adding      /tmp/myproject/branches
Adding      /tmp/myproject/tags
Adding      /tmp/myproject/trunk
Adding      /tmp/myproject/trunk/foo.c
Adding      /tmp/myproject/trunk/bar.c
Adding      /tmp/myproject/trunk/Makefile
...
Committed revision 1.
$
```

```
$ svn checkout file:///var/svn/repos/myproject/trunk myproject
A myproject/foo.c
A myproject/bar.c
A myproject/Makefile
...
Checked out revision 1.
$
```

**Mal gucken...?!**

# SVN Daily Use

- **svn update** - Repository auf HEAD-Revision aktualisieren
- **svn status** - Status der Working Copy anzeigen
  - 'A' - Added
  - 'M' - Modified
  - 'D' - Deleted
  - '?' - Nicht unter Versionskontrolle
  - '!' - fehlt
  - uvm. - siehe `svn help status`
- **svn info <Datei>** - Allgemeine Infos (letzte Änderung, Autor, Revision...)
- **svn add|(rm|del) <Datei|Ordner>** - Datei hinzufügen|löschen
- **svn (mv|move) <Datei|Ordner>** - Datei verschieben
- **svn commit** - Änderungen ins Repository schreiben
- **svn revert** - Änderungen an der Working Copy zurücknehmen
- **svn blame|praise <Datei>** - Autoren/Editoren einer Datei anzeigen

- 1 Good to know - Hintergrundwissen
  - Was ist Configuration Management?
    - Offizieller Ansatz
    - "Pragmatischer Ansatz"
  - Was ist eine Versionsverwaltung?
  - Softwarelösungen
  - Subversion - Features und Benefits
  - Wie funktioniert Subversion?
  - SVN und Nagios?
  - Vokabular
- 2 Subversion in der Praxis
  - SVN-Repository anlegen
  - SVN Daily Use
- 3 Gedanken zur Nagios-Konfiguration
  - Konfiguration strukturieren
    - Grundsätzliches
    - Vorüberlegungen
  - Konfiguration entschlacken: Ballast auslagern
  - Best Practices
    - Vererbung von gruppierten Services
    - Distributed Monitoring: Konfiguration verteilen mit SVN
- 4 Fazit
  - Zusammenfassung
  - F & A

# Grundsätzliches

- Die Anforderungen an das Monitoring unterscheiden sich je nach Einsatzgebiet.
- Es ist unmöglich, verbindliche Richtlinien festzulegen.
- Es gibt trotzdem etwas Wesentliches zu beachten:

**Struktur ist gerade in größeren Umgebungen existenziell!**

# Vorüberlegungen

## Think about!

- **Was soll überwacht werden?** - Server,Router,Switches,Measures...
- **Wie detailliert soll überwacht werden?** - Erreichbarkeit,OS-Daten,Spez. Aufgaben...
- **Womit soll überwacht werden?** - NRPE,NC\_NET,SNMP,check\_tcp...
- **Wann soll überwacht werden?** - Servicezeiten, Service Level Agreements...
- **Wer muß wann benachrichtigt werden?** - Escalations ...
- **Wie ist die Netzwerkstruktur?** - Bandbreite,Firewalls,Kundenanbindungen...
- **Wo wird überwacht?** - lokal,remote,verteilt, ...
- **Was hat Priorität?** - Services,Dependencies,Escalations...
- **Sonderfälle und Ausnahmen**

# Beispiele für eine Ordner-/Dateinamen-Struktur

## Struktur-Hilfsmittel:

- Dateinamen
- Ordner
- Templates

## mögliche Kriterien:

- konkret
  - Host/-group
  - Contact/-group
  - Timeperiod
  - [...]
- abstrakt
  - Monitoring-Server
  - Standort
  - Projekt
  - [...]

```

/etc/nagios/
|- nagios.cfg
|- timeperiods.cfg
|- checkcommands.cfg
|- serviceext/
|- templates/
|- [...]
|--- slaveserver1/
|----- gelsenkirchen
|----- herne
|----- holzwickede
|----- hosts_dmz.cfg
|----- services_dmz.cfg
|----- escalations_dmz.cfg
|----- hosts_monitoring.cfg
|----- services_monitoring.cfg
|----- escalations_monitoring.cfg
|----- [...]
|--- slaveserver2/
|----- frankfurt
|----- giessen
|----- [...]

```

# Konfiguration entschlacken: Ballast auslagern

Warum ist eine schlanke Konfiguration sinnvoll?

```
define host {
    host_name          rhs-sles10-nagios01
    host_groups        x-hardware-server, x-hardware-hp, x-os-linux, x-task-nagios
    alias              Nagios-Master-Server
    parents            unDirz-csw001
    address            172.19.40.10
    notifications_enabled 1
    event_handler_enabled 1
    flap_detection_enabled 1
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    first_notification_delay 1
    check_interval     3
    retry_interval     1
    max_check_attempts 8
    contact_groups     ops-sysadmins
    check_command      check-host-alive
    notification_interval 2
    notification_period 24x7
    notification_options d,r,s
    action_url         /nagios/cgi-bin/twiki.cgi?host=$HOSTNAMES
    icon_image         base/sles10.png
    icon_image_alt     SUSE Linux Enterprise Server 10
    vram_image         base/sles10.png
    notes              View infos for this machine in the Wiki.
    statusmap_image    base/20x20sles10.gd2
}
```

```
define host {
    use                server-sles10
    host_name          rhs-sles10-nagios01
    host_groups        x-hardware-server, x-hardware-hp, x-os-linux, x-task-nagios
    alias              Nagios-Master-Server
    parents            unDirz-csw001
    address            172.19.40.10
}
```

## Weniger Text...

- ... erleichtert die Übersicht.
- ... spart Zeit und Arbeit.
- ... ist weniger fehleranfällig.

=> **Templates! :-)**

# Konfiguration entschlacken: Ballast auslagern

Wo und wie kann man Templates einsetzen?

## Basis-Daten

- `nagios.cfg`
- `cgi.cfg`
- `ressource.cfg`
- `Timeperiods`
- `Check Commands`
- `Misc. Commands`
- `Contacts`
- `Contact Groups`

## Erweiterte Daten ("Bewegdaten")

- `Hosts`
- `Host Groups`
- `Host Escalations`
- `Host Dependencies`
- `Services`
- `Service Groups`
- `Service Escalations`
- `Service Dependencies`

## Templates...

- ... werden mit `name` in der Objektdefinition initiiert.
- ... werden mit `use <name>` von anderen Objektdefinitionen genutzt/geerbt.
- ... können mit `register 1` auch weiterhin als eigenes Objekt existieren.

# Konfiguration entschlacken: Ballast auslagern

## Beispiel: Einfaches Host-Template

```
define host {
  host_name      testserver1
  alias          Testserver 1
  address       192.168.1.1
  max_check_attempts 5
  check_interval 5
  retry_interval 2
  check_period  24x7
  contact_groups ops-admins
  notification_interval 5
  notification_period 24x7
}

define host {
  host_name      testserver2
  alias          Testserver 2
  address       192.168.1.2
  max_check_attempts 5
  check_interval 5
  retry_interval 2
  check_period  24x7
  contact_groups ops-admins
  notification_interval 5
  notification_period 24x7
}

define host {
  host_name      testserver3
  alias          Testserver 3
  address       192.168.1.3
  max_check_attempts 5
  check_interval 5
  retry_interval 2
  check_period  24x7
  contact_groups ops-admins
  notification_interval 5
  notification_period 24x7
}
```

```
define host {
  use            basic-host
  host_name      testserver1
  alias          Testserver 1
  address       192.168.1.1
}

define host {
  use            basic-host
  host_name      testserver2
  alias          Testserver 2
  address       192.168.1.2
}

define host {
  use            basic-host
  host_name      testserver3
  alias          Testserver 3
  address       192.168.1.3
}

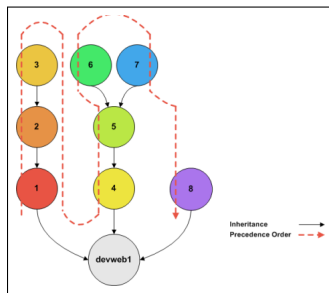
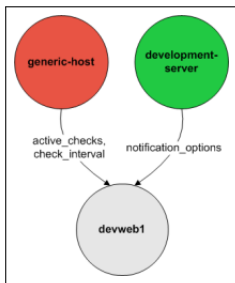
define host {
  name           basic-host
  max_check_attempts 5
  check_interval 5
  retry_interval 2
  check_period  24x7
  contact_groups ops-admins
  notification_interval 5
  notification_period 24x7
  register      0
}
```

- Ersparnis hier:  
7 Zeilen/Host,  
und das bei  
Minimalconfig!
- realistisch: 18  
Zeilen/Host,  
20 Zeilen/Service
- bei ca. 750 Hosts:  
**13.500 Zeilen**
- bei ca. 7000 Services:  
**140.000 Zeilen**

# Konfiguration entschlacken: Ballast auslagern

Worauf muss man achten?

- Standard-Parameter auslagern (z.B. Host: 43 Parameter, nur 8 benötigt)
- Templates für unterschiedliche Konstellationen (Icon Images, Check-Intervalle)
- Nagios 3: mehrere Templates zuweisen, Kettenvererbung



# Ein Schritt weiter...

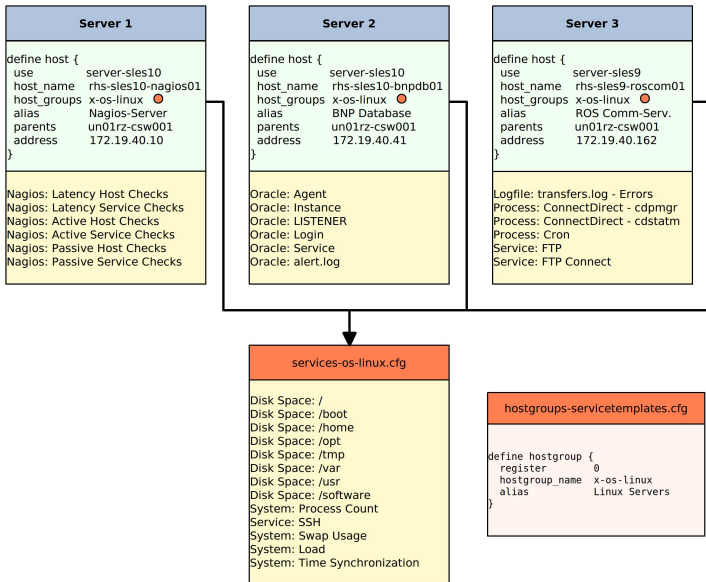
## Vererbung von gruppierten Services

- Nagios kennt eine Beziehung zwischen Service und Host Groups.
- Daraus läßt sich eine Gruppierungsfunktion für Services konstruieren.
- Wie?
  - 1 Erstelle eine neue, leere Hostgruppe.
  - 2 Definiere mehrere Services und weise sie der Hostgruppe zu.
  - 3 Füge der Hostgruppe Hosts hinzu.
  - 4 Alle Hosts der Hostgruppe erben die Services.

# Beispiel: Vererbung von gruppierten Services

Server 1	Server 2	Server 3
<pre>define host { use       server-sles10 host_name rhs-sles10-nagios01 alias     Nagios-Server parents   un01rz-csw001 address   172.19.40.10 }</pre>	<pre>define host { use       server-sles10 host_name rhs-sles10-bnppdb01 alias     BNP Database parents   un01rz-csw001 address   172.19.40.41 }</pre>	<pre>define host { use       server-sles9 host_name rhs-sles9-roscom01 alias     ROS Comm-Serv. parents   un01rz-csw001 address   172.19.40.162 }</pre>
<p>Disk Space: /  Disk Space: /boot  Disk Space: /home  Disk Space: /opt  Disk Space: /tmp  Disk Space: /var  Disk Space: /usr  Disk Space: /software  System: Process Count  Service: SSH  System: Swap Usage  System: Load  System: Time Synchronization  Nagios: Latency Host Checks  Nagios: Latency Service Checks  Nagios: Active Host Checks  Nagios: Active Service Checks  Nagios: Passive Host Checks  Nagios: Passive Service Checks</p>	<p>Disk Space: /  Disk Space: /boot  Disk Space: /home  Disk Space: /opt  Disk Space: /tmp  Disk Space: /var  Disk Space: /usr  Disk Space: /software  System: Process Count  Service: SSH  System: Swap Usage  System: Load  System: Time Synchronization  Oracle: Agent  Oracle: Instance  Oracle: LISTENER  Oracle: LogIn  Oracle: Service  Oracle: alert.log</p>	<p>Disk Space: /  Disk Space: /boot  Disk Space: /home  Disk Space: /opt  Disk Space: /tmp  Disk Space: /var  Disk Space: /usr  Disk Space: /software  System: Process Count  Service: SSH  System: Swap Usage  System: Load  System: Time Synchronization  Logfile: transfers.log - Errors  Process: ConnectDirect - cdpmgr  Process: ConnectDirect - cdstatm  Process: Cron  Service: FTP  Service: FTP Connect</p>

# Beispiel: Vererbung von gruppierten Services



# Ein Schritt weiter...

## Vererbung von gruppierten Services

### Nutzen?

- Nochmal weniger Tipparbeit! :-)
- Anlegen neuer Hosts mit Standard-Diensten nach dem Baukastenprinzip
- Gruppen gleichartiger Maschinen im Frontend (Inventarisierung, Updates etc.)
- Einfaches und schnelles Hinzufügen von Standarddiensten
- Nur noch Ausnahmen in den `services.cfg`-Dateien

# Distributed Monitoring: Konfiguration verteilen mit SVN

## Bedingungen

- SSH-Schlüsselaustausch zwischen den Nagios-Maschinen
- Heartbeat/DRBD-Kombination mit Nagios-Spoolverzeichnis im DRBD

## nagios-update-monitoring-slaves

```
#!/bin/sh
#####
# Script to update nagios configs on monitoring slaves.

SLAVES="172.19.40.186 172.19.40.187"
SLAVE=""

read -p "Enter your Subversion-Username: " SVNUSER
stty -echo
read -p "Enter your Subversion-Password: " SVNPASS; echo
stty echo

if [ ! -e ${SVNUSER} ] && [ ! -e ${SVNPASS} ]; then
for SLAVE in ${SLAVES}; do
ssh ${SLAVE} << EOF
cd /etc/nagios
svn --username=${SVNUSER} --password=${SVNPASS} update && rcnagios check >/dev/null && rcnagios reload
EOF
done
else
echo " * No Username or no Password defined. Bailing out..."
fi
```

- 1 Good to know - Hintergrundwissen
  - Was ist Configuration Management?
    - Offizieller Ansatz
    - "Pragmatischer Ansatz"
  - Was ist eine Versionsverwaltung?
  - Softwarelösungen
  - Subversion - Features und Benefits
  - Wie funktioniert Subversion?
  - SVN und Nagios?
  - Vokabular
- 2 Subversion in der Praxis
  - SVN-Repository anlegen
  - SVN Daily Use
- 3 Gedanken zur Nagios-Konfiguration
  - Konfiguration strukturieren
    - Grundsätzliches
    - Vorüberlegungen
  - Konfiguration entschlacken: Ballast auslagern
  - Best Practices
    - Vererbung von gruppierten Services
    - Distributed Monitoring: Konfiguration verteilen mit SVN
- 4 **Fazit**
  - Zusammenfassung
  - F & A

# Zusammenfassung

Alle Kandidaten nochmal im Schnelldurchlauf...

## Subversion

- Super für Dokumentation, Disaster Recovery und Konfliktbehandlung
- Einfache Anwendung
- Eine Installation, viele Repositories (auch für selbstentwickelte Plugins etc.)
- Konfiguration lokal mit dem Editor des Vertrauens editieren

## Nagios-Konfigurationsstrukturen

- bessere Übersichtlichkeit
- weniger Fehleranfälligkeit
- weniger Tipparbeit
- jede Installation hat andere Prioritäten
- Vorher überlegen, wie es hinterher aussehen soll
- Dateinamen, Ordnerstrukturen und Templates
- für ganz Faule: gruppierte Services

**Questions, anyone?!**

# Das wars!

**Save often, commit early! ;-)  
Danke für euer Interesse!**

#### Quellen:

<http://de.wikipedia.org>  
<http://www.nagios.org>  
<http://www.itil.org/de/>  
<http://www.itil-officialsite.com>

#### Weiterführendes zum Thema:

Subversion-Homepage - <http://subversion.tigris.org>  
Subversion-Handbuch - <http://svnbook.red-bean.com>  
Nagios-Doku - <http://nagios.sourceforge.net/docs/>  
ITIL - <http://www.itlibrary.org/>