Nagios −−− System− und Netzwerk−Überwachung (C) 2008 T.Birnthaler, OSTC GmbH
========== http://www.ostc.de

INHALTSVERZEICHNIS

1) Einführung

* Aufgabe: Überwachen von Netzwerken und ihren Komponenten

* Nagios = Netzwerk + (H)Agios (griechisch: "Heiliger")
  + Acronym: "Nagios Ain't Gonna Insist On Sainthood" ;−)
  + Autor: Ethan Galstad (Amerikaner)
  + 1999 begonnen (als "NetSaint") −> 9 Jahre Entwicklungszeit

* Beobachtet permanent beliebige Bestandteile eines Netzwerks und
  schlägt Alarm, sobald etwas aus dem Ruder läuft

* Stellt Netzwerk−Struktur mit logischen Abhängigkeiten dar
  Stellt auf "Messdaten"−Basis "Zustand" der Netzwerk−Komponenten dar

* STATUSORIENTIERT −> Liefert QUALITATIVE Aussagen gemäß AMPELSYSTEM
  + Schwellwerte für Warning/Critical PRO CHECK flexibel einstellbar

```
check_icmp  ...  −w 200.0,40%  −c 1000.0,80%
check_smtp  ...  −w 5.0        −c 8.0
check_disk  ...  −w 10%        −c 5%
```

Wert	Farbe	Status	Code
0 < ... < Warn−Grenze	Grün	Ok	0
Warn−G <= ... < Critical−G	Gelb	Warning	1
... >= Critical−G	Rot	Critical	2
unbekannt?	Orange	Unknown	3

* Liefert KEINE QUANTITATIVEN Aussagen (zeitlicher Verlauf der Messwerte)
  + Zeichnet allerdings solche Daten auf
  + Auswertung mit Zusatztools ("Addons")

2) Zweck von "Monitoring"

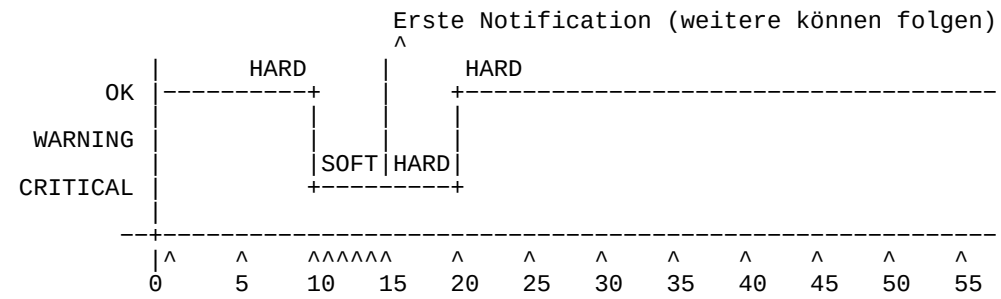
* Fehler erkennen
  + Frühzeitig (bevor vom Benutzer wahrgenommen ;−)
  + Auch wenn man nicht vor dem Rechner sitzt
  + Nicht so offensichtliche
    − Klimaanlage im Rechenzentrum ausgefallen −> 70 Grad!
    − Wassereinbruch
    − Kurzeitiger Stromausfall −> USV angesprungen
    − Diebstahl/Einbruch/Sabotage
    − ...

```
* Gezielte Fehlersuche unterstützen
  (nicht an falscher Stelle suchen!)

* Unnötige Nachrichten vermeiden (99.9% ist OK)
  damit wichtige Ereignisse nicht im Wust von Meldungen untergehen!

* System− und Netzwerkdokumentation!
  (Netzwerkinfrastruktur ändert sich ständig − wächst ;−)

* Dokumentation der Verfügbarkeit gegenüber Kunden
  (Einhaltung von SLA = Service Level Agreements)

* Trends ermitteln −> Kapazitätsplanung

* "Monitoring" (Nagios) ist nicht "Graphing" (MRTG, Cacti)!
  (Visualisierung von Messwerten über längere Zeiträume)


================================================================================
3) Was bietet Nagios?
================================================================================

* Überwachen
  + ÖFFENTLICHE Netzwerkdienste:    SMTP, POP3, IMAP, HTTP, FTP, ...
  + PRIVATE Rechner−Betriebsmittel: CPU, RAM, Disk, User, Prozesse, ...
    −> Agent auf Gerät notwendig:   SSH, NRPE, NSCA, NWSC, SNMP, ...
  + Netzwerkfähige GERÄTE:          Switch, Hub, Router, Drucker, VoIP, ...

* Alle Betriebssystem überwachbar (Linux, UNIX, Windows, MacOS, Host, ...)
  (läuft selbst nur unter Linux/UNIX)

* Modular erweiterbar über PLUGINS

* Berücksichtigt ABHÄNGIGKEITEN (Dependencies) von Netzwerk−Komponenten
  −> Nachrichten−Unterdrückung abhängig vom Zustand anderer Hosts/Services

* Alarmierung (Notification) bei Auftreten/Behebung von Problemen
  −> Email, SMS, Anruf, Chat, ...

* Eskalationsmanagement (falls keine Reaktion erfolgt)
  −> Information anderer/weiterer Kontaktgruppen

* Ereignisbehandlung (Eventhandler) zur automatischen Problemlösung
  (angenehm + gefährlich da Problem evtl. nicht wahrgenommen!)

* Grafische Visualisierung des aktuellen Zustandes

* Aufzeichnung des zeitlichen Verlaufs (Logdateien)

* Open Source (GPLv2)

* Große (deutsche!) Community


================================================================================
4) Anmerkungen zu Nagios
================================================================================

* Nagios ist ein "Framework" und kann erst mal kaum etwas
  Muss zu 100% gesagt bekommen, was es tun soll (nichts automatisch!)
  −> "Nagios ist ein sehr guter Schüler,
      braucht aber auch einen guten Lehrer!"

* Nagios kümmert sich NUR um
  + Plugin−Aufrufe
  + Aufzeichnen + Auswerten der Plugin−Ergebnisse
  + Reaktionen darauf (wiederum über Plugins!)
  −> "Made im Speck"

* Wenige fest eingebaute Funktionen
  "Rest" machen Plugins (+ UNIX−Kommandos!)
  −> Modular + erweiterbar
  −> Skalierung durch Entkopplung der Komponenten

* Aussagen von vergleichbarer Qualität
  + "Nagios kann alles überwachen"
  + "Unser Produkt kann per XML Daten austauschen",
  + "Unser Produkt ist per SNMP auslesbar und steuerbar"


================================================================================
5) Nagios−Struktur
```

```
================================================================================

*                          +---------------+
       +------------------> Web-GUI (CGI) +-------------+
       |                  +-------^-------+             |
       |                          |                     |
       |  +--------+   +-----------v-----------+         |
       |  | AddOns +---> Status- + Log-Dateien |         |
       |  +---^----+   +-----------^-----------+         |
       |      |                   |                     |
    +--+------+-----+      +-------v-------+    +--------v----------+
    | Konfiguration +------> Nagios-Daemon <----+ External Cmd File |
    +---------------+      +-------^-------+    +--------^----------+
                                   |                     |
                             +-----v------+              |
                             | Plugins... +--------------+
                             +--^------+--+
                                |      |
                             +--+      +------+
                             |                |
                        +----+---+   +--------v---------+
                        | Check  |   | Notification     |
                        | (Test) |   | (Benachrichtigung) |
                        +--------+   +------------------+

* Web-GUI:           Übersicht Systemstatus
                     Reportgenerierung
                     Trendanalyse
                     Tests + Benachrichtigungen ein/ausschalten
                     Wartungsfenster ein/ausschalten
                     Rescheduling

* Nagios-Daemon:     Führt periodisch (aktiv) Tests aus
                     Nimmt (passiv) Tests entgegen
                     Wertet Testergebnisse aus
                     Löst Alarme aus

* Status/Log-Datei: Sammeln Plugin-Meldungen
                     Aktueller Nagios-Zustand

* Konfiguration:    Beschreibung zu beobachtender Geräte/Dienste/Ressourcen
                     + ihre Abhängigkeiten

* Check-Plugin:     Durchführung Einzeltest (lokal/remote, aktiv/passiv)
                     Rückmeldung an Nagios-Daemon (Status + Text)

* Notification-Pl:  Versenden von Benachrichtigungen


================================================================================
6) Umfangreiche Web-GUI
================================================================================

* Aufruf: http://localhost/nagios

* In C geschrieben (sic!)

* Darstellung des aktuellen Zustandes
  (auf Basis permanenter Logdatei-Analyse)

* Nagios-Daemon steuern
  + Checks/Benachrichtigungen ein/ausschalten
  + Störungsmeldungen quittieren
  + Kommentare hinterlegen ("in Bearbeitung")
  + SCHEDULING der Anfragen beeinflussen

* Zunächst NUR HTTP von "localhost" erlaubt
  + Erweiterbar auf beliebige Rechner
  + Verschlüsselter Zugriff per HTTPS möglich

* Anmeldung konfigurieren
  + Zunächst nur "Basic Authentication"
  + Beeinflusst Sichtbarkeit von Hosts/Services
  + Beeinflusst erlaubte Tätigkeiten


================================================================================
7) Was ist notwendig um Nagios laufen zu lassen?
================================================================================

* Mindestens
  + Linux/UNIX-Rechner
```

```
    + Quellcode von Nagios−Daemon, Nagios−Plugins und Nagios−GUI (C)
    + C−Compiler
    + Netzwerkkarte + konfiguriertes TCP/IP−Netzwerk

  * Für GUI
    + Apache2 Webserver
    + gd−library

  * Für Plugin−Ausführung
    + Perl
    + Shell
    + ...

  * Für Benachrichtigung
    + Mailserver Postfix/Exim/Sendmail
    + SMS−Gateway
    + Echte "Ampel" am Serverschrank ;−)

  * Für einige Addons
    + Quellcode Addons
    + Datenbank MySQL (PostgreSQL)

  * Lizenz ;−)
    + GNU General Public Licence V2 (GPLv2, analog Linux)
    + Open Source (sehr wichtig!)

  * Geld ;−(
    + Einarbeitung
    + Dienstleistung
    + Hardware
    + Konfiguration
    + Problem−Bearbeitung


  ================================================================================
  8) Wie als Anfänger beginnen?
  ================================================================================

  * "Geduld ist eine Tugend"
    + Nagios ist mächtig und flexibel
    + Konfiguration wie gewünscht dauert wegen
      − Lernen von Begriffe + Verhalten + Optionen
      − Eigenes Netzwerk verstehen
      − Konfiguration erstellen
      − Plugins finden/entwickeln/installieren/testen

  * Quickstart lesen
    + Installation aus Quellen ist sinnvoll (immer gleich)
    + Zunächst nur Nagios−Maschine lokal monitoren
    + Netzwerk−Dokumentation schrittweise erarbeiten

  * Plugin IMMER zuerst auf Kommandozeile austesten!

  * Überwachung schrittweise erweitern

  * Benachrichtigungen auf WIRKLICH relevante "eindampfen"


  ================================================================================
  9) Zum Verständnis von Nagios
  ================================================================================

  * Check−Unterscheidung
    + Active:  Von Nagios−Daemon ausgelöst
               In regelmässigen Abständen SYNCHRON ausgeführt (einstellbar)
               Plugin wird aufgerufen und Rückgabewert ausgewertet (Timeout)
    + Passive: Durch externes Programm selbständig durchgeführt
               Liefert ASYNCHRON Status an Nagios−Daemon (z.B. SNMPTrap)
    + Host:    Ping               −> Host antwortet
    + Service: Verbindungsversuch −> Service antwortet
    + Local:   Auf Nagios−Server
    + Remote:  Auf überwachter Maschine oder vermittelnder Maschine

  * Alarmstatus−Typen (falsche Alarme vermeiden)
    + SOFT = Bemerkt aber noch nicht sicher (1x fehlgeschlagen)
      − Logging
      − Eventhandler ausführen
      − KEINE Benachrichtigung, aber Anzeige in Weboberfläche
    + HARD = Sicher genug erkannt (Nx fehlgeschlagen)
      − Logging
      − Eventhandler ausführen
      − Benachrichtigung und Anzeige in Weboberfläche
```

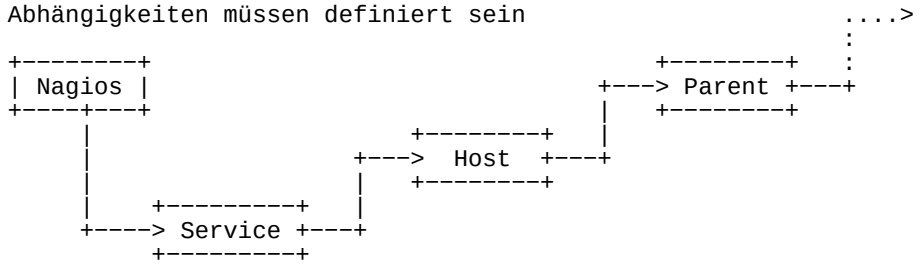
− Max. Zeit bis Benachrichtigung = (5 Min + 4 x 1 Min) = 9 Min



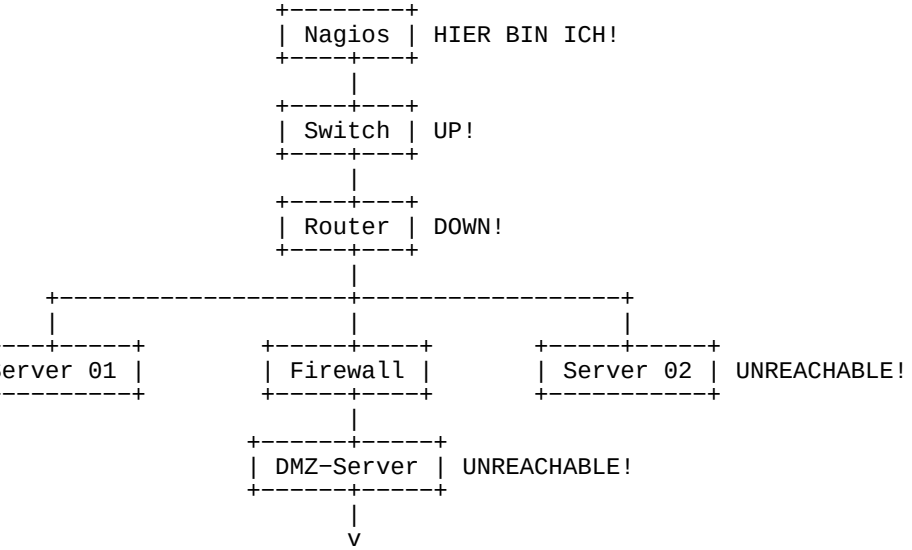
* Zustand eines Checks nach Statuswechsel −> Typ "SOFT"
  Nach definierbarer Anzahl schnellerer Wiederholungen −> Typ "HARD"

* Statuswechsel "HARD" −> "SOFT" gibt es nicht
  Wenn Host/Gerät/Dienst wieder funktioniert, dann funktioniert es/er wieder

* Unterscheidung
  + Service−Checks regelmäßig ausgeführt
  + Host−Checks nur bei Bedarf (wenn Service−Check fehlschlägt)
  + Host nicht mehr erreichbar −> Parent versucht zu erreichen
    −> Abhängigkeiten müssen definiert sein



* Schlussfolgerungen aus Netzwerk−Topologie
  (definiert durch child−parent−Abhängigkeiten)



10) Konfiguration von Nagios

* Per EDITOR in Form von TEXTDATEIEN konfiguriert (keine GUI)
  (aus bereits vorhandenen Inventarlisten per Skript generieren ;−)

* Abstrakte OBJEKTE bilden Realität ab, jedes bekommt einen INTERNEN NAMEN,
  (muss nicht mit EXTERNEM Namen, z.B. Benutzer, Gruppe, Host übereinstimmen)
  −> BAUKASTENSYSTEM basierend auf NAMEN

* Konfigurationsdateien ob ihrer Unzahl Einstellungsmögl. oft kritisiert
  −> Analoges Problem bei Samba mit "smb.conf"

* GRÜNDLICHE PLANUNG der Konfiguration passend zur eigenen Struktur wichtig
  + TEMPLATES für gleichartige Geräte/Dienste einsetzen + vererben
  + Passende Namen für alles vergeben

```
  + "Mal schnell loslegen" führt ins Chaos!

* Gleichzeitig System− und Netzwerkdokumentation!
  −> Rechenzentrumsumzug!

* Konfiguration von Nagios hört NIE auf
  −> Man kann gar nicht alles von vorne herein voraussehen
  −> Genug Zeit/Geld zum Planen und Sammeln der Daten nehmen!


================================================================================
11) Warum keine Konfiguration per GUI?
================================================================================

* Zuviele Einstellungsmöglichkeiten in Konfigurationsdateien machen
  Realisierung vollständiger GUI schwer (mit jeder Version mehr)

* Es gibt "professionelle" GUI−Ansätze (gut und schlecht gleichermassen)
  + Aber warum "weitere Abstraktionsschicht/Metaebene" einziehen?
  + Verstehen muss man die Struktur + Arbeitsweise des Systems sowieso!
  + Viele Geräte/Dienste in GUI erfassen unangenehm + fehleranfällig


================================================================================
12) Kommunikation Nagios−Server <−> Plugins
================================================================================

* NAGIOS−SERVER
  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  X                        +----------------+                                X
  X      +-----------------+---> Nagios−Daemon <----+---------+---------+      X
  X      |                 |   +---------^------+   |         |         |      X
  X      |                 |             |          | +-------+-------+ |      X
  X      |                 |             |          | | Externe Kmdo− | |      X
  X      |                 |             |          | | Schnittstelle | |      X
  X      |                 |             |          | +-------^-------+ |      X
  X      |                 |             |          |         |         |      X
  X +----v----+ +------v-----+ +-----v----+ +---v---+ +---+---+ +----v----+ X
  X |check_***| |check_by_ssh| |check_nrpe| |  SNMP | | NSCA | | check_nt| X
  X | PLUGIN  | |   PLUGIN   | |  PLUGIN  | | PLUGIN| |DAEMON| | PLUGIN  | X
  X +----^----+ +------^-----+ +-----^----+ +---^---+ +--^---+ +----^----+ X
  XXXXXXX|XXXXXXXXXXXXX|XXXXXXXXXXXXX|XXXXXXXXXX|XXXXXXXX|XXXXXXXXXX|XXXXXXX
         |              |             |          |        |          |
  XXXXXXX|XXXXXXXXXXXXX|XXXXXXXXXXXXX|XXXXXXXXXX|XXXXXXXX|XXXXXXXXXX|XXXXXXX
  X +----v----+ +------v-----+ +-----v----+ +---v---+ +--+---+ +----v---+  X
  X | Service | |    sshd    | |   nrpe   | | snmpd | |send_ | | NC_Net |  X
  X |(öffentl)| |   DAEMON   | |via xinetd| | DAEMON| | ncsa | | DIENST |  X
  X +---------+ +------^-----+ +---^−^−^−−+ +---^---+ |CLIENT| +--------+  X
  X                    |           | | |        :     +----^−+             X
  X                    |           | | |        :          |                X
  X             +------v-----+ +---v−v−v−−+ +---v------+ +−+------------+  X
  X             |  check_*** | |check_*** | |check_*** | | Ergebnis des |  X
  X             |   PLUGIN   | |  PLUGIN  | |  PLUGIN  | |Service−Checks|  X
  X             +------------+ +----------+ +----------+ +--------------+  X
  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      CLIENT1      CLIENT2        CLIENT3     CLIENT4  CLIENT5  CLIENT6

* Auch remote von Nagios angestossene Checks heißen "Plugin"!

* SSH
  + Meist automatisch installiert
  + Durch Schlüsselaustausch Anmeldung ohne Passwort (und Passphrase!)
  + Sehr aufwendiger Verbindungsaufbau
    −> "Da kommt die Brüh' teurer als das Fleisch"
  + Missbrauchbar zum Ausführen von Kommandos
  + Extrem sicheres "Rohr" auf Basis extrem unsicherer Konfiguration aufbauen
    −> Durch das "Rohr" ist erst mal alles erlaubt

* SNMP (Simple Network Management Protocol)
  + Meist automatisch installiert
  + Meist auf netzwerkfähigen Geräten vorhanden
  + Lesen und/oder Schreiben von Information
  + Existenz einer gewissen Anz. von Prozessen überwachen ist eingebaut
  + Missbrauchbar zum Ausführen von Kommandos

* NRPE (Nagios Remote Plugin Executor)
  + Verfügbar für Linux, UNIX, BSD, MacOS, Windows
  + Von Nagios angestossen
  + Führt beliebig viele Tests aus ("indirekte Checks")
  + Reicht Ergebnis an Nagios zurück
  + Client "check_nrpe" + Daemon "nrpe" + Konfiguration "nrpe.cfg"
```

```
* NSCA (Nagios Service Check Acceptor)
  + Wird von alleine tätig
  + Liefert Ergebnisse an Nagios zurück
  + Erlaubt verteilte und mehrstufige Nagios−Setups

* NWSC (Nagios Windows Service Checker)
  + Auf EINEM Windows Rechner zu installieren (Perl + Module)
  + Muss Domänenenadministrator−Rechte haben
  + WMI−Abrufe auf andere Windows−Rechner (Windows Management Instrumentation)
  + Aufträge per HTTP−Anfrage
  + Rückgabe der WMI−Daten per HTTP−Antwort


================================================================================
13) Was bei Plugins zu beachten?
================================================================================

* Nagios schreibt Kommunikationsform mit Plugins genau vor
  + Kommando mit bestimmten Optionen (teilweise vorgeschrieben)
  + Exit/Returncode 0/1/2/3 je nach Status
  + Ein/mehrzeiliger Text bestimmter Struktur
  + Pluginausgabe darf nach "|" Performance Daten (PerfData) enthalten
    -> Langfristige Trends erkennen

* Sauber definierte und einfache "Kommando−Schnittstelle" (API)
  -> Programmiersprache beliebig
  -> Plugins bleiben kompatibel
  -> Plugins leicht weiterzuentwicklen, ergänzen, korrigieren
  -> Plugin−Skripte müssen sich aber EXAKT daran halten!

* Exitcode des Plugins (= Status)

        +------+----------+-------------------------+
        | Code | Service  | Host                    |
        +------+----------+-------------------------+
        | 0    | OK       | UP                      |
        | 1    | WARNING  | UP or DOWN/UNREACHABLE  |
        | 2    | CRITICAL | DOWN/UNREACHABLE        |
        | 3    | UNKNOWN  | DOWN/UNREACHABLE        |
        +------+----------+-------------------------+

* Optionen mit reservierter Bedeutung für Plugins
  (können aber noch beliebige spezifische Optionen haben)

        +----+------------+--------------------+
        | −c | −−critical | Kritische Grenze   |
        | −w | −−warning  | Warnungsgrenze     |
        | −h | −−help     | Hilfe (kurz/lang)  |
        | −H | −−hostname | Rechner            |
        | −t | −−timeout  | Abbruch (Sek)      |
        | −v | −−verbose  | Mehr Ausgaben      |
        | −V | −−version  | Versionsnummer     |
        | −4 | −−use−ipv4 | IPv4 benutzen      |
        | −6 | −−use−ipv6 | IPv6 benutzen      |
        +----+------------+--------------------+

* Beispiel Root−Partition "/" testen
    check_disk −w 10% −c 5% /
    -> DISK OK − free space: / 8627 MB (64% inode=73%);| /=4740MB;12674;13378;0;14083

    check_disk −w 60% −c 50% /
    -> DISK WARNING − free space: / 8627 MB (64% inode=73%);| /=4740MB;12674;13378;0;14083

    check_disk −w 90% −c 80% /
    -> DISK CRITICAL − free space: / 8627 MB (64% inode=73%);| /=4740MB;12674;13378;0;14083

    df −h
    -> Filesystem Size  Used Avail Use% Mounted on
       /dev/sda7   14G  4.7G  8.5G  36% /

* Programmierung (Minibeispiele)
  + Shell:  #!/bin/bash
            echo "DISK OK − free space: ..."
            exit 0
  + Perl:   #!/usr/bin/perl −w
            use stric;
            printf "DISK WARNING − free space: ...\n";
            exit 1;

* Plugins IMMER auf Kommandozeile vollständig austestbar
  -> Riesiger Vorteil!
```

```
================================================================================
14) Auswahl von Plugins
================================================================================

* Plugins muss man immer passend zum Einsatzzweck aussuchen
  (es gibt ein paar generische, z.B. "check_udp", "check_tcp")

* Suche nach passendem Plugin aufwendig (und oft frustrierend ;−()
  Viele superspeziell auf EIN persönliches Problem hin geschrieben
  (z.B. Domänenname und Anmeldedaten fest einprogrammiert)
  −> Will man sie verwenden, muss man sie eigentlich neu schreiben

* Teilweise grottenschlechter Qualität, funktionieren nicht vernünftig
  −> Lange suchen oder lieber gleich selber schreiben?
  −> "Da kennt jemand einen, der einen kleinen Bruder hat,
      der auch schon mal die Shell benutzt hat;
      und der schreibt dann Plugins für Nagios,
      welche die Qualität von Nagios insgesamt runterziehen."

* Man MUSS programmieren können (Perl, Shell, ...)
  Kombination Shell + UNIX−Kommandozeilentools einfach + leistungsfähig

* SNMP−fähige Geräte oft schlecht dokumentiert
  −> Ausprobieren nötig

* Datenausgabe von Geräten oft unstrukturiert und zu grossx
  Relevante Daten im "Sumpf" zu suchen aufwendig
  −> Problem der Geräte, nicht von Nagios.
     "Nagios kann nur soviel, wie die zugrundeliegende Soft/Hardware"

* Man muss einen VERLÄSSLICHEN Mechanismus haben, um mit Geräten
  (z.B. Switch, Thermofühler) zu reden. Hat man den − völlig
  unabhängig von Nagios − kann man es auch in Nagios überwachen

* Sind auf der Kommandozeile keine Daten von einem Gerät zu erhalten,
  dann ist es mit Nagios nicht überwachbar
  −> Geräte mit wunderschöner Java/Flash−Weboberfläche (aber "geheimer" API)
     evtl. mit Nagios nicht überwachbar (IBM Raid −> IBM Director notwendig)

* Wenn möglich NICHTS auf Endgeräten installieren (Aufwand vermeiden)
  −> Oft "snmpd" unter Linux vergewaltigt und zum Mittler gemacht,
     da er Prozesse überwachen (vorhanden, Anzahl),
     beliebige Skripte aufrufen und
     Ergebnis via SNMP zurückgeben kann (z.B. "raidcheck"−Skript)


================================================================================
15) Nagios Objekte und ihre Beziehungen
================================================================================

* Host (genau eine IP)
  + HW−Gerät im Netz
  + Host−Check mit "ping" (ICMP)

* Hostgroup
  + Zusammenfassung von Hosts
  + Zur Darstellung, Auswertung, Reaktion

* Service (mehrere pro Host)
  + Port + Protokoll
  + Freier Festplattenplatz
  + Neue Updates für Distribution
  + Wann läuft SSL−Zertifikat ab?
  + ...

* Servicegroup
  + Zusammenfassung von Services
  + Zur Darstellung, Auswertung, Reaktion

* Dependency (Logische Abhängigkeit)
  + Eltern−Kind−Beziehungen
  + Gesamte Netzwerk−"Topologie" daraus abgeleitet

* Command (Plugin−Aufruf)
  + Service−Check mit Parametern
  + Benachrichtigung (Notification)

* Contact (Kontaktperson, Verantwortlicher)
  + Benachrichtigungen (Notification)
  + Sichtbarkeit in GUI
  + Nachrichten−Zeitraum
```

```
    + Mail, SMS, ...

  * Contactgroup (Gruppe von Verantwortlichen)
    + Benachrichtigungen
    + Sichtbarkeit in GUI
    + Nachrichten−Zeitraum
    + Mail, SMS, ...

  * Timeperiod (Zeitraum/Zeitfenster)
    + Überwachungszeitraum
      − Wartungszeitraum
      − Problembehandlung
    + Benachrichtigungszeitraum (Notification Intervall)

  * Eskalationskette

  * Template (Vorlage)
    + Unvollständig definiertes Objekt
    + Ableiten von Objekten einheitlicher Struktur
    + Mehrfache + hierarchische Vererbung


================================================================================
16) Konfigurationsdateien − Struktur
================================================================================

  * Verzeichnisstruktur unter "/usr/local/nagios"
      bin          Nagios−Daemon
      etc          Konfigurations−Dateien
      etc/objects  Weitere Konfigurations−Dateien (pro Objekt/Objektklasse)
      libexec      Plugins
      sbin         CGI−Skripte der Web−GUI
      share        Dokumentation + statische Teile der Web−GUI
      var          Status−Daten + Log−Dateien

  * Zusammenhang der Konfigurations−Dateien

                 +---------------+     +---------------+
                 | Nagios−Daemon |     | Web−GUI (CGI) |
                 +-------+-------+     +-------+-------+
                         |                     |
                         v                     v
                   +-----+------+         +---------+
                   | nagios.cfg |<--------+ cgi.cfg |
                   +--+------+--+         +---------+
                      |      |
                 +----+      +------+
                 |                  |
                 v                  v
        +----------------+   +--------------+
        |    Objekte     |   | resource.cfg |
        | commands.cfg   |   +--------------+
        | contacts.cfg   |
        | timperiods.cfg |
        | templates.cfg  |
        | ...            |
        +----------------+

  * Aufteilung der Konfigurationsdaten auf Dateien (Name + Verz.) frei wählbar
    + "nagios.cfg" ist Startpunkt

  * Alle Konfigurationsdateien nach JEDER Änderung prüfen (−v=verify)
    −> /usr/local/nagios/bin/nagios −v /usr/local/nagios/etc/nagios.cfg

  * Format der Konfigurationsdateien ändert sich kaum
    −> Versionswechsel leicht möglich (z.B. von 2.0 auf 3.0)

  * Von TEMPLATE (Vorlagen) Objekt−Eigenschaften "erben"
    + Template = UNVOLLSTÄNDIG definiertes Objekt
      − Direktive "register 0"
      − Parameter hinzufügen
      − Parameter überschreiben
    + Vererben bzw. ableiten
      − Direktive "use TEMPLATE"
    + Vordefinierte Templates (z.B. "generic−host")
    + Mehrfachvererbung + Vererbungshierarchie erlaubt

  * Mitgelieferte Beispielkonfiguration
    + Startet zunächst nicht
    + Mind. 1 Host + 1 Service + 1 Contact definieren
    + 1. Test lokal durchführen
      − "lokalhost"            erreichbar
```

```
    - "lokale NW-Verbindung" erreichbar

* Mögliche Dateien unter "/usr/local/nagios/etc/config"

    Datei              | Bedeutung                                          |
    -------------------+----------------------------------------------------+
    nagios.cfg         | Hauptkonfiguration (Server-Optionen + Includes)    |
    resource.cfg       | Benutzerdef. Macros (z.B. Pfade, Passworte)        |
    cgi.cfg            | Web-GUI CGI-Konfiguration                          |
    objects/...        | Weitere Objekte (Kommandos, Server, ...)           |
    templates.cfg      | Templates (Vorlagen) für Hosts + Services          |
    -------------------+----------------------------------------------------+
    hosts.cfg          | Überwachte Systeme (IP!)                           |
    hostgroups.cfg     | Zsfg. von Systemen zu Gruppen                      |
    services.cfg       | Überwachte Dienste und Betriebsparameter           |
    servicegroups.cfg  | Zsfg. von Services zu Gruppen                      |
    dependencies.cfg   | Abhängigkeiten von Hosts/Services untereinander    |
    -------------------+----------------------------------------------------+
    commands.cfg       | Kommandos für Tests + Benachrichtigungen           |
    checkcommands.cfg  | Checks mit Aufrufparametern der Plugins            |
    misccommands.cfg   | Kommandos (z.B. für Benachrichtigungen)            |
    timeperiods.cfg    | Definition von Zeitfenstern (z.B. "workhours")     |
    -------------------+----------------------------------------------------+
    contacts.cfg       | Kontaktpersonen                                    |
    contactgroups.cfg  | Zsfg. von Kontaktpersonen zu Gruppen               |
    escalations.cfg    | Eskalationen für anhaltende Fehler                 |
    -------------------+----------------------------------------------------+
    localhost.cfg      | Checks für Localhost                               |
    printer.cfg        | Checks für Drucker                                 |
    switch.cfg         | Checks für Switches/Hubs/Router                    |
    windows.cfg        | Checks für Hosts+Services von Windows-Rechnern     |
    -------------------+----------------------------------------------------+

* Format der Konfigurationsdateien
  + nagios.cfg
      VAR = WERT
  + Objekte
      define TYP{
        VAR WERT
        ...
      }
  + ";" leitet Kommentar ein (bis Zeilenende, nicht "#"!)
  + Listen OHNE Leerzeichen durch "," trennen (Host, Service, Contact, ...)
  + "*" steht für ALLE (Hosts, Services, ...)
  + "!" trennt Argumente beim Aufruf eines Plugins
  + "$...$" kennzeichnet Macros


================================================================================
17) Konfigurationsdateien - Inhalt
================================================================================

* "nagios.cfg" - Hauptkonfigurationsdatei
  + Wird als einzige direkt eingelesen
  + Darin Einzeldateien oder Verzeichnisse (rekursiv) einbinden
  + NUR Dateien mit Endung ".cfg" relevant!
      cfg_file = /usr/local/nagios/etc/objects/commands.cfg
      cfg_dir  = /usr/local/nagios/etc/objects
      ...

* "resource.cfg" - Pfadangaben und Passworte (max. 32 Variablen)
      $USER1$ = /usr/local/nagios/libexec
      $USER2$ = /usr/local/nagios/libexec/eventhandlers
      $USER3$ = user
      $USER4$ = geheim
      ...

* Rechner-Objekt (vollständige Definition, *=MUSS)
  + Kommando-Angabe ohne Parameter
    define host{
       *host_name             charlton
        hostgroups            linux
        alias                 Sony Notebook
       *address               192.168.0.1
        check_command         check_host        ;Kommando-Aufruf
       *max_check_attempts    5
       *check_period          24x7
       *contact_groups        admins
       *notification_interval 240
       *notification_period   24x7
       *notification_options  d,r
        parents               switch
```

```
    }

* Rechner−Template (partielle Definition, *=MUSS)
  + Kommando−Angabe ohne Parameter
    define host{
       *name                  template−host
       *register              0

        check_command         check_host        ;Kommando−Aufruf
        max_check_attempts    5
        check_period          24x7
        contact_groups        admins
        notification_interval 240
        notification_period   24x7
        notification_options  d,r
    }

* Rechner−Objekt (Template + Ergänzungen = vollständige Definition, *=MUSS)
    define host{
        use                   template−host

       *host_name             charlton
        hostgroups            linux
        alias                 Sony Notebook
       *address               192.168.0.1
        parents               switch
    }

* Hostgroup−Objekt (*=MUSS)
    define hostgroup{
       *hostgroup_name        linux
       *alias                 Linux−Rechner
        members               charlton
    }
    define hostgroup{
       *hostgroup_name        routers
       *alias                 Router/Switches
    }

* Service−Objekt (vollständige Definition, *=MUSS)
  + Kommando−Angabe mit Parametern
    define service{
       *host_name             charlton
       *service_description   PING
       *check_command         check_icmp!100.0,20%!500.0,60%     ;Plugin−Aufruf
       *max_check_attempts    3
       *normal_check_interval 5
       *retry_check_interval  1
       *check_period          24x7
       *notification_interval 240
       *notification_period   24x7
       *notification_options  c,r
       *contact_groups        admins
    }

* Service−Template (partielle Definition, *=MUSS)
    define service{
       *name                  template−service
       *register              0

        max_check_attempts    3
        normal_check_interval 5
        retry_check_interval  1
        check_period          24x7
        notification_interval 240
        notification_period   24x7
        notification_options  c,r
        contact_groups        admins
    }

* Service−Objekt (Template + Ergänzungen = vollständige Definition, *=MUSS)
  + Kommandoangabe mit Parametern
    define service{
        use                   template−service

       *host_name             charlton
       *service_description   PING
       *check_command         check_icmp!100.0,20%!500.0,60%     ;Plugin−Aufruf
    }

* Weitere Service−Objekte (*=MUSS)
    define service{
```

```
        *host_name              charlton,switch
        *service_description    PING
        ...
    }
    define service{
        *hostgroup_name         charlton,switch
        *service_description    PING
        ...
    }
    define service{
        *host_name              *
        *service_description    PING
        ...
    }

* Timeperiod−Objekt (*=MUSS)
    define timeperiod{
        *timeperiod_name        27x7
         alias                  7x24h
         sunday                 00:00−24:00
         monday                 00:00−24:00
         tuesday                00:00−24:00
         wednesday              00:00−24:00
         thursday               00:00−24:00
         friday                 00:00−24:00
         saturday               00:00−24:00
    }

* Command−Objekt (für Check, *=MUSS)
  + Plugin−Aufruf mit Parametern
    define command{
        *command_name           check_host
        *command_line           $USER1$check_host −H $HOSTADDRESS$     ;Plugin−Aufruf
    }
    define command{
        *command_name           check_icmp
        *command_line           $USER1$check_icmp −H $HOSTADDRESS$\    ;Plugin−Aufruf
                                  −w $ARG1$ −c $ARG2$
    }

* Command−Objekt (für Benachrichtigung, *MUSS)
  + Plugin−Aufruf mit Parametern
    define command{
        *command_name           notify−by−email
        *command_line           printf "%b" "MELDUNG..." |\            ;Plugin−Aufruf
                                  mail −s "SUBJECT..." ADDRESS
    }
    define command{
        *command_name           notify−by−sms
        *command_line           /usr/bin/printf "%b" "MELDUNG..." |\   ;Plugin−Aufruf
                                  smsclient NUMBER
    }

* Contact−Objekt (*=MUSS)
  + Kommandoangabe mit Parametern
    define contact{
        *contact_name                   nagadmin
         contactgroups                  admins
        *alias                          Nagios Admin
        *host_notification_period       24x7
        *service_notification_period    24x7
        *host_notification_options      d,r
        *service_notification_options   w,c,r
         host_notification_commands     host−notify−...        ;Kommando−Aufruf
         service_notification_commands  notify−by−email        ;Kommando−Aufruf
         email                          nagadmin@localhost
    }

* Contactgroup−Objekt
    define contactgroup{
        *contactgroup_name      nagadmin
        *alias                  Administratoren
    }


================================================================================
18) Benachrichtigungen
================================================================================

* Durch externe Programme durchgeführt (Plugins)

* Einzustellen ist (vielfache Einstellungsmöglichkeiten)
```

```
    + WANN generieren?
    + WANN zustellen?
    + WEM zustellen?
    + WIE zustellen?

  * Erst nach Statuswechsel in den Typ "HARD" wird per FILTERKETTE bestimmt,
    ob eine Benachrichtigung verschickt werden soll
                                                                  BEZOGEN AUF
      +-------------------------------------------------------------+
      | HARD-Statuswechsel eines Host/Service?                      |  Host/Service
      |           | JA                                              |
      | Benachrichtigungen Nagios-weit aktiv?                       |  Host/Service
      |           | JA                                              |
      | Host/Service in geplanter Downtime?                         |  Host/Service
      |      NEIN |                                                 |
      | Status in "notification_options"?                           |  Host/Service
      |           | JA                                              |
      | Aktuelle Zeit in "notification_period"?                     |  Host/Service
      |           | JA                                              |
      | Schon Nachricht über diesen Statuswechsel verschickt?       |  Host/Service
      |           |                 | JA                            |
      |      NEIN |   Seitdem "notification_interval" abgelaufen?   |  Host/Service
      |           |                 | JA                            |
      | Zuständige Kontakte ermitteln!                              |  ------------
      |           |                                                 |
      | Status in "notification_options"?                           |  Contact
      |           | JA                                              |
      | Aktuelle Zeit in "notification_period"?                     |  Contact
      |           | JA                                              |
      | Benachrichtigung auslösen!                                  |  Contact
      +-------------------------------------------------------------+

  * Einstellungen der Benachrichtigungen
    + Host/Service: notifications_enabled        = 1
                    notification_options         = c,w,u,r,f
                    notification_period          = 24x7
                    notification_interval        = 120
    + Contact:      host_notification_options    = ...
                    service_notification_options = ...
                    host_notification_options    = ...
                    service_notification_options = ...

  * Bei welcher Service-Statusänderung Benachrichtigung verschicken?
    -> service_notification_options

          +---+------------+
          | n | NONE (nie) |
          | w | WARNING    |
          | c | CRITICAL   |
          | u | UNKNOWN    |
          | r | RECOVERING |
          | f | FLAPPING   |
          +---+------------+

  * Bei welcher Host-Statusänderung Benachrichtigung verschicken?
    -> host_notification_options

          +---+------------+
          | n | NONE (nie) |
          | d | DOWN       |
          | u | UNREACHABLE|
          | r | RECOVERING |
          | f | FLAPPING   |
          +---+------------+


  ============================================================================
  19) Erweiterungen Nagios 3.0
  ============================================================================

  * Performance-Verbesserung in großen Umgebungen
    + Deutlich verbesserte Hostcheck-Logik (Hostcheck und Servicecheck parallel)
      (bisher wurden alle SC ausgesetzt, bis HC durchgeführt)
    + Zwischenspeicherung von Check-Ergebnissen (Cache)

  * Plugin-Ausgabe mehrzeilig und max. 4 KByte
    (bisher einzeilig und max. 350 Byte)

  * Timeperiods flexibilisiert (bisher nur Wochentag + Zeit möglich)

  * Benachrichtigung bei Beginn UND Ende von Downtime-Ereignissen
```

```
* Verzögerung von Benachrichtigung einstellbar

* Vererbung
  + Services erben Benachrichtigungseinstellungen vom Host (falls fehlend)
  + Mehrfachvererbung von Templates
    + "+"...    fügt hinzu
    + "null"... macht leer

* Erweiterte (hierarchische) Gruppierung
  + Host    − Hostgroup    − Hostgroup    − ...
    Service − Servicegroup − Servicegroup − ...
    Contact − Contactgroup − Contactgroup − ...
  + Ausnahmen wegnehmen mit "!..."

* Vollständig überarbeitete und neu gestaltete Dokumentation

* Readonly−Zugriff in GUI (Kontakte dürfen keine Kommandos absetzen)
  (sowie Kontakte mit GUI−Zugang ohne Benachrichtigungen)

* Embedded Perl besser konfigurierbar
  + Ein/Ausschalten pro Plugin

* Zusätzliche Optimierungsparameter

* Viele kleine, unscheinbare Dinge −> leichter, charmanter
  + Neue Macros $...$
  + Benutzerdefinierte Variablen "_XXX" (Custom Commands)

* Neue Plugins
  + check_logfile
  + check_multi
  + ...


================================================================================
20) Erweiterungen Nagios 4.0 (geplant)
================================================================================

* Neue GUI auf PHP−Basis (statt C)

* Themen für GUI

* Also: Reine Optik!


================================================================================
21) Links
================================================================================

* Deutsch
  http://www.nagios−portal.de/                      Nagios Community
  http://www.nagios−wiki.de/                        Wiki der Nagios Community
  http://listi.jpberlin.de/mailman/listinfo/nagios  Mailingliste
  http://www.swobspace.net/                         Nagios−Buch (deu)
  http://www.netways.de/de/produkte/nagios/         Netways Nagios (Firma)

* English
  http://www.nagios.com/                            Nagios Homepage (kommerziell)
  http://www.nagios.org/                            Nagios Homepage (frei)
  http://www.nagiosplugins.org/                     Nagios−Plugins Homepage
  http://www.nagiosexchange.org/                    Nagios Plugin Austauschplattform
  http://www.nagioscommunity.org/                   Nagios Community
  http://www.nagioswiki.org/                        Nagios Wiki
  http://www.nagiosforge.org/                       Nagios Forge
  http://www.groundworkopensource.com/              Groundwork (Firma)
  http://www.nagios.org/support/mailinglists.php    Nagios Mailingliste

* Addons
  http://www.nagiosql.org/                          NagiosQL (Webinterface)
  http://www.nagvis.org/                            NagVis
  http://www.gnokii.org/                            Gnokii
  http://www.sta.to/ftp/yaps/                       Yaps
  http://www.miwi−dv.com/nrpent/                    Nrpe_nt
  http://www.hendrik−sattler.de/scmxx               SCMxx (Siemens S35i)
  http://www.ederdrom.de/pnp/de/start               PNP (PNP is not PerfParse)
  http://www.pnp4nagios.org/pnp/start               PNP4Nagios
  http://perfparse.sf.net/                          PerfParse
  http://cacti.net/                                 Cacti
  http://www.nedi.ch/doku.php                       NeDi
  http://oss.oetiker.ch/rrdtool/                    RRDtool
  http://ganglia.info/                              Ganglia
  http://www.nagiosadmin.de/                        NagiosAdministrator
```

```
    http://sourceforge.net/project/showfiles.php?group_id=130574  Monarch


================================================================================
22) Vim−Makros zum Blättern im Vortrag
================================================================================
map + H/^=\+\n.*\n^=\+<CR>z<CR>:set nohls<CR><C−e>/^[*1][ )]<CR>/<BS>
map − Hk?^=\+\n.*\n^=\+<CR>z<CR>:set nohls<CR><C−e>/^[*1][ )]<CR>/<BS>
map # /^INHALTSVERZEICHNIS<CR>z<CR>/^[*1][ )]<CR>/<BS>
```