



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Jan Lepel

**Automatisierte Erstellung und Provisionierung von ad hoc
Linuxumgebungen -
Prototyp einer Weboberfläche zur vereinfachten
Inbetriebnahme individuell erstellter
Entwicklungsumgebungen**

Jan Lepel

**Automatisierte Erstellung und Provisionierung von ad hoc
Linuxumgebungen -
Prototyp einer Weboberfläche zur vereinfachten
Inbetriebnahme individuell erstellter
Entwicklungsumgebungen**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: MSc Informatik Oliver Neumann

Eingereicht am: 1. Januar 2015

Jan Lepel

Thema der Arbeit

Automatisierte Erstellung und Provisionierung von ad hoc Linuxumgebungen -
Prototyp einer Weboberfläche zur vereinfachten Inbetriebnahme individuell erstellter Entwicklungsumgebungen

Stichworte

Ad hoc Umgebung, automatisierter Umgebungsaufbau und Provisionierung

Kurzzusammenfassung

Dieses Dokument ...

Jan Lepel

Title of the paper

TODO

Keywords

Keywords, Keywords1

Abstract

This document ...

Listings

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | Anforderungsanalyse | 1 |
| 1.1 | Bestandsaufnahme existierender Lösungen | 1 |
| 1.2 | Zielsetzung | 1 |
| 1.2.1 | Stakeholder | 2 |
| 1.2.2 | Funktionale Anforderungen | 2 |
| 1.2.3 | Nichtfunktionale Anforderungen | 4 |

1 Anforderungsanalyse

Die Anforderungsanalyse hilft Systemeigenschaften und Systemanforderungen der einzelnen beteiligten Gruppen, auch als Stakeholder bezeichnet, zu erfassen, zu analysieren und ggf. eine Diskussionsgrundlage zu schaffen. Das resultierende Ergebnis, kann dann wiederum als Grundstein für ein zukünftiges Lastenheft genutzt werden.

Im Folgenden werden die Anforderungen an die Software detaillierter konzipiert, aufgelistet und beschrieben. Zudem grundlegende Fragen bezüglich der Kernaufgaben, Qualitätsanforderungen, beteiligte Gruppen und benachbarter Systeme geklärt.

1.1 Bestandsaufnahme existierender Lösungen

DAS EHER NACH HINTEN?!

1.2 Zielsetzung

Die zu erstellende Applikation unterstützt den Anwender bei der Umsetzung und der Konfiguration von virtuellen Entwicklungsumgebungen. Angestrebte Funktionalitäten, wie der Aufbau einer Entwicklungsumgebung inklusive der automatisierten Installation von Programmen, oder der Austausch von bereits erstellten Entwicklungsumgebungen zwischen beteiligten Benutzern, sollen dem Anwender in seinen Tätigkeiten hilfreich sein. Durch ein übersichtliches Konfigurationsspektrum, soll es dem Anwender vereinfacht werden, sich innerhalb kürzester Zeit in die Anwendung einzufinden um sich die gewünschte virtuelle Umgebung mit geringem Zeitaufwand zu erstellen. Um die aus dieser Anforderungsbeschreibung hervorgehenden Kernfunktionalitäten und Qualitätsmerkmale näher zu betrachten, werden als erstes die Stakeholder definiert und ihnen dann die Kernfunktionalitäten zugeordnet. Im weiteren wird auf die Qualitätsmerkmale eingegangen, die auch als nichtfunktionale Anforderungen bezeichnet werden und als Qualitätskriterium an System und Software angesehen werden können.

1.2.1 Stakeholder

'Stakeholder in Softwareentwicklungsprojekten sind alle Personen (oder Gruppen von Personen) die ein Interesse an einem Softwarevorhaben oder dessen Ergebnis haben.' Zörner (2012)

| | |
|--------------|--|
| Rolle | Anwender |
| Beschreibung | Ein Anwender ist ein Benutzer des Systems, ohne administrative Einflüsse auf die Applikation. |
| Ziel | Gute Benutzbarkeit, schnell erlernbar, komfortable Steuerung, leichter Aufbau der gewünschten Umgebung |

| | |
|--------------|--|
| Rolle | Administrator |
| Beschreibung | Der Administrator kann die Applikation wie der Anwender nutzen. Er hat erweiterte Möglichkeiten, im Bezug auf die Konfiguration des Systems. |
| Ziel | leicht ersichtliche Konfigurationsmöglichkeiten, schnelles auffinden von auftretenden Fehlern, gut protokollierte Fehler |

Abbildung 1.1: Stakeholder

1.2.2 Funktionale Anforderungen

Anforderungen Anwender

- FA 1 Die Anwendung muss über den Browser ausführbar sein zu, ohne zusätzliche lokale Installationen auf Anwenderseite.
- FA 2 Falls der Anwender keine zusätzliche Software auf der virtuellen Maschine haben möchte, muss die Anwendung eine entsprechende Option dafür anbieten.
- FA 3 Die Anwendung muss dem Anwender die Möglichkeit bieten, Software mit zu installieren.
- FA 4 Falls der Anwender diese Option nutzt, sollte die Anwendung Softwarekomponenten vorschlagen oder es ermöglichen eigene Dateien auszuwählen.
- FA 4 Die Anwendung sollte fähig sein, dem Administrator Bearbeitungsmöglichkeiten für die vorzuschlagenden Softwarekomponenten anzubieten.

FA 6 Die Anwendung sollte dem Anwender die Option bieten, virtuelle Maschinen zu exportieren.

FA 5 Die Anwendung sollte fähig sein den Export zu komprimieren.

FA 5 Ist der Export durchgeführt worden, muss die Anwendung das mit einer Meldung auf dem Bildschirm bestätigen.

FA 5 Die Anwendung muss fähig sein, Exporte wieder importieren zu können Falls während der Importierung Datenfehler auftreten, muss die Anwendung den jeweiligen Fehler (Fehlerbeschreibung) auf dem Bildschirm ausgeben.

FA 5 Ist der Import erfolgreich durchgeführt worden, sollte die Anwendung eine entsprechende Meldung anzeigen.

FA 7 Wenn der Anwender eine virtuelle Maschine erstellen möchte, muss die Anwendung bei wichtigen Konfigurationsschritten, für den Benutzer sichtbare Statusmeldungen anzeigen.

FA 5 Treten Fehler bei der Erstellung einer virtuellen Maschine auf, muss das System eine Fehlermeldung ausgeben.

Anforderungen Administrator

FA 8 Falls während des Betriebes der Anwendung Änderungen an der Konfiguration durchgeführt werden müssen, sollte die Anwendung dies nur durch einen expliziten Administratoren-Account zulassen.

FA 8 Solange der Administrator angemeldet ist, sollte die Anwendung ihm die Möglichkeit bieten, den Speicherort und Name von Logdateien persistent zu ändern.

FA 5 Falls während der Änderung ein Fehler auftritt, muss die Anwendung eine Fehlermeldung auf dem Bildschirm ausgeben.

FA 8 Die Anwendung sollte dem Administrator die Option bieten, sich den Inhalt von Logdateien anzeigen zu lassen.

Definition Funktionale Anforderungen nach **Rupp und die SOPHISTen (2014)**

1.2.3 Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen können als Qualitätskriterium an System und Software angesehen werden, wobei die Qualität der Software/des Systems immer auf die folgenden Qualitätsmerkmale bezogen wird.

1. **Funktionalität** Die geforderten Funktionen sind in der Software umgesetzt worden und erfüllen die definierten Aufgaben und Ziele.
2. **Zuverlässigkeit** Die Software kann unter bestimmten Bedingungen und über einen bestimmten Zeitraum ein definiertes Leistungsniveau aufrechterhalten.
3. **Benutzbarkeit** Analyse der Benutzung und des Aufwandes. Der Benutzer steht hier im Fokus.
4. **Effizienz** Das Verhältnis zwischen der zu erbringenden Leistung der Software und der dafür zugrundeliegenden Betriebsmittel.
5. **Änderbarkeit** Der Aufwand, der betrieben werden muss um Änderungen an der Software vorzunehmen. Änderungen sind definiert als Korrekturen, Verbesserungen oder Anpassungen an Veränderungen der Umgebung, der Anforderungen oder der funktionalen Spezifikationen.
6. **Übertragbarkeit** Der Aufwand, der zum Übertragen oder Implementieren in eine neue Umgebung benötigt wird. Als Umgebung ist sowohl Hardware- als auch Softwareumgebung gemeint.

(Qualitäten nach ISO/IEC 9126 (DIN 66272))

Das Bilden von Qualitätsszenarien hilft bei der Erfassung von Qualitäten und deren Sortierung zu entsprechenden Qualitätsmerkmalen. Zusätzlich können neue Anforderungen aufgedeckt werden und leichter Anwendungsfälle abgeleitet werden.

Qualitätsszenarien

- NFA 1 Ein Benutzer ohne Vorkenntnisse muss bei der erstmaligen Verwendung des Systems innerhalb von 10 Minuten in der Lage sein, die gewünschte Funktionalität zu lokalisieren und zu verwenden.
- NFA 2 Die Anzahl der gleichzeitig laufenden virtuellen Umgebungen, ist maximal bei 10.
- NFA 3 Jeder auftretende Fehler ist eindeutig identifizierbar und nachvollziehbar.

- NFA 4 Die Architektur soll so erweiterbar sein, dass es zukünftig möglich ist, mehrere Maschinen gleichzeitig aufbauen zu können.
- NFA 5 Das Importieren von Konfigurationsdateien beträgt <2 Minuten.
- NFA 6 Die Validierung einer YAML-Datei beträgt <1 Minute.
- NFA 7 Der bei Import verwendete Validierungsalgorithmus muss unter 0.5 Personentagen austauschbar sein.
- NFA 8 Der Export von Konfigurationsdateien beträgt <1 Minute.
- NFA 9 Änderungen an der Softwarekomponentenliste muss innerhalb von <10 Sekunden in der GUI sichtbar sein.
- NFA 10 Das Betriebssystem soll auf eine höhere Version aktualisiert werden. Dies kann ohne Änderungen des Quellcodes der Software vorgenommen werden.
- NFA 11 Soll ein anderer Provisionierer verwendet werden, liegt der Aufwand des Austausches bei unter einem Personentag.
- NFA 12 Die Implementierung weiterer Grundfunktionalitäten soll möglichst einfach erfolgen.
- NFA 13 Softwareupdates der benutzen Softwarekomponenten müssen mit Rücksprache des Entwicklers erfolgen.
- NFA 14 Die Kommunikation zwischen Frontend und Backend muss nicht verschlüsselt ablaufen.
- NFA 15 Die Anwendung sollte zu 99.0 Prozent der Zeit lauffähig sein.

See also ?.

Literaturverzeichnis

- [Rupp und die SOPHISTen 2014] RUPP, Chris ; SOPHISTEN die: *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil.* 6., aktualisierte und erweiterte Auflage. Carl Hanser Verlag GmbH und Co. KG, 10 2014. – URL <http://amazon.de/o/ASIN/3446438939/>. – ISBN 9783446438934
- [Zörner 2012] ZÖRNER, Stefan: *Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten.* Carl Hanser Verlag GmbH und Co. KG, 5 2012. – URL <http://amazon.de/o/ASIN/3446429247/>. – ISBN 9783446429246

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 1. Januar 2015 Jan Lepel
