



Individuelle Produktivarbeit

Schulungs- & Temporär-Laptop Umgebung

Thomas Galliker

Schindler Informatik AG
Zugerstrasse 13
CH-6030 Ebikon

Nat. ++41 (0) 79 504 80 70
Tel. ++41 (0) 41 493 17 60
Fax ++41 (0) 41 493 00 42

I. Inhaltsverzeichnis

I.	Inhaltsverzeichnis	2
II.	Tabellenverzeichnis	5
III.	Abbildungsverzeichnis	5
1	Management Summary	6
1.1	Organisation	6
1.2	Projektübersicht	6
1.3	Lehrbetrieb	6
1.4	Involvierte Personen	6
2	Aufgabenstellung	7
2.1	Ausgangslage	7
2.2	Zielsetzung	7
2.3	Rahmenbedingungen	7
2.3.1	Systemgrenzen	7
2.3.2	Hilfsmittel	7
2.3.3	Vorkenntnisse	8
2.3.4	Vorarbeiten	8
2.3.5	Kosten	8
2.3.6	Arbeitsumgebung	8
3	Projektplanung	9
3.1	Soll-Zeitplan	9
3.2	Risiken	10
3.2.1	Externe Einflüsse	10
3.2.2	Datenverlust	10
3.2.3	Falsche Zeiteinschätzung	10
3.2.4	Krankheit / Unfall	10
4	Analyse	11
4.1	Auftragsdivision	12
4.1.1	Teilaufgabe 1: Entwicklung des Preinstallation Environments	12
4.1.2	Teilaufgabe 2: Integration der Authentifizierungsanwendung	12
4.1.3	Teilaufgabe 3: Entwicklung der Setupanwendung	12
4.1.4	Teilaufgabe 4: Integration der Softwareverteilung	12
4.2	Analyse des Gesamtsystems	13
4.2.1	Benutzerschnittstelle	13
4.2.2	Entwicklung	13
4.2.3	Chronologischer Ablauf	13
4.2.4	Informationsakquisition	13
4.3	Evaluation des Preinstallation Environments	14
4.3.1	Evaluierte Produkte	14
4.3.2	Entscheidungskriterien	14
4.3.3	Entscheidungsmatrix	14
4.3.4	Auswertung	15
4.4	Evaluation der RAMDisk-Produkte	16
4.4.1	Evaluierte Produkte	16
4.4.2	Entscheidungskriterien	16
4.4.3	Entscheidungsmatrix	16
4.4.4	Auswertung	16
4.5	Evaluation der Installationsmethode	18
4.5.1	Methode 1: Unattended Windows Installation	18
4.5.2	Methode 2: Imaging mit Ghost	18
4.5.3	Methode 3: Imaging mit Ximage	18
4.5.4	Entscheidungskriterien	19
4.5.5	Entscheidungsmatrix	19
4.5.5.1	Muss-Analyse	19
4.5.5.2	Kann-Analyse	19
4.5.6	Auswertung	19
5	Realisierung	21
5.1	Entwicklung des Preinstallation Environments	21
5.1.1	Entwicklungsumgebung vorbereiten	21
5.1.1.1	Verzeichnisstruktur	21

5.1.1.2	OPK Tools einrichten	21
5.1.1.3	Betriebssystem Quelle vorbereiten	21
5.1.2	Windows PE Basis-Image erstellen	22
5.1.3	RAMDisk integrieren	22
5.1.3.1	Problemstellung	22
5.1.3.2	Integrationsvorgang	22
5.1.4	Tastaturlayout anpassen	23
5.1.4.1	Problemstellung	23
5.1.4.2	Sprachcode Referenz	23
5.1.4.3	Integrationsvorgang	23
5.1.5	Shutdown-Tool integrieren	24
5.1.5.1	Problemstellung	24
5.1.5.2	Lösung	24
5.1.5.3	Integrationsvorgang	24
5.1.6	Scripting Components	24
5.1.6.1	Beschreibung	24
5.1.6.2	Integrationsvorgang	25
5.1.7	Netzwerktreiber integrieren	25
5.1.7.1	Problemstellung	25
5.1.7.2	Voraussetzungen / Empfehlungen	25
5.1.7.3	Integrationsvorgang	25
5.1.8	ISO Image erstellen	26
5.2	Realisierung der Hypertext-Anwendungen	27
5.2.1	Factory Components	27
5.2.1.1	Beschreibung	27
5.2.1.2	Entwicklungsvorgehen für Phase 1	27
5.2.1.3	Entwicklungsvorgehen für Phase 2	27
5.2.1.4	Integrationsvorgang	28
5.2.2	Login Form	28
5.2.2.1	Eigenschaften	28
5.2.2.2	Modifikationen	28
5.2.2.3	Integrationsvorgang	29
5.2.3	Setup Assistant	29
5.2.3.1	Entwicklungsvorgehen	29
5.2.3.2	Funktionen	30
5.2.3.3	Datenfluss	31
5.2.3.4	Integrationsvorgang	32
5.3	Integration der Softwareverteilung	33
5.3.1	Container für Software-Sets erstellen	33
5.3.2	Software-Set erstellen	33
5.3.3	Software-Set konfigurieren	33
6	Testphase	34
6.1	Ziel der Testphase	34
6.2	Planung der Tests	34
6.2.1	Methoden	34
6.2.2	Mittel	34
6.3	Testserie 1: Entwicklertest	35
6.3.1	Komponententest 1	35
6.3.2	Komponententest 2	35
6.3.3	Integrationstest	36
6.3.4	Sicherheitstest	36
6.3.5	Systemtest	37
6.4	Testserie 2: Benutzertest	39
6.5	Testserie 3: Akzeptanztest	40
7	Projektauswertung	41
7.1	Planungsauswertung	41
7.1.1	Ist-Zeitplan	41
7.1.2	Planungsdiskrepanz	41
7.1.3	Kommentar	41
7.2	Persönlicher Erfahrungsbericht	42
8	Arbeitsprotokolle	43
8.1	Wochenübersicht	43
8.1.1	Woche 1	43

8.1.2	Woche 2	44
8.1.3	Woche 3	45
8.2	Tagesprotokolle	46
8.2.1	Tagesprotokoll vom Montag, 13.03.2006.....	46
8.2.2	Tagesprotokoll vom Dienstag, 14.03.2006.....	46
8.2.3	Tagesprotokoll vom Mittwoch, 15.03.2006.....	47
8.2.4	Tagesprotokoll vom Donnerstag, 16.03.2006	47
8.2.5	Tagesprotokoll vom Freitag, 17.03.2006	48
8.2.6	Tagesprotokoll vom Montag, 20.03.2006.....	48
8.2.7	Tagesprotokoll vom Dienstag, 21.03.2006.....	49
8.2.8	Tagesprotokoll vom Mittwoch, 22.03.2006.....	49
8.2.9	Tagesprotokoll vom Donnerstag, 23.03.2006	50
8.2.10	Tagesprotokoll vom Freitag, 24.03.2006	50
8.2.11	Tagesprotokoll vom Montag, 27.03.2006.....	51
8.2.12	Tagesprotokoll vom Dienstag, 28.03.2006.....	51
8.2.13	Tagesprotokoll vom Mittwoch, 29.03.2006.....	52
8.2.14	Tagesprotokoll vom Donnerstag, 30.03.2006	52
8.2.15	Tagesprotokoll vom Freitag, 31.03.2006	53
8.3	Sitzungsprotokolle	54
8.3.1	Sitzung 1: Kickoff Meeting.....	54
8.3.2	Sitzung 2: Expertenbesuch	55
8.3.3	Sitzung 3: Besprechung.....	56
	Sitzung 4: Projektübergabe.....	56
8.4	Problemprotokoll.....	57
8.4.1	Problem 1 – HTA Interface Refresh Problem.....	57
8.4.2	Problem 2 – Mking.cmd Problem	57
8.4.3	Problem 3 - Partitionierungsproblem	57
8.4.4	Problem 4 – Fehlender NIC-Treiber	57
8.4.5	Problem 5 – CD-RW Problem	57
8.4.6	Problem 6 – Unity Asset Informationen	57
8.4.7	Problem 7 – Latenzzeiten	58
9	Glossar.....	59
10	Anhang.....	62
10.1	Anhang A – Software Assurance Lizenzkosten.....	62
10.2	Anhang B – Quellcode	63
10.2.1	Login Form	63
10.2.1.1	Login.hta	63
10.2.1.2	Config.ini.....	70
10.2.2	Factory Components.....	72
10.2.2.1	Startnet.cmd.....	72
10.2.2.2	Factory.hta	72
10.2.3	Setup Assistant.....	75
10.2.3.1	Setup.hta	75
10.2.3.2	Config.ini.....	97
10.3	Anhang C – RAMDisk Konfigurationsdatei	98
10.4	Anhang D – Ist-Zeitplan.....	99
10.5	Anhang E – Detailliertes Ablaufdiagramm	100
10.6	Anhang F – OPK Hilfe	101
10.7	Anhang G – Netzwerkanalyse.....	101
10.8	Anhang H – Hypertext Anwendungen	101
10.9	Anhang I – Präsentation.....	101
10.10	Anhang J – Quellenverzeichnis	101
10.10.1	Generelle Quellen.....	101
10.10.2	Referenzierte Quellen	102

II. Tabellenverzeichnis

Nummer	Beschreibung	Seite
Tabelle 1:	Übersicht PE Produkte.....	14
Tabelle 2:	Kriterienkatalog der PE Evaluation.....	14
Tabelle 3:	Analyse der PEs.....	14
Tabelle 4:	Auswertung PE Evaluation.....	15
Tabelle 5:	Übersicht RAMDisk-Produkte.....	16
Tabelle 6:	Kriterienkatalog der RAMDisk Evaluation.....	16
Tabelle 7:	Analyse der RAMDisk-Produkte.....	16
Tabelle 8:	Auswertung der RAMDisk Evaluation.....	16
Tabelle 9:	Kriterienkatalog der Installationsmethoden.....	19
Tabelle 10:	Muss-Analyse der Installationsmethoden.....	19
Tabelle 11:	Kann-Analyse der Installationsmethoden.....	19
Tabelle 12:	Auswertung der Installationsmethoden-Evaluation.....	20
Tabelle 13:	Verzeichnisstruktur auf dem Entwicklungssystem.....	21
Tabelle 14:	Konfigurationsdatei "ramdisk.inf".....	23
Tabelle 15:	Sprachcode Referenztafel.....	23
Tabelle 16:	Konfigurationsdatei "keyboard.inf".....	24
Tabelle 17:	Änderung an der Konfigurationsdatei "Config.inf".....	24
Tabelle 18:	Phase 1: Initialisierungseinträge zum Start von Phase 2.....	27
Tabelle 19:	Funktionen der Anwendung Setup Assistant.....	30
Tabelle 20:	Bemerkungen zum Informationsfluss in Setup Assistant.....	31
Tabelle 21:	Protokoll des Komponententests 1.....	35
Tabelle 22:	Protokoll des Komponententests 2.....	35
Tabelle 23:	Protokoll des Integrationstests.....	36
Tabelle 24:	Protokoll des Sicherheitstests.....	36
Tabelle 25:	Protokoll des Systemtests.....	37
Tabelle 26:	Benutzertest: Protokoll des Systemtests.....	39
Tabelle 27:	Akzeptanztest.....	40
Tabelle 28:	Planungsauswertung mit Abweichungsberechnung.....	41
Tabelle 29:	RAMDisk Konfigurationsdatei "ramdisk.inf".....	98
Tabelle 30:	Hexadezimale "DiskSize"-Werte für RAMDisk.....	98

III. Abbildungsverzeichnis

Nummer	Beschreibung	Seite
Abbildung 1:	Abstrakte Ansicht des Ablaufdiagramms.....	13
Abbildung 2:	Auszug aus einer Windows Setupkonfigurationsdatei (Unattend.txt).....	18
Abbildung 3:	Imaging-Software "Ghost 9.0".....	18
Abbildung 4:	Command-Line Imagingsoftware "Ximage".....	18
Abbildung 5:	Shutdown-Tool für Windows PE: "ShutdownPE.exe".....	24
Abbildung 6:	Driver Injection Tool zur Treiberintegration.....	26
Abbildung 7:	Ursprüngliche Benutzeroberfläche der Login-Anwendung.....	28
Abbildung 8:	Überarbeitete Benutzeroberfläche der Login-Anwendung.....	29
Abbildung 9:	Setup Assistant, Entwurfsansicht.....	29
Abbildung 10:	Informationsakquisition.....	31
Abbildung 11:	Lese- und Schreibberechtigung für Unity Administratoren.....	32
Abbildung 12:	Leseberechtigung für Unity Supportmitarbeiter.....	32
Abbildung 13:	Erstellen der OU „EDU_INF“.....	33
Abbildung 14:	Erstellen des Software-Sets „Autodesk CAD Training“.....	33
Abbildung 15:	Kerberos5 Authentifizierung mit Login Form.....	37
Abbildung 16:	Auswertungsdiagramm mit Ist-/Soll-Vergleich.....	41
Abbildung 17:	Mail von D.Popp: Windows Server 2003 SA Lizenzkosten.....	62

1 Management Summary

1.1 Organisation

Die Experten bilden zusammen mit dem Fachausbildner und dem Kunden den Auftraggeber. Zusammen sind sie für die Formulierung der Aufgabenstellung und Bewertung der Projektarbeit als Ganzes zuständig.

1.2 Projektübersicht

Diese Dokumentation, die daraus entstandene Präsentation sowie das anschliessende Fachgespräch sind Bewertungsgrundlagen für meine praktische Lehrabschlussprüfung (Individuelle Produktivarbeit, kurz IPA).

Nach den Standards des Projekt Managements werden folgende Projektphasen dokumentiert:

- ? Aufgabenstellung
- ? Planung
- ? Analyse
- ? Realisierung
- ? Testphase
- ? Auswertung

Zusätzliche Informationen, welche in der nachfolgenden Dokumentation zu finden sind:

- ? Abbildungsverzeichnis
- ? Tabellenverzeichnis
- ? Quellenverzeichnis
- ? Protokolle
- ? Glossar
- ? Anhang

1.3 Lehrbetrieb

Schindler Informatik AG
Zugerstrasse 13
6031 Ebikon

Telefon: +41 41 445 34 34
Fax: +41 41 443 38 86

1.4 Involvierte Personen

Auszubildender:	Galliker Thomas	+41 79 504 80 70
Fachausbildner:	Gehrig Patrick	+41 41 445 57 25
Kunde:	Stefan Epp	+41 41 445 38 80
Kunde (Stv.):	Michael Jenny	+41 41 445 38 28

2 Aufgabenstellung

2.1 Ausgangslage

Die Abteilung DC-WIN der Schindler Informatik AG entwickelt, betreut und beaufsichtigt die Desktop-Client Installationen in zahlreichen europäischen Schindler Konzerngesellschaften (KG). Zu den Aufgabenbereichen gehören hauptsächlich die Verteilung von Desktop-Betriebssystemen bzw. das Entwickeln neuer Verteilungsmethoden, die Softwareverteilung sowie die Software Paketierung.

Aufgrund veralteter Hardware und teurer Raumkosten wurde der bestehende Schulungsraum in der Schindler Informatik abgebaut. Der Raum wird heute als Arbeitsplatzfläche genutzt.

Nun wird eine neue Schulungsumgebung gewünscht, welche mobil in Sitzungszimmern sowie auch losgelöst vom Schindler Netzwerk genutzt werden kann.

2.2 Zielsetzung

Die Investition für die Laptops wurde an die Bedingung geknüpft, dass diese auch für temporäre und Notfall-Einsätze genutzt werden können. Zudem sollen sie dem "Schindler Informatik Standard-Client" entsprechen. Die neuen Laptops wurden bereits angeschafft.

Die Liste der Anforderungen wird nachfolgend ausführlich illustriert:

- ? Das Aufsetzen soll möglichst wenig Interaktion erfordern und in die bestehende Softwareverteilungsumgebung (Unity) integriert werden.
 - o Vollautomatische, komplette Partitionierung und Formatierung der lokalen Systemdisk (Partition C und D).
 - o Zurücksetzen der Unity-Konfiguration, resp. anpassen an die Anforderung der kommenden Schulung.
 - o Bereinigen des Userhomes des betreffenden Edu-Users (Daten und Profile).
- ? Die Laptops sollen auch offline funktionieren.
- ? Ein einzelner Laptop soll auch als Standard-Client der Schindler Informatik funktionieren.
- ? Zu schulendes 'Software-Set' soll in einer Form als 'Gruppe' definiert und beim Installieren ausgewählt werden können.
- ? Edu-User und Lehrer besitzen nur User-Rechte.
- ? Administration wird über den Schindler Informatik Support abgewickelt.
 - o Verantwortung Problem Management und Asset Configuration:
 - ✍ INF-DC-CSC (1st Level Support)
 - o Verantwortung System Management und Installation:
 - ✍ INF-DC-EUC (2nd Level Support)
- ? Zukünftige Betriebssystem Updates müssen zu implementieren sein.
- ? Umfang und Aufwand der Arbeit entsprechen den Vorgaben von 80 Arbeitsstunden inklusive Dokumentation und Präsentation.

2.3 Rahmenbedingungen

2.3.1 Systemgrenzen

Für die Bewerkestellung der praktischen Arbeit steht neben der herkömmlichen Arbeitsplatzausstattung folgendes Equipment zur Verfügung:

- ? Windows Betriebssysteme.
- ? Funktionsfähige Softwareverteilung (Unity).
- ? Sämtliche Hardware ist vorhanden und soll bestmöglich genutzt werden.
- ? Das Standard Software-Set der Schindler Informatik kann genutzt werden.
- ? Zusätzliche, spezielle Software muss mit Kunde abgestimmt werden.
- ? Bestehende Software Lizenzen können genutzt, neue sollen keine beschafft werden.
- ? Sprache für GUI's ist Englisch.

- ? Logistische Verwaltung der Hardware (Case).
- ? Netzwerk (inkl. Verkabelung, Router, Switches).
- ? Netzwerkdienste des Schindler Netzwerks (DHCP, DNS,...).

2.3.2 Hilfsmittel

Zur Realisierung der Arbeit sind sämtliche Hilfsmittel erlaubt, welche im Rahmen der IPA zugelassen sind. Inbegriffen sind sowohl bestehende (eigene und unter Vermerkung auch fremde) Dokumentationen, Datenbanken, Code und Literatur, welche in der Abteilung vorzufinden sind, als auch das Internet und

andere, deklarierte Quellen. Befragungen von Mitarbeiter und/oder externen Hilfspersonen müssen nachweisbar dokumentiert werden.

Primäres Dokumentationsmittel ist Microsoft Word. Zusätzliche Hilfsprogramme wie Microsoft Visio, Project und Powerpoint sind ebenfalls zugelassen. Bildschirmaufzeichnungen können mit der Software SnagIt durchgeführt werden.

Zur Bewerksstellung der praktischen Arbeiten steht sämtliche Software und Hardware zur Verfügung, welche im Rahmen der geschäftlichen und/oder privaten Lizenz genutzt werden darf. Die Nutzung von lizenzfreier Software ist an entsprechender Stelle zu deklarieren.

Tests können mit Hilfe der Virtualisierungslösung VMware GSX Server (Freeware Version) durchgeführt werden. Entsprechende Testsysteme werden zur Verfügung gestellt.

2.3.3 Vorkenntnisse

Eine transparente Bewertung der durchgeführten Projektarbeit erfordert die Preisgabe der fundierten Vorkenntnisse. Projektrelevante Vorkenntnisse werden stichwortartig festgehalten:

- ? Windows 2000/XP/2003 Systemkenntnisse.
- ? Windows 2000/XP/2003 Unattended Setup.
- ? Windows Registry.
- ? Softwareverteilung mit Unity.
- ? Softwareentwicklung: C, C#.
- ? Scripting: DOS Batch, VBScript, HTML.

2.3.4 Vorarbeiten

Um den vorgegebenen Projektaufwand von 80 Stunden nicht zu überschreiten, ist es wichtig, einige Vorarbeiten zu leisten:

- ? Teile der Dokumentation sowie der Dokumentationsvorlage wurden bereits in früheren Projekten erstellt und mussten für die IPA lediglich angepasst werden.
- ? Vorlage der Powerpoint Präsentation wurde vom Lehrbetrieb zur Verfügung gestellt.
- ? Hardware für Testsysteme wurde bereits assembliert und installiert.
- ? VMware GSX Server (Freeware Version) wurde auf Testsystem vorinstalliert.

2.3.5 Kosten

Aufgrund des geplanten Zeitaufwands und der Tatsache, dass sämtliche Hard- und Software zur Bewerksstellung des Projekts bereits vorhanden ist, stellen sich die Kosten wie folgt zusammen:

Zeitaufwand (Total)	Stundensatz	Kostenaufwand (Total)
80h	Junior, 139 SFr/h	11'120 SFr

2.3.6 Arbeitsumgebung

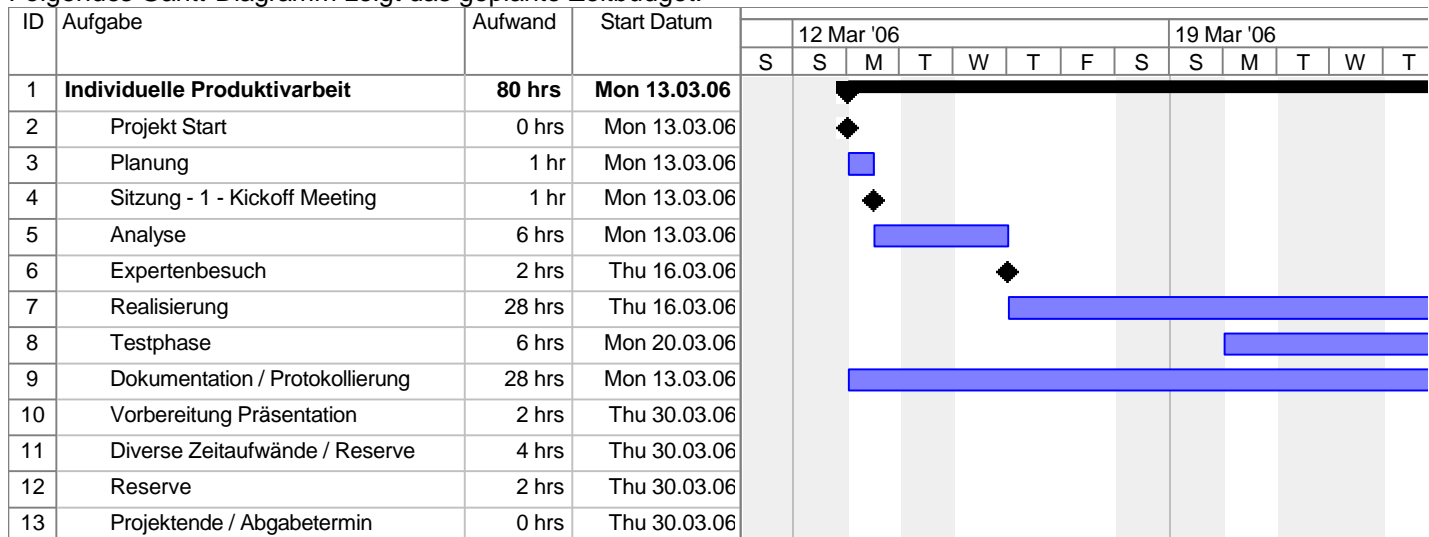
Arbeitsort ist das Büro 0013 im Gebäude der Schindler Informatik AG.

3 Projektplanung

Die Planung stellt ein wichtiges Hilfsmittel dar und wurde deshalb äusserst sorgfältig erstellt. Hier kann während der Projektzeit mit der effektiv benötigten Zeit übereinstimmt. Gegebenenfalls können Zeitknappheiten frühzeitig erkannt und Projekt

3.1 Soll-Zeitplan

Folgendes Gantt-Diagramm zeigt das geplante Zeitbudget:



3.2 Risiken

3.2.1 Externe Einflüsse

Die Projektarbeit ist teilweise abhängig von externen Einflüssen, welche die planmässige Durchführung gefährden könnten. Zu den externen Einflüssen gehören beispielsweise Server, Netzwerk, Internetzugang, Stromversorgung, usw.

3.2.2 Datenverlust

Um dem Verlust von projektrelevanten Daten vorzubeugen, werden diese regelmässig gesichert. Es besteht jedoch ein geringes Restrisiko, dass Daten während der Projektarbeit verloren gehen können.

3.2.3 Falsche Zeiteinschätzung

Aus Erfahrung kann ich davon ausgehen, dass der geplante Zeitaufwand nicht genau eingehalten werden kann. Damit das Projekt trotzdem planungsgemäss erledigt werden kann, wurden Zeitreserven in die Planung mit einbezogen.

3.2.4 Krankheit / Unfall

Die Projektarbeit kann aus gesundheitlichen Gründen jederzeit unterbrochen werden. Der Endtermin des Projekts verschiebt sich entsprechend der verlorenen Zeit.

4 Analyse

Die Analyse ist der erste Schritt in einem Herstellungsprozess. Hier wird untersucht, was das zu erstellende "Produkt" leisten soll. Daher ist es äusserst wichtig, dass die Aufgabenstellung korrekt interpretiert wird. Im Nachfolgenden werden sowohl Produkt- als auch Methoden-Analysen durchgeführt.

Zur Lösungsfindung wurden verschiedene Analyseverfahren angewandt. Jede analysierte Lösung wurde einer Muss- und/oder Kann-Analyse unterzogen. Nachfolgend werden Details zu den Verfahren erläutert:

? **Muss-Analyse**

Durch das Gegenüberstellen verschiedener Lösungsvarianten werden in einem ersten Schritt alle Lösungsansätze herausgefiltert, welche die Aufgabe mindestens befriedigend lösen können. Lösungsvarianten, welche ein Muss-Kriterium aus einem Grund nicht erfüllen können, werden im weiteren Verlauf der Lösungsfindung nicht mehr weiter berücksichtigt. Falls eine Entscheidung bereits während der Muss-Analyse fällt, so ist es nicht nötig, zusätzlich eine Kann-Analyse durchzuführen.

? **Kann-Analyse**

In diesem Analysevorgang wird geprüft, wie gut die analysierten Lösungsvarianten zusätzliche Kriterien erreichen können. Dabei werden wiederum alle Varianten mit allen Kann-Kriterien gegenübergestellt. Im Gegensatz zu der Muss-Analyse werden die Varianten mit einer Punktvorgabe rangiert. Die Priorisierung der Kriterien wird mit einer Gewichtung vorgenommen.

4.1 Auftragsdivision

Die Realisierung des Projektauftrags teilt sich in folgende Teilaufgaben:

4.1.1 Teilaufgabe 1: Entwicklung des Preinstallation Environments

Die Teilaufgabe 1 umfasst die Erstellung und Modifizierung eines Preinstallation Environments. Diese Preinstallation Environment dient im Projektsystem als Konfigurationsplattform. Darauf aufbauend werden später die Authentifizierungsanwendung und die Setupanwendung ausgeführt.

4.1.2 Teilaufgabe 2: Integration der Authentifizierungsanwendung

Eine bereits existierende Authentifizierungsanwendung dient der Authentifizierung von Anmeldeinformationen. Die Erstellung dieser Anwendung ist nicht Bestandteil dieses Projekts. Die Teilaufgabe 2 verfolgt als Ziel die Adaption der bestehenden Authentifizierungsanwendung an das Projektsystem: Dies umfasst sowohl die Überarbeitung der Benutzeroberfläche als auch die Anpassung des Quellcodes an das Projektsystem.

4.1.3 Teilaufgabe 3: Entwicklung der Setupanwendung

Die Benutzerschnittstelle dieses Projekts stellt eine Setupanwendung dar. Die Entwicklung dieser Setupanwendung nach den Anforderungen des Kunden ist zugleich eine der grössten Herausforderungen im ganzen Projekt.

4.1.4 Teilaufgabe 4: Integration der Softwareverteilung

Teilaufgabe 4 behandelt die Integration des Projektsystems in die Softwareverteilung Unity. Das Projektsystem bildet eine direkte Schnittstelle zu sämtlichen Konfigurationsinformationen von Unity. Einige marginale Änderungen und/oder Neuerungen in der Softwareverteilungsumgebung sind daher vorhersehbar.

4.2 Analyse des Gesamtsystems

4.2.1 Benutzerschnittstelle

Der Setupassistent soll nach dem Prinzip eines "Wizards" aufgebaut werden. Ein Assistent (nachfolgend „Setup Assistant“ genannt) führt den Benutzer durch die Installation. Die Benutzeroberfläche ist übersichtlich und benutzerfreundlich zu gestalten. Als Grundlage für die Gestaltung der Benutzeroberfläche dient das Design des neuen Schindler Corporate Identity (CI). Die Sprache wird gemäss Aufgabenstellung „Englisch“ sein.

4.2.2 Entwicklung

- ? Sämtliche in einem Preinstallation Environment lauffähigen Software-Tools und Scriptsprachen können verwendet werden.
- ? Es wird beabsichtigt, die Anwendung mit VisualBasic Script (VBS) zu erstellen, da diese leistungsfähige Scriptsprache in beiden evaluierten Preinstallation Environment lauffähig ist.
- ? Die erstellte Anwendung wird in einer RAMDisk ausgeführt. Der Einsatz einer RAMDisk erhöht die Funktionalität der Anwendung immens: Einige Scriptfunktionen können nicht direkt vom Medium des Preinstallation Environments ausgeführt werden und müssen deshalb auf ein Speichermedium verschoben werden, auf welchem sie exklusiven Schreibzugriff erhalten
- ? Die grafische Benutzerschnittstelle wird mittels Hypertext Application (HTA) realisiert.

4.2.3 Chronologischer Ablauf

Der chronologische Setupablauf wurde minutiös durchdacht. Nachfolgender Ablauf zeigt die abstrakte Ansicht des Ablaufdiagramms. Eine detaillierte Ansicht desselben Diagramms kann im [Anhang E](#) betrachtet werden.

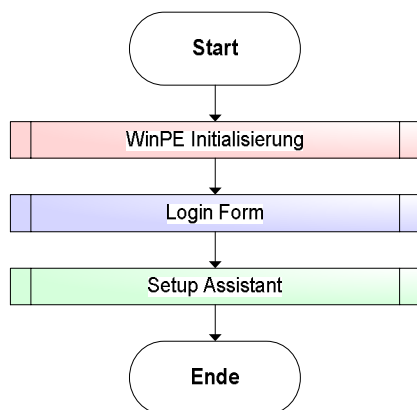


Abbildung 1: Abstrakte Ansicht des Ablaufdiagramms.

4.2.4 Informationsakquisition

- ? Die aktuelle Betriebssystemquelle liegt auf einer Netzwerkfreigabe des produktiven Softwareverteilungsservers.
- ? Die Konfigurationsinformationen der Client Computer (sog. "Asset Information") liegen in einem Verzeichnis auf dem Softwareverteilungsserver und werden in Form von INI-Dateien bereitgestellt.
- ? Die Informationen zu den verschiedenen Software-Set's werden von der Softwareverteilung "Unity" in einer INI-Datei bereitgestellt.
- ? Lokale Hardwareinformationen können via Windows Management Instrumentation (WMI) ausgelesen werden.

4.3 Evaluation des Preinstallation Environments

4.3.1 Evaluierte Produkte

Nach ausführlichen Recherchen im Internet wurden zwei mögliche Preinstallation Environments gefunden. Das mit der Software "Bart PE Builder" erstellte "Bart PE" steht in unmittelbarer Konkurrenz zum Microsoft Windows Preinstallation Environment, kurz "Windows PE". Die beiden Produkte können nicht direkt beim Softwarehändler gekauft werden. Microsoft bietet eine Entwicklungsumgebung namens "OEM Preinstallation Kit", kurz "OPK", mit welcher sich ihr Preinstallation Environment erstellen lässt.

Nachfolgend werden die beiden evaluierten Produkte mit ihren jeweiligen Entwicklungsumgebungen aufgezeigt.

Tabelle 1: Übersicht PE Produkte.

	Produkt 1	Produkt 2
Hersteller	Bart	Microsoft
Produktname	Bart PE	Windows PE
Betriebssystem Quelle ¹	Windows Server 2003	Windows Server 2003
Entwicklungsumgebung	PE Builder	OEM Preinstallation Kit
Version	3.1.10.a	2005

Eine Muss-Analyse wurde in dieser Evaluation absichtlich nicht durchgeführt, da beide Produkte den Anforderungen des Projekts genügen.

4.3.2 Entscheidungskriterien

Mit Hilfe der nachfolgend aufgezählten Entscheidungskriterien soll es möglich sein, die Stärken und Schwächen der beiden Produkte abzuschätzen.

Tabelle 2: Kriterienkatalog der PE Evaluation.

Nr.	Kriterium	Beschreibung
1.	Startgeschwindigkeit	Wichtigstes Kriterium aus Sicht des Kunden stellt die Startzeit des jeweiligen Preinstallation Environments dar. Um den Vergleich transparent zu halten, wird die Zeitmessung auf demselben Testsystem, unter gleichen Voraussetzungen (d.h. gleicher Funktionsumfang, etc.) durchgeführt. Die Messung startet mit dem Start des Testsystems und endet mit dem geladenen, benutzungsfähigen PE.
2.	Funktionsumfang	Mindestens ebenso wichtig wie der Zeitbedarf eines Starts ist der Funktionsumfang, welcher das jeweilige Preinstallation Environment bietet. VisualBasic Script (VBS), Hypertext Application (HTA) und ActiveX Data Objects (ADO) und Windows Management Instrumentation (WMI) muss lauffähig sein, um die Anforderungen des Projekts wunschgemäss umsetzen zu können.
3.	Support	Die Firma Schindler Informatik erwartet von jedem Produkt, welches eingekauft wird, dass eine Supportmöglichkeit garantiert wird. Die Art und Weise des Supports ist in dieser Evaluation nicht ausschlaggebend.
4.	Lizenzkosten	Ein weiteres Entscheidungskriterium stellen die Kosten zur Erwerbung von Produktlizenzen dar.

4.3.3 Entscheidungsmatrix

Die Kriteriennummern referenzieren jeweils mit den im Kriterienkatalog erfassten Kriterien.

Tabelle 3: Analyse der PEs.

¹ Als Betriebssystemquelle zur Erstellung eines Preinstallation Environments wurde jeweils das Betriebssystem Windows Server 2003 SP1 angewendet.

Diese Wahl begründet sich einerseits in der höheren Aktualität des Windows Server Kernels gegenüber dem Windows XP Kernel und andererseits in der Gegebenheit, dass die Schindler Informatik nur Software Assurance (SA) Verträge im Serverbereich besitzt.

Nr.	Kriterium	Gewichtung ¹	Produkt 1		Produkt 2	
			Punkte ²	Total ³	Punkte	Total
1.	Startgeschwindigkeit	3	2	6	3	9
2.	Funktionsumfang	3	3	9	2	6
3.	Support	2	1	2	3	6
4.	Lizenzkosten	1	3	3	1	1
Total				20		22

4.3.4 Auswertung

Sowohl vor als auch nach der Evaluation bin ich der Meinung, dass beide Produkte genauso gut verwendet werden könnten. Die Entscheidungsmatrix der Kann-Analyse zeigt uns jedoch sehr deutlich, welches Produkt unsere Bedürfnisse am Besten befriedigen

Tabelle 4: Auswertung PE Evaluation.

Nr.	Kriterium	Auswertung
1.	Startgeschwindigkeit	<p>✗ Produkt 1: ? 3 Minuten. Der Ladevorgang von Bart PE verlängerte sich insbesondere darum, weil Bart PE standardmässig das WMI - Repository während der Laufzeit erstellt. WMI ist eine zwingende technische Anforderung für die Realisierung der Projektarbeit.</p> <p>✓ Produkt 2: ? 1.5 Minuten.</p>
2.	Funktionsumfang	<p>✓ Produkt 1: WSH, HTA, ADO und WMI werden als Plug-Ins angeboten. Es gibt diverse Communities im Internet, welche Plug-Ins für Bart PE erstellen und verwalten. Plug-Ins können in der "PE Builder"-Software sehr einfach aktiviert/deaktiviert werden.</p> <p>✗ Produkt 2: WSH, HTA, ADO und WMI werden in Form von sog. "Optional Components" angeboten. Der Integrationsvorgang gestaltet sich jedoch weniger komfortabel wie bei Bart PE. Die Verbreitung von zusätzlichen Komponenten ist nicht so gross wie bei Bart PE. Diese Tatsache hat wohl den Grund, dass Windows PE nur gegen teure Software Assurance Lizenzen erhältlich ist. Bart PE hingegen ist "frei" verfügbar.</p>
3.	Support	<p>✗ Produkt 1: Bart bietet keinen Herstellersupport auf seine Produkte [Quelle 2.01]. Es besteht jedoch die Möglichkeit, Bart PE bei der Firma CTS (Computer Technology Systems) gegen eine anwendungsabhängige Lizenzgebühr zu beziehen. Diese Firma ist dann zuständig für den Support des Produkts. [Quelle 2.02]</p> <p>✓ Produkt 2: Im Rahmen des Enduser License Agreements (EULA) bietet Microsoft Support auf das erworbene Produkt.</p>
4.	Lizenzkosten	<p>✓ Produkt 1: Für privaten wie auch für kommerziellen Gebrauch (zu Hause und innerhalb der eigenen Firma) darf Bart PE kostenlos und uneingeschränkt genutzt werden. Weiterverkauf ist untersagt.</p> <p>✗ Produkt 2: Windows PE ist derzeit nur über das kostenintensivste und umfangreichste Lizenzprogramm von Microsoft erhältlich: Software Assurance (SA). Lizenzen können sowohl für Desktop- als auch für Server-Betriebssysteme erworben werden. Siehe Anhang A.</p>

¹ Gewichtung: 1=Weniger wichtig, 2=Wichtig, 3=Sehr wichtig.

² Punkte: 1=Genügend erfüllt, 2=Gut erfüllt, 3=Sehr gut erfüllt.

³ Total = Gewichtung * Punkte.

4.4 Evaluation der RAMDisk-Produkte

4.4.1 Evaluierbare Produkte

Treiber für RAMDisks gibt es von diversen Herstellern. Da Windows PE einen sehr beschränkten Funktionsumfang mit sich bringt, ist es nicht ausgeschlossen, dass ein Windows XP-kompatibler Treiber seinen Betrieb unter Windows PE verweigert. Festzustellen ist eine Inkompatibilität unter Windows PE meistens nur durch einen Bluescreen.

Die evaluierten Produkte werden in nachfolgender Tabelle illustriert:

Tabelle 5: Übersicht RAMDisk-Produkte.

	Produkt 1	Produkt 2	Produkt 3
Produkt Name	Microsoft RAMDisk	Qsoft RAMDisk	Qsoft RAMDisk
Version	Windows 2000	5.0.2.2	5.2.10.2
Freeware	Ja.	Ja.	Nein, \$6.00USD / Lizenz.
Bemerkung	? Quellcode ist verfügbar. ? Original Windows Treiber.	? Quellcode ist verfügbar. ? Basiert auf Windows Treiber. ? Limitiert auf max. 64MB.	? Quellcode ist verfügbar. ? Basiert auf Windows Treiber.
Weitere Infos	? [Quelle 2.03]	? [Quelle 2.04]	? [Quelle 2.05]

4.4.2 Entscheidungskriterien

Mit Hilfe der nachfolgend aufgezählten Entscheidungskriterien soll es möglich sein, die Stärken und Schwächen der evaluierten RAMDisk-Produkte abzuschätzen.

Tabelle 6: Kriterienkatalog der RAMDisk Evaluation.

Nr.	Kriterium	Beschreibung
1.	Kompatibilität und Stabilität	Die RAMDisk muss kompatibel sein zu unserem Preinstallation Environment. Zusätzlich wird geprüft, ob die RAMDisk während dem Betrieb (lesen/schreiben) Stabilitätsprobleme aufweist.
2.	Variable Diskgrösse	Die Grösse der RAMDisk muss variabel sein. Das heisst jedoch nicht, dass es eine Unabdingbarkeit gibt, die RAMDisk-Grösse während des laufenden Betriebs zu ändern.
3.	Freeware	Zusätzliches K.O.-Kriterium stellen die Lizenzkosten des Produkts dar. Es wird erwartet, dass keine Extrakosten entstehen.

4.4.3 Entscheidungsmatrix

Die Kriteriennummern referenzieren jeweils mit den im Kriterienkatalog erfassten Kriterien.

Tabelle 7: Analyse der RAMDisk-Produkte.

		Produkt 1	Produkt 2	Produkt 3
Nr.	Kriterium	Erfüllung	Erfüllung	Erfüllung
1.	Kompatibilität und Stabilität	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2.	Variable Diskgrösse	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.	Freeware	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4.4.4 Auswertung

Anhand der Resultate der Evaluationsphase fiel die Entscheidung auf die kostenlose Version des Qsoft RAMDisk-Treibers. Während der Recherche viel mir auf, dass sich dieser Treiber bereits in der BartPE-Szene grosser Beliebtheit freut. Produkt 2 und 3 sind modifizierte Windows Treiber. Die Speicherplatzlimitierung des gewählten Produkts spielt für die beabsichtigte Anwendung keine Rolle.

Tabelle 8: Auswertung der RAMDisk Evaluation

Nr.	Kriterium	Auswertung
1.	Kompatibilität und Stabilität	<p>⚡ Produkt 1: Treiber wurde in Windows PE eingebunden. Bereits kurze Zeit nach dem Start von Windows PE stürzte das System ab. Ein klassischer "Bluescreen" weist auf die Treiberinkompatibilität hin. Lese-/Schreibtests konnten demzufolge nicht durchgeführt werden.</p> <p>⚡ Produkt 2: Lese-/Schreibtest erfolgreich bestanden.</p> <p>⚡ Produkt 3: Lese-/Schreibtest erfolgreich bestanden.</p>

2.	Variable Diskgrösse	✎ Produkt 1 / 2 / 3: Diskgrösse wird in der Registry in Form eines hexadezimalen Werts gespeichert und kann an dieser Stelle geändert werden.
3.	Freeware	✎ Produkt 1: Kostenlos erhältlich, sowohl für privaten als auch für kommerziellen Einsatz.
		✎ Produkt 2: Kostenlos erhältlich, sowohl für privaten als auch für kommerziellen Einsatz. Limitierung der Diskgrösse auf 64MB.
		✎ Produkt 3: Lizenzkosten von 6.00USD pro Lizenz.

4.5 Evaluation der Installationsmethode

4.5.1 Methode 1: Unattended Windows Installation

Die klassische Installationsmethode, welche bis heute bei Schindler praktiziert wird, ist die "Unattended Windows Installation". Diese "unbeaufsichtigte Installation" erfordert keine Supporter-Interaktion während dem Windows Installationsprozess. Es wird eine blanke Neuinstallation durchgeführt, welche völlig hardwareunabhängig ist.

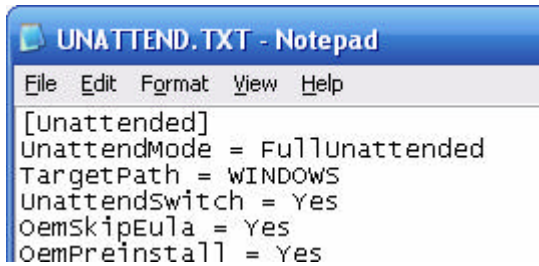


Abbildung 2: Auszug aus einer Windows Setupkonfigurationsdatei (Unattend.txt).

4.5.2 Methode 2: Imaging mit Ghost

Die Methode 2 sieht vor, Ghost Images zur Verteilung des Betriebssystems zu nutzen. Ein Image repräsentiert ein Datenträgerabbild. In unserem Fall würde ein Image die Systempartition C: eines Clients abbilden. Ghost wird bei der Schindler Informatik oftmals als Backup-Werkzeug für Testsysteme eingesetzt.



Abbildung 3: Imaging-Software "Ghost 9.0".

4.5.3 Methode 3: Imaging mit Ximage

Eine weitere interessante Methode bietet Ximage. Dieses Command-Line Tool wird im "Windows Automated Installation Kit", kurz "WAIK" mitgeliefert und ist zentraler Bestandteil des Windows Vista Setups. Von der Software Ximage liegt derzeit eine "CTP"-Version ("Community Technology Preview") vor. Mit Ximage lassen sich dateibasierte Images erstellen. Diese "WIM"-Images ("Microsoft Windows Imaging") sind, im Gegensatz zu sektorbasierten Images, hardwareunabhängig.

[\[Quelle 2.06\]](#)

[\[Quelle 2.07\]](#)

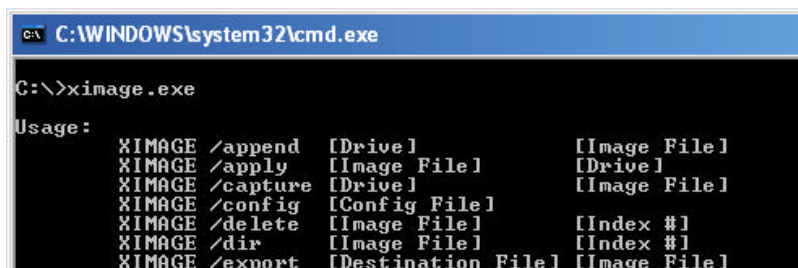


Abbildung 4: Command-Line Imagingsoftware "Ximage".

4.5.4 Entscheidungskriterien

Mit Hilfe der nachfolgend aufgezählten Entscheidungskriterien soll es möglich sein, die Stärken und Schwächen der evaluierten Installationsmethoden abzuschätzen.

Tabelle 9: Kriterienkatalog der Installationsmethoden.

Nr.	Kriterium	Beschreibung
1.	Vollautomatisierter Setupprozess	Der Setupprozess darf nur so wenig Benutzerinteraktion erfordern, wie nötig ist. Unnötige Benutzereingaben sollen vermieden werden.
2.	Integration in Softwareverteilung	Die bestehende Softwareverteilung "Unity" muss mit in die Installation einbezogen werden.
3.	Hardware Unabhängigkeit	Die Installation muss auf sämtlicher Hardware lauffähig sein, welche als Schindler Standard deklariert wurde.
4.	Niedriger Entwicklungsaufwand	Ein wichtiges Kriterium bei der Wahl der Installationsmethode ist selbstverständlich der Entwicklungsaufwand. Da dieses Projekt von einem zeitlichen und somit einem finanziellen Rahmen umspannt ist, gilt es diesem Kriterium ein grosses Augenmerk zu geben.
5.	Schnelle Installationsgeschwindigkeit	Dieses Kriterium ist vor allem während der Anwendung des Produkts von grosser Bedeutung: Lange Wartezeiten sind, sofern dies in eigenen Kompetenzen liegt, zu vermeiden.
6.	Niedriger Administrationsaufwand	Ein Punkt, welcher nicht vernachlässigt werden darf, stellt der Administrationsaufwand dar. Es stellen sich dabei folgende Fragen: Welcher Zeitaufwand muss im Fall einer Anpassung in Kauf genommen werden? Wie umfangreich gestaltet sich ein Aktualisierungsvorgang?

4.5.5 Entscheidungsmatrix

Die Kriteriennummern referenzieren jeweils mit den im Kriterienkatalog erfassten Kriterien.

4.5.5.1 Muss-Analyse

Tabelle 10: Muss-Analyse der Installationsmethoden

Nr.	Kriterium	Methode 1	Methode 2	Methode 3
		Erfüllung	Erfüllung	Erfüllung
1.	Vollautomatisierter Setupprozess	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2.	Integration in Softwareverteilung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3.	Hardware Unabhängigkeit	<input checked="" type="checkbox"/>	<input type="checkbox"/> ¹	<input checked="" type="checkbox"/>

4.5.5.2 Kann-Analyse

Tabelle 11: Kann-Analyse der Installationsmethoden

Nr.	Kriterium	Gewichtung ²	Methode 1		Methode 3	
			Punkte ³	Total ⁴	Punkte	Total
4.	Niedriger Entwicklungsaufwand	3	2	6	1	3
5.	Schnelle Installationsgeschwindigkeit	2	1	2	3	6
6.	Niedriger Administrationsaufwand	1	3	3	2	2
Total				11		11

4.5.6 Auswertung

Anhand der Kann-Analyse kann festgestellt werden, dass Methode 1 und 3 eine äquivalente Punktzahl erreichten. Da es sich bei der Wahl der Installationsmethode um ein sehr wichtiges Ausscheidungsverfahren handelt, war ich der Meinung, die Entscheidung nicht ohne Absprache des Fachausbildners zu treffen. Nach einer kurzen Diskussion hatten wir zusammen beschlossen, die bewährte Methode 1 anzuwenden. Grund für diese Entscheidung war mitunter die Tatsache, dass von der Software "Ximage" lediglich eine Beta-Version (sog. "Community Technology Preview") verfügbar ist. Des Weiteren würde uns die Managementsoftware fehlen, welche benötigt wird, um ein WIM-Image mit Service Packs und Hotfixes zu ergänzen. Von dieser Managementsoftware gibt es derweilen noch nicht einmal eine Beta-Version.

¹ Dieses Muss-Kriterium wurde nicht erfüllt und wird daher in der Kann-Analyse nicht mehr berücksichtigt.

² Gewichtung: 1=Weniger wichtig, 2=Wichtig, 3=Sehr wichtig.

³ Punkte: 1=Genügend erfüllt, 2=Gut erfüllt, 3=Sehr gut erfüllt.

⁴ Total = Gewichtung * Punkte.

Tabelle 12: Auswertung der InstallationsmethodenEvaluation.

Nr.	Kriterium	Auswertung
1.	Vollautomatisierter Setupprozess	✗ Methode 1: Windows Unattended Setup bietet eine vollautomatisierte Installation des Betriebssystems.
		✗ Methode 2: "Ghost" führt zwar kein Windows Setup aus, kann jedoch ebenfalls vollautomatisiert ein Betriebssystem auf die Harddisk eines Computers legen.
		✗ Methode 3: Nach einem ähnlichen Prinzip wie Ghost Imaging funktioniert Ximage. Aufgezeichnete WIM-Images können 1:1 auf ein Zielsystem entpackt werden.
2.	Integration in Softwareverteilung	✗ Methode 1: Die Clientsoftware "Unity Closed Client" der Softwareverteilung "Unity" ist bereits in der Betriebssystemquelle des bestehenden Schindler Setups integriert.
		✗ Methode 2: Die Clientsoftware "Unity Closed Client" kann vor dem Aufzeichnen des Ghost-Images installiert werden.
		✗ Methode 3: Die Clientsoftware "Unity Closed Client" kann vor dem Aufzeichnen des WIM-Images installiert werden.
3.	Hardware Unabhängigkeit	✗ Methode 1: Windows Setup installiert Gerätetreiber automatisch, sofern für das entsprechende Gerät passende Treiber vorhanden sind.
		✗ Methode 2: Sektorbasierte Images, wie sie mit Ghost erstellt werden können, sind hardwareabhängig. Microsoft stellt Tools zur Verfügung, welche vor dem Aufzeichnen eines Images benutzer- und gerätespezifische Daten vom Mastercomputer entfernen können. Trotzdem muss für jeden HAL (Hardware Abstraction Layer) ein eigenes Image erstellt und unterhalten werden. Weiterer Nachteil von Methode 2: Die Partitionsgrösse der Systempartition ist abhängig vom Image und kann nicht nach belieben variiert werden. <i>Der Kunde hat bereits einige Erfahrungen mit Imaging gemacht und wünscht sich eine alternative Lösung.</i>
		✗ Methode 3: Ximage erstellt dateibasierte Images, welche völlig hardwareunabhängig sind. Die Partitionsgrösse der Zielpartition kann frei gewählt werden.
4.	Niedriger Entwicklungsaufwand	✗ Methode 1: Ein Windows Unattended Setup wird bei der Schindler Informatik bereits eingesetzt. Die Parameter für das Setup der unbeaufsichtigte Windows Installation sind trivial.
		✗ Methode 3: Ximage würde garantiert mehr Entwicklungsaufwand in Anspruch nehmen. Der Grund dafür findet sich im Konzept von Ximage: Betriebssystem Images müssen auf einem Master-Client Computer „aufgezeichnet“ (engl. „captured“) werden. Schliesslich kommt dazu, dass wir mit Ximage schlicht keine Erfahrung haben, was einen negativen Einfluss auf den Entwicklungsaufwand haben kann.
5.	Schnelle Installationsgeschwindigkeit	✗ Methode 1: Eine Windows Unattended Installation dauert 40-50 Minuten. ✗ Methode 3: In rund 15 Minuten war ein Windows XP SP2 Image auf dem Testsystem installiert.
6.	Niedriger Administrationsaufwand	✗ Methode 1: Eine tagaktuelle Windows XP Quelle liegt bereits auf dem Softwareverteilungsserver. Updates wie Service Packs oder Hotfixes können ohne grossen Aufwand eingespielt (engl. „slipstreamed“) werden.
		✗ Methode 3: Um ein WIM-Image zu aktualisieren muss dieses zuerst mit Ximage gemountet werden. Eine Managementsoftware ist bei Microsoft in Planung, wird jedoch erst mit Windows Vista offiziell erhältlich sein.

5 Realisierung

5.1 Entwicklung des Preinstallation Environments

5.1.1 Entwicklungsumgebung vorbereiten

5.1.1.1 Verzeichnisstruktur

Die nachvollziehbare Dokumentation des Entwicklungsprozesses von Windows PE bedingt eine verbindliche Strukturierung der Verzeichnisse auf dem Entwicklungssystem. Um dies zu gewährleisten wird in dieser Dokumentation stets die in der nachfolgenden Tabelle illustrierte Verzeichnisstruktur angewendet.

Einige Verzeichnisse (z.B. „Components“) dienen lediglich der Erhöhung der Flexibilität während des Entwicklungsvorgangs. Kommandos zur Erweiterung von Windows PE werden in dieser Dokumentation meistens direkt auf das Zielverzeichnis von Windows PE gerichtet. Die einzelnen Komponenten beispielsweise können jedoch in die dafür vorgesehenen Unterverzeichnisse abgelegt werden, bevor sie ins Zielverzeichnis von Windows PE gelegt werden.

Beim Reproduzieren des Preinstallation Environments empfehle ich, sämtliche relevante Kommandos in einzelne Batch-Dateien abzulegen.

Tabelle 13: Verzeichnisstruktur auf dem Entwicklungssystem.

Verzeichnisstruktur	Beschreibung des Verzeichnissesinhalts
C:	Laufwerk C:\ des Entwicklungssystems.
Windows PE	Windows PE Projektverzeichnis.
Components	Komponenten zur Erweiterung von Windows PE.
htaFactory	Factory Components zur Initialisierung von Windows PE.
htaLogin	Hypertext-Anwendung „Login Form“.
htaSetup	Hypertext-Anwendung „Setup Assistant“.
RAMDisk	Qsoft RAMDisk Treiber (Version 5.0.2.2).
Shutdown	Software zum Herunterfahren/Neustarten von Windows PE.
Scripting Components	Komponenten für WSH, HTA und ADO.
Drivers	Zusätzliche Treiber, welche in Windows PE integriert werden.
BCOM	Treiber für Broadcom NetXtreme Netzwerkkadpter.
Sources	Betriebssystem Quellen.
Win2k3_eng	Windows Server 2003 SP1, Englisch.
WinXP_eng	Windows XP Professional SP2, Englisch.
Tools	Tools, Scripts und Konfigurationen von der OPK CD-Rom.
WinPE_CD	Erstellungsort für Windows PE.
WinPE_ISO	Erstellungsort für Windows PE ISO Images.

5.1.1.2 OPK Tools einrichten

Um ein Windows PE erstellen zu können, werden die Software-Tools, Scripts und Konfigurationsdateien des OEM Preinstallation Kits (OPK) benötigt.

- ? Der Inhalt des OPK-Verzeichnisses "G:\Tools\X86" wird in das Verzeichnis "C:\WindowsPE\Tools" kopiert.
- ? Der Inhalt des OPK-Verzeichnisses "G:\WinPE" wird in das Verzeichnis "C:\WindowsPE\Tools" kopiert.

Der Laufwerksbuchstabe "G:" entspricht hier dem CD-Rom. Weitere Informationen über das OPK sind im [Glossar](#) zu finden.

5.1.1.3 Betriebssystem Quelle vorbereiten

- ? In das Verzeichnis "C:\WindowsPE\Sources" werden, in ein entsprechendes Unterverzeichnis, die zur Herstellung von Windows PE benötigte Betriebssystem Quelle kopiert.

Wichtig:

Es muss sichergestellt werden, dass der gesamte Inhalt der Windows CD in das „Sources“-Verzeichnis kopiert wird! Sollten irgendwelche Ordner - irrelevant wie unwichtig sie sind - fehlen, kann dies dazu führen, dass die zur Herstellung von Windows PE bereitliegenden OPK-Scripts ihren Dienst verweigern.

Siehe Problemprotokoll [\[Problem 2\]](#).

- ? In dieser Projektarbeit wird als Windows PE – Betriebssystem Quelle stets Windows Server 2003 SP1 (englisch) verwendet. Eine Begründung über die Wahl wurde bereits während der Evaluierung von Windows PE abgegeben.

5.1.2 Windows PE Basis-Image erstellen

Windows PE ist sehr flexibel und modular. In diesem Dokumentationsabschnitt wird ausschliesslich der grundlegende Erstellungsvorgang von Windows PE dokumentiert. Die Erweiterung durch eine beliebige Kombination von Komponenten ist selbstverständlich möglich und wird in nachfolgenden Abschnitten ausführlich dokumentiert.

- ? Zum Erstellen von Windows PE liefert Microsoft im OPK ein Script namens "mkimg.cmd", welches im OPK-Verzeichnis "Tools" zu finden ist. In der OPK-Hilfe (bzw. durch Eingabe des Command-Line Parameter "/?") können die Parameter ausfindig gemacht werden, welche dem Script übergeben werden können.
- ? Damit das Script "mkimg.cmd" korrekt ausgeführt wird, muss der Fokus der Eingabeaufforderung mit dem DOS-Befehl "cd" (gleichbedeutend mit "change directory") auf das OPK-Verzeichnis gelegt werden.

```
cd C:\WindowsPE\Tools
```

- ? Mit nachfolgendem Befehl wird ein Windows PE aus der Quelle "C:\WindowsPE\Sources\win2k3_eng" ins Zielverzeichnis "C:\WindowsPE\WinPE_CD" erstellt. Die Optionen "Windows Management Instrumentation (WMI)" und "Plug 'n' Play Device Driver Detection (PnP)" werden aktiviert. WMI wird zwingend benötigt um Hardwareinformationen auszulesen. PnP dient zur automatischen Geräteerkennung während des Betriebs vom Windows PE. Die Windows Firewall (WF) wird deaktiviert, da Windows PE kein Sicherheitsrisiko darstellt.

```
mkimg.cmd C:\WindowsPE\Sources\win2k3_eng C:\WindowsPE\WinPE_CD /WMI /PNP /NOWF
```

- ? Das Script läuft nun automatisch und wird mit einer kurzen Bestätigung beendet.

```
1250 files were copied from 1250 files.  
Successfully created WinPE directory in "C:\WindowsPE\WinPE_CD\  
C:\WindowsPE\Tools>
```

Das soeben erstellte Windows PE kann für unsere Anwendung nicht ohne weitere Modifikationen verwendet werden. Es bedarf einer massiven Erweiterung der Funktionalität.

5.1.3 RAMDisk integrieren

5.1.3.1 Problemstellung

Das Konzept von Windows PE erlaubt den Start dieses, im Speicherplatz massiv beschränkten Betriebssystems von einem Medium wie CD oder DVD, neuerdings auch von USB-Memorysticks und Remote-Installation Server (RIS). Da Windows PE ursprünglich nur für ROM-Medien ("Read-Only Memory", nur lesbares Medium) entwickelt wurde, können Programme während der Ausführung keine Daten schreiben. Dieser Nachteil zwang mich, eine RAMDisk in Windows PE einzubinden.

Quasi als positiver Nebeneffekt erreichte ich mit der Einführung einer RAMDisk, dass meine scriptbasierten Anwendungen spürbar schnellere Reaktionszeiten aufwiesen.

5.1.3.2 Integrationsvorgang

Die nachfolgenden Schritte zeigen auf, wie die QSoft RAMDisk-Treiber in ein bestehendes Windows PE integriert werden. Weitere Informationen sind zu finden unter: [\[Quelle 2.08\]](#)

- ? Zuerst muss der RAMDisk - Treiber heruntergeladen werden. Siehe Quelle
- ? Die komprimierte Archivdatei wird nach "C:\WindowsPE\Components\RAMDisk" extrahiert.
- ? Die extrahierte Treiberdatei "C:\WindowsPE\Components\RamDisk\ramdisk.sys" wird in "ramdrv.sys" umbenannt. Alle anderen entpackten Dateien des Treiberpakets werden für unsere Anwendung nicht benötigt und können gelöscht werden.
- ? Die umbenannte RAMDisk-Treiberdatei "ramdrv.sys" wird nun in das Windows PE Verzeichnis "C:\WindowsPE\WinPE_CD\i386\system32\drivers" kopiert.
- ? Nach dem Kopiervorgang wird unter "C:\WindowsPE\Tools" eine Konfigurationsdatei namens "ramdisk.inf" erstellt. Diese wird mit Registry-Informationen ergänzt. Die Liste der einzutragenden Informationen wird in der Tabelle des Anhangs C illustriert.
- ? Die Grösse unserer RAMDisk wurde auf 32MB festgelegt. Im [Anhang C](#) wird eine Tabelle abgebildet, welche weitere hexadezimale Werte für die Diskgrösse der RAMDisk enthält.
- ? Damit die in der RAMDisk-Konfigurationsdatei eingetragenen Registry-Einträge beim Neuerstellen von Windows PE wunschgemäss mit einbezogen werden, muss in der zentralen Konfigurationsdatei "C:\WindowsPE\Tools\Config.inf" ein referenzierender Eintrag gemacht werden. Dieser Eintrag

dient als Verknüpfung zur Datei "ramdisk.inf". Alle anderen Einträge der Sektion "[Add Registry Existing]" bleiben unverändert.

Tabelle 14: Konfigurationsdatei "ramdiskinf".

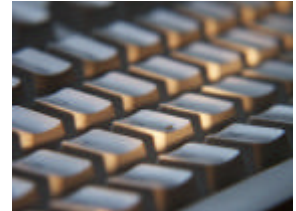
```
[Add Registry Existing]
setupreg.hiv = .\ramdisk.inf,AddReg
```

- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.1.4 Tastaturlayout anpassen

5.1.4.1 Problemstellung

Windows PE installiert standardmässig das US-amerikanische Tastaturlayout. Sofern in einer Windows PE - Anwendung keine Benutzerinteraktion erfordert würde, müsste das Tastaturlayout nicht geändert werden. In diesem Projekt ist es also unumgänglich, dass das Tastaturlayout an die regionalen Einstellungen anzupassen. Es ist unter Windows PE möglich, wenn auch nicht ganz trivial, ein alternatives Tastaturlayout zu aktivieren.



5.1.4.2 Sprachcode Referenz

Nachfolgende Tabelle illustriert eine Auswahl von Sprachen mit den dazugehörigen Codes. [\[Quelle 2.09\]](#).

Tabelle 15: Sprachcode Referenztable.

Sprache	Code
Czech	00000405
Danish	00000406
Dutch (Standard)	00000413
English (United States)	00000409
English (United Kingdom)	00000809
Finnish	0000040B
French (Standard)	0000040C
French (Belgian)	0000080C
French (Swiss)	0000100C
French (Canadian)	00000C0C
German (Standard)	00000407
German (Swiss)	00000807
Hungarian	0000040E
Icelandic	0000040F
Italian (Standard)	00000410
Norwegian (Bokmal)	00000414
Polish	00000415
Portuguese (Portugal)	00000816
Portuguese (Brazil)	00000416
Russian	00000419
Slovak	0000041B
Slovenian	00000424
Spanish (Mexican)	0000080A
Spanish (Traditional Sort)	0000040A
Swedish	0000041D

5.1.4.3 Integrationsvorgang

In den nächsten Schritten wird gezeigt, welche Änderungen an der bestehenden PE Kompilation vorgenommen werden müssen, um das Tastaturlayout "German (Swiss)" zu aktivieren.

- ? Wie sich nach einigen Recherchen herausstellte, handelt es sich zwei fehlende Registry Keys, mit welchen ein alternatives Tastaturlayout konfiguriert werden kann. Um neue Registry Einträge einzubinden, erstellen wir eine neue Konfigurationsdatei namens "keyboard.inf" im Verzeichnis "C:\WindowsPE\Tools".

Nachfolgende Tabelle zeigt den Inhalt dieser Konfigurationsdatei "keyboard.inf", dabei entspricht der markierte Registry Wert dem in der letzteren Tabelle definierten Sprachcode für "German (Swiss)".

Tabelle 16: Konfigurationsdatei "keyboard.inf".

```
[Version]
Signature="$WINDOWS NT$"

[AddReg]
HKCU,"Keyboard Layout\Preload",,0x00000012
HKCU,"Keyboard Layout\PreLoad","1",0x00020000,"00000807"
```

- ? Damit die neuen Registry Einträge der Konfigurationsdatei "keyboard.inf" einstweilen auch in einem Windows PE vorzufinden sind, muss in der INI-Sektion "[Add Registry Existing]" der zentralen Windows PE Konfigurationsdatei ("C:\WindowsPE\Tools\Config.inf") ein neuer Eintrag erstellt werden. Dieser Eintrag dient als Verknüpfung zur Datei "keyboard.inf". Alle anderen Einträge der Sektion "[Add Registry Existing]" bleiben unverändert bestehen.

Tabelle 17: Änderung an der Konfigurationsdatei "Config.inf".

```
[Add Registry Existing]
default = .\keyboard.inf,AddReg
```

- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.1.5 Shutdown-Tool integrieren

5.1.5.1 Problemstellung

Windows PE kann nicht auf dieselbe Art und Weise heruntergefahren werden, wie man sich dies in herkömmlichen Windows Betriebssystemen gewohnt ist. Microsoft stellt zu meinem Verblüffen keine Tools zur Verfügung, mit denen sich Windows PE herunterfahren liesse. Einen Neustart kann selbst nur dann ausgelöst werden, wenn alle Anwendungen (inkl. Eingabeaufforderung) geschlossen werden.

Die in Microsoft Betriebssystemen normalerweise zur Verfügung gestellte Anwendung "shutdown.exe" ist in Windows PE nicht zu finden. Selbst wenn dieses Microsoft Shutdown-Tool von einem bestehenden Windows Betriebssystem kopiert und in Windows PE eingefügt wird, kann das System nicht erfolgreich heruntergefahren werden.

5.1.5.2 Lösung

Nach Recherchen in Internet entdeckte ich einige semiprofessionelle Shutdown-Tools. Darunter die kostenlose Software "ShutdownPE".[\[Quelle 2.12\]](#)

```
Version 1.0, October 2005, David Clarke
Shutdown or Restart Windows PE only

PEDOWN SHUTDOWN : RESTART : POWEROFF

See http://mystuff.clarke.co.nz

This software is provided "as-is", without any express or
implied warranty. In no event will the author be held liable
for any damages arising from the use of this software.
```

Abbildung 5: Shutdown-Tool für Windows PE: "ShutdownPE.exe".

5.1.5.3 Integrationsvorgang

- ? Die Software besteht lediglich aus einer ausführbaren EXE-Datei. Diese Datei ("ShutdownPE.exe") wird ins Windows PE Erstellungsverzeichnis, nach "C:\WindowsPE\WinPE_CD\i386\system32" kopiert.
- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.1.6 Scripting Components

5.1.6.1 Beschreibung

Im OPK befindet sich ein Script namens "BuildOptionalComponents.vbs" zur Integration von ADO, HTA und WSH in Windows PE. Microsoft nennt diese „Optional Components“, kurz „OC“. Da in dieser Dokumentation noch von einige anderen Komponenten die Rede ist, nahm ich mir das Recht, der sehr allgemeingültige Name „Optional Components“ in „Scripting Components“ umzutaufen.

Wie bereits erwähnt, bieten die Scripting Components wahlweise Unterstützung für ActiveX Data Objects (ADO), Hypertext Applications (HTA) und Windows Scripting Host (WSH). In den allermeisten Anwendungsfällen von Windows PE ist es unumgänglich, die Scripting Components einzubinden. Es ist sehr wichtig, dass die Scripting Components jeweils aus derselben Windows Quelle erstellt werden, wie bereits Windows PE erstellt wurde. In unserem Fall muss also zwingend Windows Server 2003 SP1 ("C:\WindowsPE\Sources\win2k3_eng")

5.1.6.2 Integrationsvorgang

- ? Damit das Script "BuildOptionalComponents.vbs" korrekt ausgeführt wird, navigieren wir mit dem DOS-Befehl "cd" (gleichbedeutend mit "change directory") zum Quellverzeichnis von "BuildOptionalComponents.vbs".
`cd C:\WindowsPE\Tools`
- ? Mit folgendem Befehl werden sämtliche verfügbaren Technologien (ADO, HTA und WSH) aus der Windows Quelle extrahiert und in das Zielverzeichnis von Windows PE kopiert. Der Parameter "/Q" dient zur Unterdrückung von Informationsaufrufen.
`buildoptionalcomponents.vbs /S:C:\WindowsPE\Sources\win2k3_eng /D:C:\WindowsPE\WinPE_CD_W2K3 /ADO /HTA /WSH /Q`
- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.1.7 Netzwerktreiber integrieren

5.1.7.1 Problemstellung

Plug'n'Play-kompatible Hardware wird von Windows PE erkannt und installiert. Dies bedingt jedoch, dass für die entsprechende Hardware passende Windowstreiber vorhanden sind. Liegt für ein Gerät kein Treiber vor, so kann dieses Gerät nicht verwendet werden.

Nach dem ersten kompletten Testdurchlauf auf einem realen Testsystem wurde ein Fehler erkannt, welcher bei den bisherigen Tests auf den virtuellen Testsystemen nicht erkannt werden konnten: Windows PE konnte den Treiber für die Netzwerkkarte des Testsystems nicht finden.

Das Nichtvorhandensein von einigen Treiberarten spielt in unserem Anwendungsfall keine entscheidende Rolle (z.B. Treiber für multimediale Geräte, Grafikbeschleuniger und Vergleichbare). Auf den betriebswichtigen Netzwerkkartentreiber kann in unserem Preinstallation Environment nicht verzichtet werden. Ein passender Treiber muss integriert (oder wie es in der Fachsprache so schön heisst „injiziert“) werden.

5.1.7.2 Voraussetzungen/ Empfehlungen

- ? Treiber können nur in Verzeichnis eines bereits existierenden Windows PE integriert werden.
- ? Das verwendete Windows PE muss mit der Mking-Option "/PnP" erstellt worden sein, damit die neu eingebunden Treiber erkannt werden. Siehe [5.1.2 Windows PE Basis-Image erstellen](#).
- ? Die benötigten Windows Treiber können von der Hersteller-Webseite heruntergeladen werden. Alternativ besteht auch die Möglichkeit, Treiber von einer OEM CD-Rom zu beziehen. Allerdings besteht hier das Risiko, dass die Treiber veraltet sind und daher Kompatibilitätsprobleme verursachen könnten.
- ? Es gilt zu beachten, dass die Treiber für das zur Herstellung von Windows PE verwendeten Betriebssystem heruntergeladen werden. In unserem Fall werden folglich die Treiber für Windows 2003 Server heruntergeladen. (In der Tat sind diese meistens äquivalent mit Windows 2000/XP Treiber).

5.1.7.3 Integrationsvorgang

- ? Das Zielgerät benötigt Broadcom NIC-Treiber. Diese können entweder auf der Herstellerseite des Geräts oder auf der Herstellerseite des Netzwerkkartenherstellers heruntergeladen werden. Siehe [\[Quelle 2.16\]](#).
- ? Nach dem Herunterladen wird das Treiberpaket nach "C:\WindowsPE\Drivers\BCOM" entpackt.
- ? Für die Integration von Treibern bietet Microsoft im OPK eine Applikation namens "Drvinst.exe" („Driver Injection Tool“), welches wir im OPK-Verzeichnis "Tools" finden. Damit die Applikation "Drvinst.exe" korrekt ausgeführt wird, navigieren wir in der Eingabeaufforderung zum "Tools"-Verzeichnis.
`cd C:\WindowsPE\Tools`
- ? Anschliessend wird "Drvinst.exe" mit den erforderlichen Parametern gestartet: Mit dem Parameter "/inf" kann ein Verzeichnis angegeben werden, welches die nötigen Treiberdateien enthält. Der zweite Parameter definiert das Windows PE Zielverzeichnis, in welches die Treiber integriert werden sollen.
`drvinst.exe /inf:C:\WindowsPE\Drivers\BCOM C:\WindowsPE\WinPE_CD`

- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

```
Microsoft (R) Driver Injection Tool for Windows Preinstallation Environment
(WinPE) 1.5 Version 0.99. Copyright (C) Microsoft Corporation 2003. All rights
reserved.

Warning: The file C:\WindowsPE\WinPE_CD\i386\system32\NicCo32.dll in the WinPE i
mage is being overwritten.
Warning: The file C:\WindowsPE\WinPE_CD\i386\system32\NicIn32.dll in the WinPE i
mage is being overwritten.
Warning: The file C:\WindowsPE\WinPE_CD\i386\system32\Prounstl.exe in the WinPE
image is being overwritten.
Warning: The file C:\WindowsPE\WinPE_CD\i386\system32\DRIVERS\e1000325.sys in th
e WinPE image is being overwritten.

File in WinPE: C:\WindowsPE\WinPE_CD\i386\system32\DRIVERS\e1000325.sys Ver:6.3.
6.31, Date:29.06.2005
File to be injected: e1000325.sys Ver:8.6.11.0, Date:24.03.2003
Overwrite C:\WindowsPE\WinPE_CD\i386\system32\DRIVERS\e1000325.sys? (Answer "yes
" or "no", default "yes") yes

4 warning(s), 0 error(s)
2 INF file(s) were added to the WinPE image. 13 driver file(s) were copied into
the image.

C:\WindowsPE\Tools>
```

Abbildung 6: Driver Injection Tool zur Treiberintegration.

5.1.8 ISO Image erstellen

Zum Erstellen eines startbaren ISO Images aus den Dateien von Windows PE kann die von Microsoft bereitgestellte Anwendung "oscdimg.exe" verwendet werden.

- ? Mit dem folgenden Befehl werden die im Windows PE Verzeichnis abgelegten Daten in ein startbares ISO Image verpackt.

```
C:\WindowsPE\Tools\oscdimg.exe C:\WindowsPE\WinPE_CD
C:\WindowsPE\WinPE_ISO\WinPE.iso -h -n -bC:\WindowsPE\Tools\etfsboot.com
```

- ? Das Script läuft nun automatisch und wird mit einer kurzen Bestätigung beendet.

5.2 Realisierung der Hypertext-Anwendungen

In den nachfolgenden Abschnitten werden die entwickelten Hypertext-Anwendungen (HTA) dokumentiert. Sämtliche grafische Benutzeroberflächen wurden nach dem Schindler Standard entwickelt.

Ursprünglich versuchte ich das Schindler CI Design selber nachzubauen, da ich mir keinen vollen Zugriff zu den Style-Informationen verschaffen konnte. Schliesslich bediente ich mich jedoch dem Stylesheet (CSS) der Schindler Website. Das Darstellungsschema aus dem Schindler Stylesheet konnte ohne wesentliche Änderungen übernommen werden. Es musste lediglich die Ausrichtung von Tabellentext in einigen Style-Klassen korrigiert werden.

Auf den Webseiten von Microsoft fand ich die beiden Hilfsprogramme „Helpomatic“ und „Scriptomatic“. Beide Anwendungen leisteten mir während der Entwicklungsarbeit gute Dienste. Weitere Informationen siehe [\[Quelle 2.23\]](#) und [\[Quelle 2.24\]](#)

5.2.1 Factory Components

5.2.1.1 Beschreibung

In diesem Projekt werden sämtliche Komponenten als "Factory Components" bezeichnet, welche zur Initialisierung von Windows PE beitragen. Im Wesentlichen bestehen diese „Factory Components“ aus zwei Scriptroutinen: Batch-Script "Startnet.cmd" und Hypertext-Anwendung „Factory“.

Die Initialisierungsphase von Windows PE teilen wir somit in zwei Phasen:

	Phase 1	Die Batch-Datei "Startnet.cmd" wird automatisch beim Start von Windows PE gestartet. In dieser ersten Phase laufen einige Registrierung von Dynamic Link Libraries (DLL) für ADO, HTA und WSH. Diese Phase läuft im Batch-Script "Startnet.cmd" ab. Hier steht keine grafische Oberfläche zur Verfügung, da die dafür benötigten Komponenten gerade in diesem Moment installiert werden. Phase 1 initialisiert Phase 2.
	Phase 2	In Phase 2 wird eine grafische Oberfläche namens „Factory“ gestartet. Diese hat hauptsächlich zwei Aufgaben: Erstens muss die Windows PE Initialisierung mit der Software "Factory.exe" durchgeführt werden - und - zweitens müssen die Anwendungsdaten der Anwendung „Login Form“ vom Startmedium (CD/DVD) in die RAMDisk kopiert werden.

5.2.1.2 Entwicklungsvorgehen für Phase 1

- ? Bei der Installation der so genannten „Optional Components“ (siehe Scripting Components) wird im System-Verzeichnis von Windows PE eine Batch-Datei namens "OC2.bat" angelegt, welche bereits alle nötigen Registrierungsinformationen enthält.
- ? Die Informationen aus "OC2.bat" werden 1:1 in die Batch-Datei "Startnet.cmd" kopiert. Die Datei "OC2.bat" wird anschliessend nicht mehr benötigt und kann gelöscht werden.
- ? Alle bestehenden Einträge in "Startnet.cmd" werden bei diesem Vorgang überschrieben!
- ? Zusätzlich werden nach den Registrierungseinträgen folgende Zeilen in "Startnet.cmd" eingefügt.

Tabelle 18: Phase 1: Initialisierungseinträge zum Start von Phase 2

```
mshta.exe %SystemRoot%\System32\htaFactory\Factory.hta
Z:
cd Z:\htaLogin
start Z:\htaLogin\Login.hta
```

- ? Der Quellcode von "Startnet.cmd" ist zu finden unter [Anhang B](#).

5.2.1.3 Entwicklungsvorgehen für Phase 2

- ? Neue HTA erstellen. HTA-Grundgerüst mit Hilfe von Helpomatic erzeugen. Siehe [\[Quelle 2.23\]](#).
- ? Aufbau der HTA analog zur Schindler Website. Grundgerüst der Tabellen kann kopiert übernommen werden. Unnötige Informationen werden herausgelöscht.
- ? Schindler Stylesheet einbinden.
- ? Da diese Anwendung direkt vom Startmedium ausgeführt wird, schlagen die Funktionen zum Lesen von INI-Dateien fehl. Demzufolge müssen sämtliche Textinformationen statisch (d.h. nicht konfigurierbar) programmiert werden.

- ? Wie bereits erwähnt muss die Anwendung zwei Aufgaben übernehmen: Ausführen der Anwendung "Factory.exe" mit Parameter "-winpe" zur Initialisierung von Windows PE sowie den Kopiervorgang der Authentifizierungsanwendung vom Quellverzeichnis des Startmediums zum Zielverzeichnis der RAMDisk ("Z:\htaLogin").
- ? Der Quellcode von „Factory.hta“ ist zu finden unter [Anhang B](#).

5.2.1.4 Integrationsvorgang

- ? Scripting Components entsprechend den Anweisungen dieser Dokumentation in Windows PE integrieren. Siehe [Scripting Components](#).
- ? Die Batch-Datei "Startnet.cmd" wird in das Windows PE Verzeichnis nach „C:\WindowsPE\WinPE_CD\i386\system32\htaFactory“ kopiert. (Die bereits existierende "Startnet.cmd"-Datei wird überschrieben).
- ? Die komplette Anwendungsumgebung der Hypertext-Anwendung „Factory“ wird in das Windows PE Verzeichnis nach „C:\WindowsPE\WinPE_CD\i386\system32\htaFactory“ kopiert.
- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.2.2 Login Form

Wie bereits in der Aufgabenstellung ersichtlich ist, wurde bereits eine Authentifizierungsanwendung im Rahmen eines vorhergehenden Projekts entwickelt. In diesem Abschnitt werden einige Eigenschaften dieser Anwendung (genannt „Login Form“) kurz aufgezeigt.

5.2.2.1 Eigenschaften

Der Aufbau dieser Hypertext-Anwendung wurde ganzheitlich mit HTML realisiert. Die Anwendung umfasst rund 600 Zeilen Code, wovon VBScript mit rund 450 Zeilen den umfassendsten Teil ausmacht. Als Gestaltungsvorlage für die Benutzeroberfläche diente einstweilen das Anmeldefenster von Windows XP.

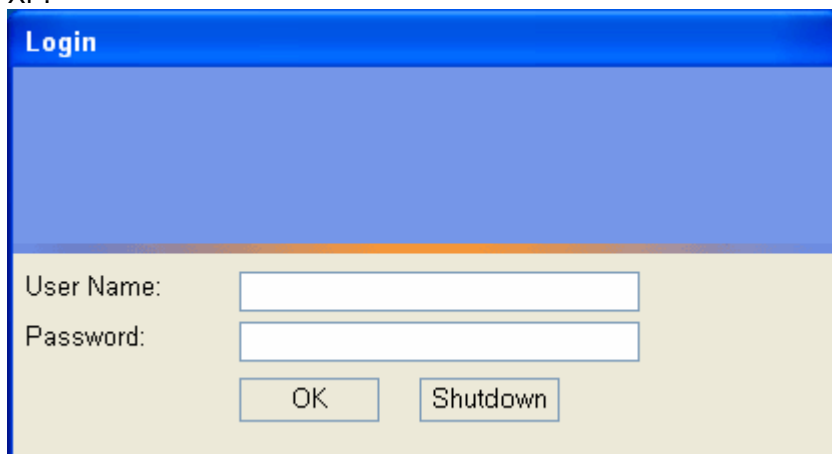


Abbildung 7: Ursprüngliche Benutzeroberfläche der Login-Anwendung.

Grundsätzlich authentifiziert die Anwendung einen Benutzer mittels der Software "Net.exe" gegen eine Netzwerkfreigabe. Diese DOS-Software wurde zur Anbindung von Netzlaufwerken bzw. zur Authentifizierung an dergleichen konzipiert. Diese Methode der Authentifizierung schien mir sehr unsicher zu sein. Doch nach der Protokollierung des Netzwerksverkehrs während einer Authentifizierung durch einen Netzwerksniffer konnte ich erkennen, dass die Authentifizierung über das sichere Authentifizierungsprotokoll „Kerberos“ ablief.

Die zweite verfügbare Methode authentifiziert Benutzername und Passwort gegen den Verzeichnisdienst „Active Directory“. Die Wahl der Authentifizierungsmethode steht dem Kunden frei. Zusätzlich kann der Kunde weitere Optionen wie Titel und Hintergrundfarbe selbstständig festlegen.

5.2.2.2 Modifikationen

Am Grundgerüst der Anwendung wurden lediglich unwesentlichen Änderungen vorgenommen. Der Kunde hat die Möglichkeit, Schindler Domänen mit ihren jeweiligen primären Unity Server (Netzwerkfreigabe) in der Konfigurationsdatei zu erfassen. Ein Benutzer wird dann gegen den in der Combobox „Unity Master“ gewählten, domänenspezifischen Unity Server authentifiziert. Die Benutzeroberfläche wurde vollständig dem neuen Schindler Standard angepasst. Sämtliche Grafiken mussten überarbeitet werden.

An einigen Stellen wurden Bereinigungen am bestehenden Code vorgenommen, um die Leserlichkeit zu Erhöhen.



Abbildung 8: Überarbeitete Benutzeroberfläche der LoginAnwendung.

Die Anwendung bekam zudem die Aufgabe, nach einer erfolgreichen Authentifizierung die Daten der Anwendung „Setup Assistant“ vom gewählten Unity Server in die lokale RAMDisk zu kopieren und anschliessen zu starten. Der genaue Ablauf wird im Diagramm des [Anhangs E](#) illustriert. Der Quellcode von „Login.hta“ ist zu finden unter [Anhang B](#).

5.2.2.3 Integrationsvorgang

- ? Die komplette Anwendungsumgebung wird in das Windows PE Verzeichnis nach „C:\WindowsPE\WinPE_CD\i386\system32\htalogin“ kopiert.
- ? Gegebenenfall können nun noch Änderungen an der Konfiguration der Login Form Anwendung vorgenommen werden.
- ? Zum Applizieren der Änderungen, welche durch diesen Integrationsvorgang vorgenommen wurden, muss das Windows PE Image neu erstellt werden. Siehe [5.1.8 ISO Image erstellen](#).

5.2.3 Setup Assistant

Die Hauptanwendung dieses Projekts wurde „Setup Assistant“ getauft. Dieser Assistent steuert den gesamten Installation- und Konfigurationsprozess. Er nimmt vom Benutzer Informationen entgegen, wertet diese aus und führt die gewünschten Aktionen aus.

Die Entwicklung dieser Anwendung erforderte einige Sitzungen mit dem Kunden, da einige Anforderungen nicht genug präzise formuliert wurden.

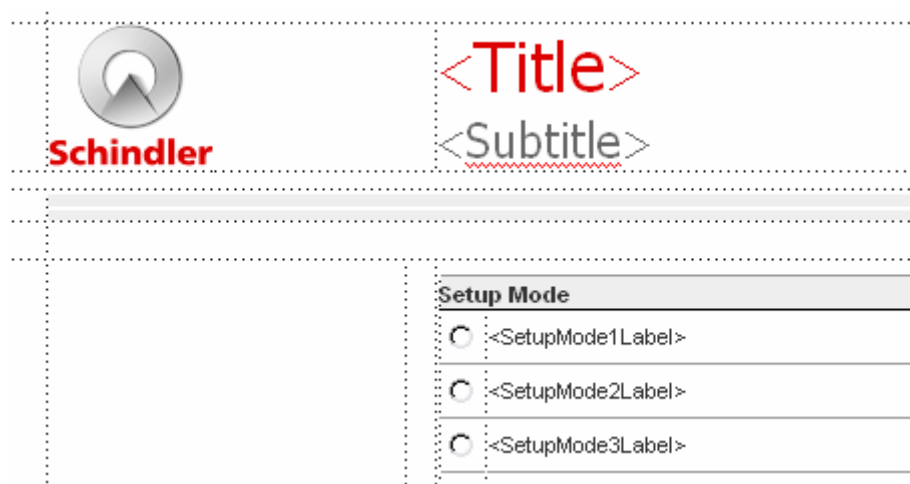


Abbildung 9: Setup Assistant, Entwurfsansicht.

5.2.3.1 Entwicklungsvorgehen

Mit nachfolgender Auflistung versuche ich das Vorgehen der Entwicklung der Anwendung „Setup Assistant“ aufzuzeigen:

- ? Neue HTA erstellen und Grundgerüst mit Hilfe von Helpomatic erzeugen. Siehe [\[Quelle 2.23\]](#).

- ? Aufbau der HTA analog zur Schindler Website. Grundgerüst der Tabellen kann kopiert übernommen werden. Unnötige Informationen werden herausgelöscht.
- ? Schindler Stylesheet einbinden.
- ? INI-Datei als Konfigurationsdatei für Setup Assistant erstellen. Diese soll vorerst nur Beschriftungsinformationen enthalten, wie z.B. „Title“, „Subtitle“, „CopyRightNotice“, und Vergleichbare. Später werden weitere funktionsrelevante Einträge erstellt. Zum Lesen dieser INI-Datei wurde aus Zeitgründen keine eigene Funktion geschrieben, sondern eine Funktion verwendet, welche nach Recherchen im Internet gefunden wurde. Siehe [\[Quelle 2.18\]](#).
- ? Nach der gruppierenden Sektorierung der Benutzeroberfläche wurde die Sichtbarkeit der einzelnen Elemente (Radio-Buttons, Input-Boxes, Combo-Boxes, DIVs, usw.) programmiert. Ich musste mir folgende Gedanken machen: Welche Elemente muss bei welcher Ausgangslage welche Standardwerte enthalten? Welche Elemente müssen bei welcher Benutzerangabe aktiviert/deaktiviert werden?
- ? Nach und nach setzte ich die Anforderungen in Scriptcode um.
- ? Informationen zu Harddisks und Partitionen werden via WMI des lokalen Computers ausgelesen und in Combo-Boxes präsentiert.
- ? Entwicklung von Funktionen zum Lesen von Unity Asset Informationen.
- ? Informationen zu den Software-Sets konnte ich aus Unity Konfigurationsdateien lesen und auf der Benutzeroberfläche von Setup Assistant darstellen.

5.2.3.2 Funktionen

In der nachfolgenden Tabelle wird eine Auswahl wesentlicher Funktionen präsentiert und beschrieben. Auf alle Code Fragmente von Setup Assistant kann aus Zeitgründen nicht eingegangen werden. Code Fragmente von externen Quellen werden jeweils mit einer Quellenangabe versehen.

Tabelle 19: Funktionen der Anwendung Setup Assistant.

Funktionsname	Beschreibung
GetVariables()	In dieser Initialisierungsfunktion werden Konfigurationsdateien, Asset Informationen und andere, vergleichbare Informationsquellen gelesen und in globalen Variablen gespeichert.
GetDisks()	Die Funktion liest sämtliche verfügbare lokale Datenträger aus der WMI Klasse "Win32_DiskDrive". Die vom System zur Verfügung gestellten Informationen über die lokalen Datenträger wird in eine dafür vorgesehene Combobox gespeichert. [Quelle 1.06]
GetPart(<i>disknumber</i>)	Diese Funktion wird aus der Funktion "GetDisks()" aufgerufen. Für jeden lokalen Datenträger (Parameter " <i>disknumber</i> ") werden die dazugehörigen Partitionen ausgelesen und in eine dafür vorgesehene Combobox gespeichert. Informationen über Partitionen können aus der WMI Klasse "Win32_DiskPartition" abgerufen werden. [Quelle 1.06]
Partitioning(<i>disknumber</i> , <i>partnumber</i> , <i>partsize</i> , <i>diskletter</i>)	Diese Funktion ist verantwortlich für die Partitionierung der gewählten Festplatte. Als Partitionierungssoftware wird das Windows-Bordmittel "Diskpart.exe" verwendet.
Format(<i>Drive</i> , <i>Label</i>)	Formatiert die lokalen Partitionen.
CleanEDUProfiles()	Löscht lokale Profildaten von gewählten EDU-Benutzerkonten.
CleanEDUProfilesServer()	Löscht servergespeicherte Profildaten von gewählten EDU-Benutzerkonten. Ausgenommen vom Löschvorgang sind die Dateien "Notes\<benutzername>.id" und "Notes\names.nsf" (Kundenwunsch).
GetINIString(<i>Section</i> , <i>Keyname</i> , <i>Default</i> , <i>Filename</i>)	Funktion zum Lesen von INI-Dateien. Mit dieser Funktion werden beispielsweise Abwendungskonfigurationen, Asset- und Unity-Konfigurationen gelesen. [Quelle 2.18]
Start()	Diese Funktion wird ausgeführt, wenn die Schaltfläche "Start" gedrückt wird. Aus ihr heraus wird die Eingabevalidierung ausgeführt, temporäre Variablen der Eingabefelder geschrieben und die entsprechenden Funktionen des gewählten Setup Modus' ausgeführt.

RunSetupMode1()	<p>Funktion, welche bei der Wahl des Setup Modus' "NewInstallation" ausgeführt wird. Folgende Aufgaben werden von dieser Funktion gelöst:</p> <ul style="list-style-type: none"> ? Neu erstellen der Unity Asset Konfiguration durch die Funktion "CreateUnityAssetFile()". ? Unity Asset Werte werden aktualisiert. Dazu wurde die Funktion "UpdateUnityAssetFile(Field, Value)" entworfen. ? Erstellen der Partition 0 und 1 auf der gewählten Festplatte. Partition 0 wird mit dem Laufwerksbuchstaben C:, Partition 1 mit Laufwerksbuchstaben D: versehen. ? Formatierung beider Partitionen der gewählten Festplatte. ? Starten des Windows Unattended Setups. Das Betriebssystem wird auf die Partition 0 der gewählten Festplatte installiert. ? Neustarten des Computers.
RunSetupMode2()	<p>Funktion, welche bei der Wahl des Setup Modus' "ReInstallation" ausgeführt wird. Folgende Aufgaben werden von dieser Funktion gelöst:</p> <ul style="list-style-type: none"> ? Neu erstellen der Unity Asset Konfiguration durch die Funktion "CreateUnityAssetFile()", sofern die entsprechende Option aktiviert wurde. ? Unity Asset Werte werden aktualisiert. Dazu wurde die Funktion "UpdateUnityAssetFile(Field, Value)" entworfen. ? Formatierung der Partition 0 der gewählten Festplatte. ? Starten des Windows Unattended Setups. Das Betriebssystem wird auf die Partition 0 der gewählten Festplatte installiert. ? Neustarten des Computers.
RunSetupMode3()	<p>Funktion, welche bei der Wahl des Setup Modus' "Clean Up" ausgeführt wird. Folgende Aufgaben werden von dieser Funktion gelöst:</p> <ul style="list-style-type: none"> ? Mit Hilfe der Funktion "UpdateSoftwareSet()" werden die Software-Set Informationen aktualisiert. ? Eigenschaft "OU" der Asset Konfiguration wird gegebenenfalls aktualisiert. ? Lokale und servergespeicherte Profile der ggf. markierten EDU-Benutzer werden gelöscht. ? Neustarten des Computers.

5.2.3.3 Datenfluss

Folgendes Schema visualisiert die Informationsakquisition. In der darauf folgenden Tabelle sind jeweils die Erklärungen zu den einzelnen Nummern ersichtlich.

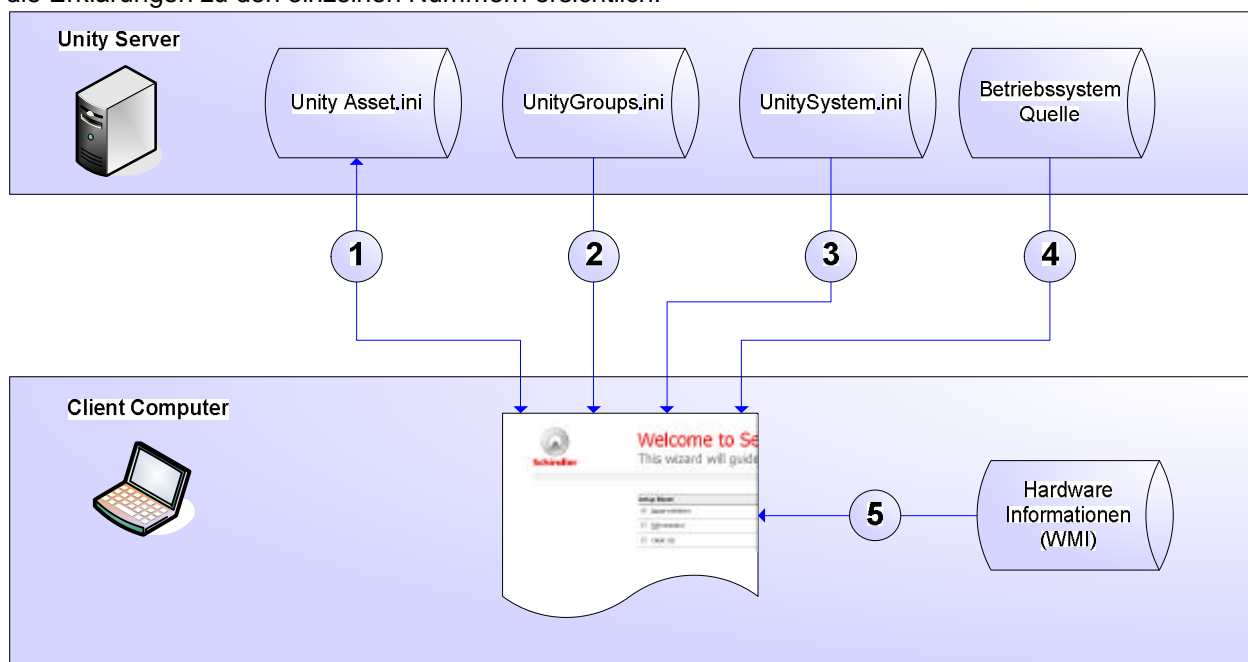


Abbildung 10: Informationsakquisition.

Tabelle 20: Bemerkungen zum Informationsfluss in Setup Assistant.

Nr. Beschreibung	
1	Lesen / Schreiben der Computer Konfiguration.
2	Lesen der Software-Sets.
3	Lesen von Unity der Konfiguration.
4	Kopieren von Quelldaten und Unattended Setup Informationen.
5	Lesen von Disk- und Partitions-Informationen.

5.2.3.4 Integrationsvorgang

- ? Die komplette Anwendungsumgebung der Hypertext-Anwendung „Setup Assistant“ wird in die dafür vorgesehene Netzwerkfreigabe des produktiven Unity Servers kopiert („\\infw0226\ui_prod\$\Tools\htaSetup“).
- ? Temporäre Dateien (*.db, usw...) werden aus dem Verzeichnis gelöscht.
- ? Änderungen an der Konfiguration können jederzeit vorgenommen werden.
- ? Da, wie bereits erwähnt, die Authentifizierung auf Verzeichnisebene stattfindet, müssen die Berechtigungen zur Benutzung von Setup Assistant direkt auf das Verzeichnis „htaSetup“ gesetzt werden. Die beiden berechtigten Gruppen sind „INF\GR_INF_UNITYADM“ und „INF\GR_INF_UNITYINST“, wobei die letztere Gruppe nur Leserechte besitzt.

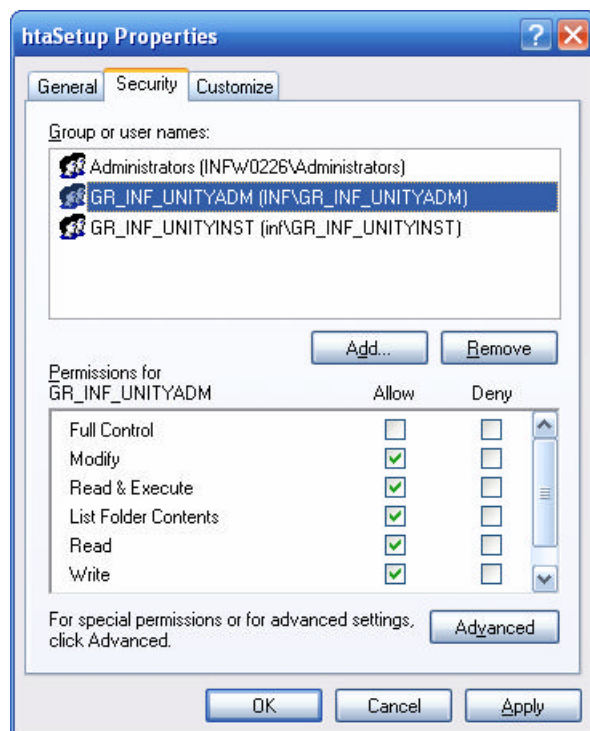


Abbildung 11: Lese- und Schreibberechtigung für Unity Administratoren.

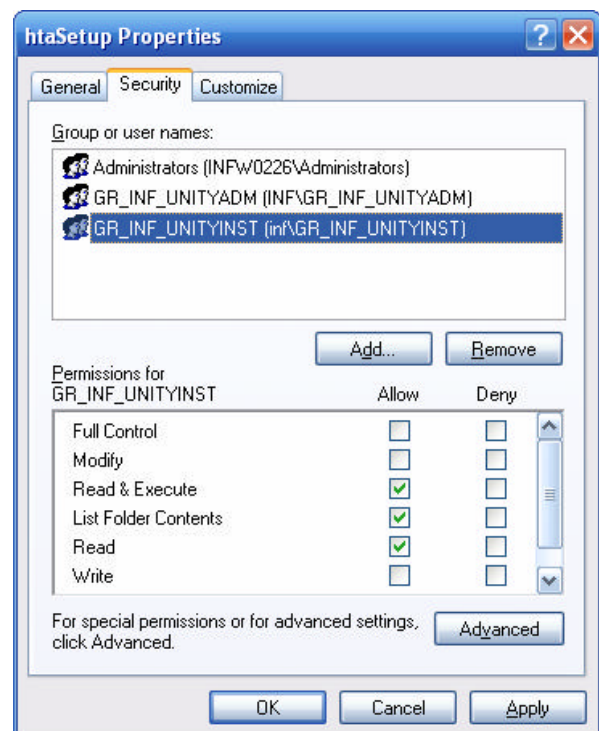


Abbildung 12: Leseberechtigung für Unity Supportmitarbeiter.

5.3 Integration der Softwareverteilung

An der Konfiguration der bestehenden Softwareverteilung Unity müssen einige Anpassungen vorgenommen werden, damit eine einwandfreie Zusammenarbeit zwischen dem Projektsystem und Unity möglich wird.

Es wird eine hierarchisch übergeordnete Organisation Unit namens „EDU_INF“ erstellt. Unter dieser Unity OU werden dann die verschiedenen Software-Sets errichtet. Die Benennung der übergeordneten OU sowie der Software-Sets kann frei gewählt werden. Es gilt jedoch zu beachten, dass der Name der übergeordneten OU jeweils in der Konfigurationsdatei von Setup Assistant angepasst wird.

5.3.1 Container für Software-Sets erstellen

Der nachfolgend dokumentierte Vorgang zur Erstellung des Containers für die Software-Sets muss nur einmal durchgeführt werden. Mehrere übergeordnete Organisationseinheiten, sprich Containers, werden von Setup Assistant nicht unterstützt.

- ? Unity Deployment Manager von produktivem Unity Server ("[\\infw0226\\ui_prod\\$\\Tools\\DplManager](#)") starten.
- ? Im Menü „View“ wird die Option „Groups“ gewählt.
- ? Rechtsklick auf „Organisation Units“ ↗ Kontextmenü „Add“ wählen, um eine neue OU hinzuzufügen.
- ? Die OU wird „EDU_INF“ benannt. Siehe nachfolgende Abbildung:

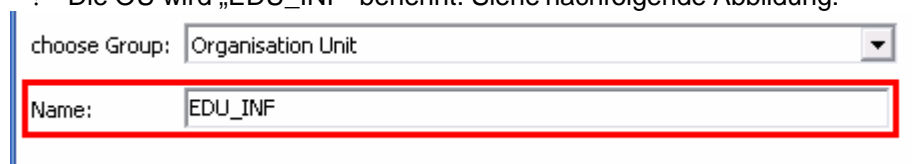


Abbildung 13: Erstellen der OU „EDU_INF“.

- ? Durch Klicken auf die Schaltfläche „Ok“ wird die OU erstellt.

5.3.2 Software-Set erstellen

Nachfolgend wird dokumentiert, wie ein Software-Set erstellt werden kann. Zum Erstellen weiterer Software-Sets kann dieser Vorgang wiederholt werden.

- ? Unity Deployment Manager von produktivem Unity Server ("[\\infw0226\\ui_prod\\$\\Tools\\DplManager](#)") starten.
- ? Im Menü „View“ wird die Option „Groups“ gewählt.
- ? Rechtsklick auf die OU „EDU_INF“ ↗ Kontextmenü „Add“ wählen, um eine neue OU hinzuzufügen.

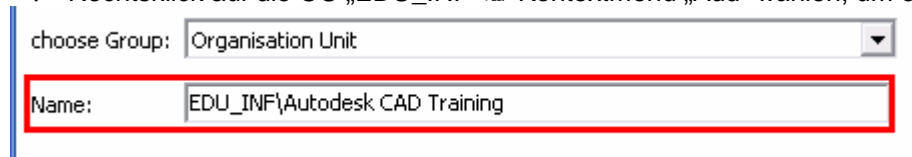


Abbildung 14: Erstellen des Software-Sets „Autodesk CAD Training“.

- ? Durch Klicken auf die Schaltfläche „Ok“ wird die OU erstellt.

5.3.3 Software-Set konfigurieren

- ? Unity Deployment Manager von produktivem Unity Server ("[\\infw0226\\ui_prod\\$\\Tools\\DplManager](#)") starten.
- ? Im Menü „View“ wird die Option „Groups“ gewählt.
- ? OU „EDU_INF\\Autodesk CAD Training“ wählen.
- ? Im Menü „Infos“ wird die Option „Assigned Packages“ gewählt.
- ? In der Hauptansicht erscheint nun eine Auswahl von Software Paketen. Die angezeigten Pakete beinhalten nicht nur explizit für die OU „Autodesk CAD Training“ konfigurierte Pakete, sondern auch von den übergeordneten OUs geerbte Software Pakete.
- ? Software, welche dem Software-Set „Autodesk CAD Training“ hinzugefügt werden soll, wird mit der rechten Maustaste angeklickt ↗ Kontextmenü „to Install“.
- ? Anschliessend wird die Konfiguration über die Schaltfläche „Save the current asset“ gespeichert.

6 Testphase

6.1 Ziel der Testphase


In der Testphase wird grundsätzlich geprüft, ob das Projektsystem die Anforderungen des Projektauftrags erfüllen. Tests werden stets unter der Annahme durchgeführt, dass das Projektsystem noch Fehler aufweist. Aus meiner Sicht (also aus der Sicht des Entwicklers) geht es in dieser Phase darum, die Anforderungen des Projekts mit dem Projektsystem zu vergleichen und zu testen, ob diese genügend erfüllt werden. Tests werden nicht nur vor dem unmittelbaren Ende des Projekts durchgeführt. Einzelne Komponenten werden im Prototyping-Verfahren bereits während der Entwicklungsphase getestet.

6.2 Planung der Tests

6.2.1 Methoden

Die Migration des Projektssystems in die produktive IT-Umgebung von Schindler muss sehr minutiös geplant und getestet werden. Dies bedingt eine kritische und rücksichtslose Verifikation des erstellten Projektsystems, sowohl von Entwickler- wie auch von Benutzer- und Kundenseite.

Die Tests werden in drei Testserien gegliedert:



Testserie 1 Entwicklertest	Der Entwickler prüft, ob Teile des Systems technische Mängel aufweisen. Dabei werden folgende Teststufen durchlaufen: ? Komponententest ? Integrationstest ? Sicherheitstest ? Systemtest
Testserie 2 Benutzertest	Ein Benutzer prüft bestimmte Testszenarien. In dieser Testserie wird geprüft, ob ein Benutzer in der Lage ist, das Projektsystem zu bedienen. Hier wird lediglich die Teststufe „Systemtest“ durchlaufen.
Testserie 3 Akzeptanztest	Der Auftraggeber prüft das Gesamtsystem gemäss den Anforderungen des Projektauftrags. Die Testphase wird mit dem durch den Auftraggeber visierten Akzeptanztest-Protokoll abgeschlossen.

6.2.2 Mittel

Tests werden vorzugsweise auf der vom Kunden zur Verfügung gestellten Hardware durchgeführt. Auf diesen Systemen werden primär komplette Systemtests durchgeführt. Komponenten- und Integrationstests werden auf dem Entwicklungssystem in virtuellen Systemen durchgeführt, da diese massiv kürzere Wartezeiten bieten. Als Virtualisierungslösung wird auf den Testsystemen die freie Beta-Version von VMware Server genutzt.

6.3 Testserie 1: Entwicklertest

6.3.1 Komponententest 1

Bedingungen

Der Komponententest 1 wurde am 27.03.2006 von Thomas Galliker durchgeführt. Dieser Test wurde auf einem virtuellen Testsystem durchgeführt.

Tabelle 21: Protokoll des Komponententests 1.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Ist Start vom Startmedium möglich?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde RAMDisk-Treiber geladen?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde das korrekte Tastaturlayout installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde NIC-Treiber geladen?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden die „Scripting Components“ korrekt installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden die „Factory Components“ fehlerfrei ausgeführt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird „Login Form“ fehlerfrei ausgeführt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird „Setup Assistant“ fehlerfrei ausgeführt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Auswertung

Der Test verlief fehlerfrei. Die Anwendung hat teilweise hohe Latenzzeiten. Siehe [\[Problem 7\]](#).

Massnahmen

Es müssen keine Massnahmen getroffen werden. Nach diesem Komponententest wurden trotzdem einige kleinere Design-Anpassungen vorgenommen, welche jedoch auf das Verhalten der Anwendungen keinen Einfluss hatten.

6.3.2 Komponententest 2

Bedingungen

Der Komponententest 2 wurde am 29.03.2006 von Thomas Galliker durchgeführt. In diesem Test wurde zum ersten Mal auf einem reellen Testsystem getestet. Das Testsystem war ein Laptop, welches vom Kunden freundlicherweise zur Verfügung gestellt wurde.

Tabelle 22: Protokoll des Komponententests 2.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Ist Start vom Startmedium möglich?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde RAMDisk-Treiber geladen?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde das korrekte Tastaturlayout installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wurde NIC-Treiber geladen?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Problemursache und –Lösung siehe Problemprotokoll [Problem 4] .
Werden die „Scripting Components“ korrekt installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden die „Factory Components“ fehlerfrei ausgeführt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Wird „Login Form“ fehlerfrei ausgeführt?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Folgefehler von [Problem 4] .
Wird „Setup Assistant“ fehlerfrei ausgeführt?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Folgefehler von [Problem 4] .

Auswertung

Der Test verlief meines Erstaunens nicht fehlerfrei. Als Symptom des Fehlers zeigte sich die fehlerhaft geladene Login Anwendung. Ich stellte fest, dass Windows PE keinen Netzwerkkartentreiber für das Gerät installieren konnte. Siehe [\[Problem 4\]](#). Folglich war auch der Start der Setup Assistant Anwendung nicht möglich.

Massnahmen

Broadcom NetXtreme NIC-Treiber mussten in Windows PE integriert werden. Nach der Integration wurden kompromisslos alle Testfälle erfüllt.

Siehe [Netzwerktreiber integrieren](#).

6.3.3 Integrationstest**Bedingungen**

Der Integrationstest wurde am 29.03.2006, in Anschluss an die Korrekturarbeiten des Komponententests 2, von Thomas Galliker durchgeführt. Getestet wurde wieder auf reellen Testsystemen.

Tabelle 23: Protokoll des Integrationstests.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Übergang von PE Start zur Initialisierung durch „Startnet.cmd“.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Interaktion zwischen Startnet.cmd und Factory Components.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Interaktion zwischen Factory Components und Login Form.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Interaktion zwischen Login Form und Setup Assistant.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Wiederholt lange Verzögerungen. Siehe [Problem 7] .
Interaktion zwischen Setup Assistant und Unity.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Auswertung

Der Test verlief fehlerfrei. Sämtliche Schnittstellen interagierten fehlerfrei.

Massnahmen

Es müssen keine Massnahmen getroffen werden.

6.3.4 Sicherheitstest**Bedingungen**

Der Sicherheitstest wurde am 30.03.2006 von Thomas Galliker durchgeführt. Getestet wurde auf einem reellen Testsystem. Zum Einsatz kam die Netzwerkanalyse-Software „Packetyzer“. Siehe [\[Quelle 2.37\]](#).

Tabelle 24: Protokoll des Sicherheitstests.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Ist die Sicherheit beim Anmelden via Login Form garantiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Auswertung

Der Sicherheitstest verlief positiv. Login Form authentifiziert über das sichere Authentifizierungsprotokoll Kerberos5, wie in der nachfolgenden Abbildung zu sehen ist. Es gibt keine sicherheitstechnischen Bedenken. Die aufgezeichnete Netzwerkanalyse befindet sich unter [Anhang G](#).

Num	Source Address	Dest Address	Summary
1	10.16.1.7	136.238.79.1	SMB: [TCP Out-Of-Order] Session Setup AndX Request
2	136.238.79.1	10.16.1.7	SMB: [TCP Out-Of-Order] Session Setup AndX Response
3	10.16.1.7	10.16.0.7	KRB5: AS-REQ
4	10.16.0.7	10.16.1.7	KRB5: KRB Error: KRB5KRB_ERR_RESPONSE_TOO_BIG
5	10.16.1.7	10.16.0.7	KRB5: [TCP Out-Of-Order] AS-REQ
6	10.16.0.7	10.16.1.7	KRB5: [TCP Out-Of-Order] AS-REP[Unresembled Packet][Unr...
7	10.16.1.7	10.16.0.7	KRB5: TGS-REQ[Unresembled Packet]
8	10.16.0.7	10.16.1.7	KRB5: KRB Error: KRB5KRB_ERR_RESPONSE_TOO_BIG
9	10.16.1.7	10.16.0.7	KRB5: [TCP Out-Of-Order] TGS-REQ[Unresembled Packet (in...
10	10.16.0.7	10.16.1.7	KRB5: [TCP Out-Of-Order] TGS-REP[Short Frame]

Abbildung 15: Kerberos5 Authentifizierung mit Login Form.

Massnahmen

Es müssen keine Massnahmen getroffen werden.

6.3.5 Systemtest

Bedingungen

Der Systemtest wurde am 30.03.2006 von Thomas Galliker durchgeführt. Getestet wurde auf einem realen Testsystem. Betroffen war das ganze Projektsystem.

Tabelle 25: Protokoll des Systemtests.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Erfolgreicher Start von Windows Preinstallation Environment.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Login Form startet fehlerfrei.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Authentifizierung verläuft problemlos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Assistant startet fehlerfrei.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „NewInstallation“			
Wird gewählte Festplatte korrekt formatiert / partitioniert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „ReInstallation“			
Wird der korrekte Computername angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird das korrekte Tastaturlayout angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird gewählte Festplatte korrekt formatiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „Clean Up“			
Wird der korrekte Computername angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird das korrekte Tastaturlayout angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden lokale EDU-Benutzerprofile gelöscht?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Werden servergespeicherte EDU-Benutzerprofile bereinigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Auswertung

Einwandfreier Testverlauf. Einige kleinere, unwesentliche Änderungen.

Massnahmen

Es müssen keine Massnahmen getroffen werden.

6.4 Testserie 2: Benutzertest

Bedingungen

Der Systemtest wurde am 30.03.2006 vom Testbenutzer Patrick Gehrig durchgeführt. Getestet wurde auf einem realen Testsystem. Betroffen war das ganze Projektsystem.

Tabelle 26: Benutzertest: Protokoll des Systemtests.

Testfall	Erfüllung		Kommentar
	Ja	Nein	
Erfolgreicher Start von Windows Preinstallation Environment.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Login Form startet fehlerfrei.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Authentifizierung verläuft problemlos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Assistant startet fehlerfrei.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „NewInstallation“			
Wird gewählte Festplatte korrekt formatiert / partitioniert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „ReInstallation“			
Wird der korrekte Computername angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird das korrekte Tastaturlayout angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird gewählte Festplatte korrekt formatiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Setup Modus „Clean Up“			
Wird der korrekte Computername angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird das korrekte Tastaturlayout angezeigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden lokale EDU-Benutzerprofile gelöscht?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Werden servergespeicherte EDU-Benutzerprofile bereinigt?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Wird Software gemäss gewähltem Software-Set installiert?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Auswertung

Persönliche Auswertung mit Testbenutzer

Massnahmen

Es müssen keine Massnahmen getroffen werden.

6.5 Testserie 3: Akzeptanztest

Bedingungen

Der Systemtest wurde am 31.03.2006 von Stefan Epp und Michael Jenny durchgeführt. Getestet wurde auf einem realen Testsystem. Betroffen war das ganze Projektsystem.

Tabelle 27: Akzeptanztest.

Anforderung	Erfüllung		Kommentar
	Ja	Nein	
Das Projektsystem kann sowohl für Schulungs- als auch für Notfall-Einsätze gebraucht werden.	<input type="checkbox"/>	<input type="checkbox"/>	
Laptop funktionieren auch jenseits des Schindler Netzwerks (offline).	<input type="checkbox"/>	<input type="checkbox"/>	
Die Partitionierung und Formatierung der lokalen Datenträger erfolgt anforderungsgemäss und vollautomatisch.	<input type="checkbox"/>	<input type="checkbox"/>	
Die Benutzeroberflächen wurden wunschgemäss gestaltet.	<input type="checkbox"/>	<input type="checkbox"/>	
Das Mass der Benutzerinteraktion liegt im gewünschten Rahmen.	<input type="checkbox"/>	<input type="checkbox"/>	
Die Änderungen an Konfigurationsinformationen von Computern werden jeweils wunschgemäss in die Softwareverteilung übertragen.	<input type="checkbox"/>	<input type="checkbox"/>	
Benutzerprofile werden einwandfrei bereinigt.	<input type="checkbox"/>	<input type="checkbox"/>	
Zu schulendes Software-Set kann gewählt und angewandt werden.	<input type="checkbox"/>	<input type="checkbox"/>	
Angewandtes Software-Set wird korrekt installiert.	<input type="checkbox"/>	<input type="checkbox"/>	
Das Resultat dieses Projekts erfüllt die Wünsche und Anforderungen des Projektauftrags vollumfänglich.	<input type="checkbox"/>	<input type="checkbox"/>	

Wichtig:

Durch die Visierung dieses Akzeptanztest-Protokolls bestätigt der Kunde, dass das Projekt anforderungsgemäss umgesetzt wurde. Änderungen im Rahmen dieses Projekts finden ab diesem Zeitpunkt keine mehr statt.

Ebikon, 31.03.2006

.....
Stefan Epp
 Auftraggeber
 Schindler Informatik AG

.....
Michael Jenny
 Auftraggeber (Stv.)
 Schindler Informatik AG

7 Projektauswertung

7.1 Planungsauswertung

7.1.1 Ist-Zeitplan

Der Ist-Zeitplan wurde anhand der Tagesprotokolle erstellt. Das erstellte MS Project Gantt-Diagramm befindet sich unter [Anhang D](#).

7.1.2 Planungsdiskrepanz

Nach der Kontrolle der Tagesprotokolle und der Erstellung des Ist-Zeitplans hatte ich genügend Informationen, um die Auswertung der Ist-/Soll-Zeitpläne vorzunehmen. Visuelle Darstellungen sagen in der Regel mehr als viele Sätze. Das nachfolgende Auswertungsdiagramm soll diese Aussage bestätigen:

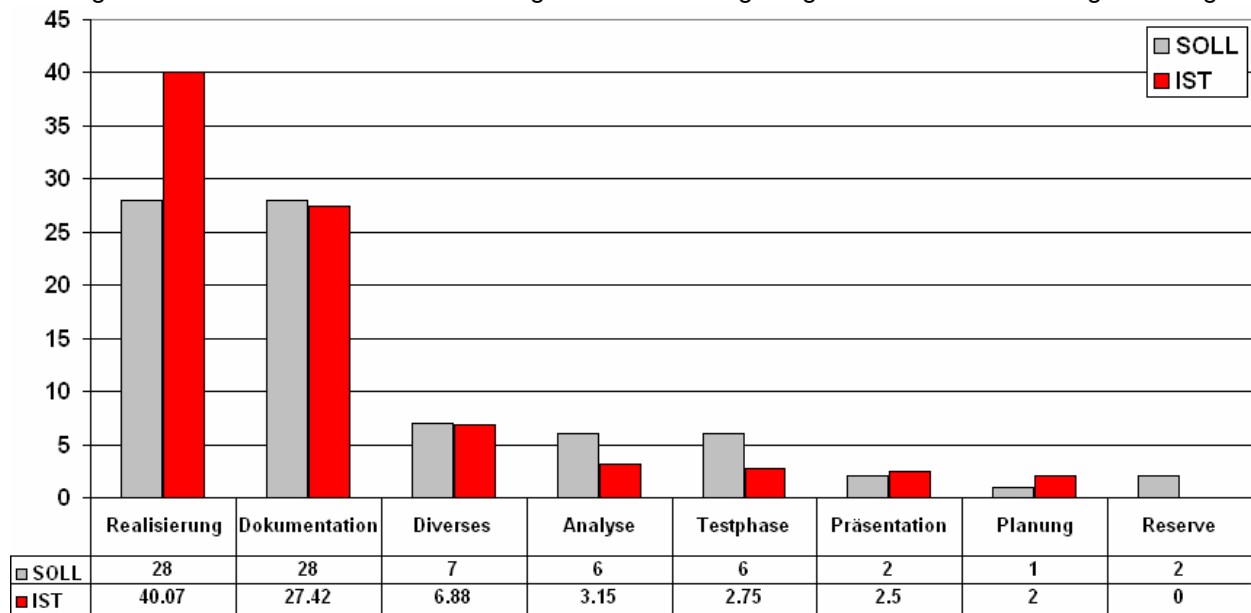


Abbildung 16: Auswertungsdiagramm mit Ist/Soll-Vergleich.

Die einzelnen Posten sind nach der Grösse des geplanten Zeitaufwands, in absteigender Reihenfolge geordnet (davon ausgenommen ist der Posten „Reserve“). Auskunft über die genauen Abweichungen soll uns folgende Tabelle geben.

Tabelle 28: Planungsauswertung mit Abweichungsberechnung.

	Planung (Soll)		Realität (Ist)		Abweichung
Realisierung	28 Stunden	(35%)	40.07 Stunden	(47.3%)	+ 43.1%
Dokumentation	28 Stunden	(35%)	27.42 Stunden	(32.4%)	- 2.1%
Diverses	7 Stunden	(8.75%)	6.88 Stunden	(8.1%)	- 1.7%
Analyse	6 Stunden	(7.5%)	3.15 Stunden	(3.7%)	- 47.5%
Testphase	6 Stunden	(7.5%)	2.75 Stunden	(3.2%)	- 54.2%
Präsentation	2 Stunden	(2.5%)	2.5 Stunden	(3.0%)	+ 25.0%
Planung	1 Stunde	(1.25%)	2 Stunden	(2.4%)	+ 100.0%
Reserve	2 Stunden	(2.5%)	0 Stunden	(0.0%)	- 100.0%
Total	80 Stunden	(100%)	84.77 Stunden	(106%)	6.0%

7.1.3 Kommentar

Die Planungswerte sind reine Erfahrungswerte aus früheren Projekten. Am augenfälligsten ist die massive Diskrepanz zwischen Ist/Soll beim Posten „Realisierung“. Diese Abweichung von der Planung begründet sich im massiv grösseren Entwicklungsaufwand der Anwendung „Setup Assistant“. Einige schwerwiegende Probleme sorgten zudem für Zeitverzögerungen im ordentlichen Projektablauf.

Eine ebenfalls grössere Abweichung ins beim Posten „Testphase“ festzustellen: Diese Diskrepanz ist dadurch entstanden, weil viele Tests bereits während der Entwicklungsarbeit stattfanden (Prototyping-Verfahren). So konnte wertvolle Zeit für ausgiebige Systemtests eingespart werden. Allgemein waren Tests nicht von allzu langer Dauer.

7.2 Persönlicher Erfahrungsbericht

Allgemein

Dieses Projekt forderte sehr viel Denkarbeit. Es waren viele heikle Angelegen zu überwinden, welche interdisziplinäres Denken forderten. Obwohl ich schon mehrere Testläufe von Projektarbeiten im Stil einer Individuellen Produktivarbeit (IPA) erarbeitet habe, traf ich in diesem Projekt wieder völlig neue Probleme an.

Planung

Bei der Planung konnte ich mich sehr an meine persönlichen Erfahrungswerte halten. Das Gantt-Diagramm des Soll-Zeitplans (und später auch jenes des Ist-Zeitplans) war eine völlig neue Erfahrung für mich.

Analyse

Diese Phase war die erste grösste Herausforderung im Projekt. Aus Zeitgründen beschränkte ich mich auf einige wichtige Evaluationen, wobei selbstverständlich noch mehr und noch detaillierter evaluiert hätte werden können. Die Evaluationsphase brachte einige bisher unbekannte Herausforderungen mit sich.

Realisierung

Der wohl grösste Teil der Projektarbeit stellte die Realisierungsphase dar. Ich war von der Projektarbeit sehr gefordert. Zudem kam gegen Ende des Projekts noch zusätzlich starker Zeitdruck, was mich enorm stresste. In der Realisierungsphase konnte ich mir viele neue Erfahrungen aneignen. Einige Erfahrungen werden nachfolgend aufgelistet:

- ? Vertiefen und Erweitern der VBScript/HTA Kenntnisse.
- ? Vertiefen und Erweitern der Windows Systemkenntnisse.
- ? Aufgabenstellungen und Kundenwünsche korrekt interpretieren lernen.
- ? Kompromisse eingehen; Prioritäten setzen.
- ? Systematisches Problemlöseverfahren – auch in Stresssituationen.

Testphase

Diese Phase war wiederum völlig neu für mich. Die Testprotokolle, insbesondere die Benutzertestprotokolle, forderten flexibles Denken. Die Tests mit unbeteiligten Testpersonen bestätigten einmal mehr, dass man als Entwickler nicht alle Fehler und Tücken des eigenen Produkts sehen kann.

8 Arbeitsprotokolle

8.1 Wochenübersicht

8.1.1 Woche 1

Woche 1 (13.03.2006 - 17.03.2006)	Datum	Tätigkeiten, Bemerkung
	Montag, 13.03.2006	<i>Planung:</i> ? Projektplanung (Gantt-Diagramm). <i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Dokumentationsvorlage anpassen. ? Soll-Zeitplan erstellen. ? Sitzungsprotokoll 1 erstellen. <i>Diverses:</i> ? Sitzung 1: Kickoff Meeting. ? Projektordner einrichten.
	Dienstag, 14.03.2006	? Schule (BM).
	Mittwoch, 15.03.2006	<i>Analyse:</i> ? Projektauftrag in Teilaufgaben dividieren. ? Analysieren des Gesamtsystems; Überlegungen zum Ablaufdiagramm. ? Preinstallation Environments evaluieren. ? RAMDisk-Produkte evaluieren. ? Installationsmethoden evaluieren. <i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Aufgabenstellung aus WO übertragen. ? Auftragsdivision dokumentieren. ? Evaluation des Preinstallation Environments dokumentieren. ? Evaluation der RAMDisk-Produkte dokumentieren. ? Evaluation der Installationsmethoden dokumentieren. ? Analyse des Gesamtsystems dokumentieren.
	Donnerstag, 16.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Tabelle der Projektkosten ergänzen. ? Sitzungsprotokoll 2 erstellen. ? Entwicklungsumgebung für Windows PE dokumentieren. ? Erstellungsvorgang Windows PE und ISO Image dokumentieren. ? Dokumentation der Shutdown-Software. <i>Realisierung:</i> ? Entwicklungsumgebung einrichten. ? Windows PE / ISO Image erstellen. ? Integration der Shutdown-Software. ? Erstellung und Integration der Scripting Components. ? Entwicklung der Factory Components. <i>Diverses:</i> ? Sitzung 2: Expertenbesuch.
	Freitag, 17.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Integration der Scripting Components dokumentieren. ? Entwicklung und Integration der Factory Components dokumentieren. ? Importieren des Dokumentationsteils „Tastaturlayout anpassen“. ? Importieren des Dokumentationsteils „RAMDisk integrieren“. <i>Realisierung:</i> ? Entwicklung der Factory Components; Factory HTA. ? Anpassungen an Login Form HTA. ? Tastaturlayout von Windows PE anpassen. ? RAMDisk Treiber in Windows PE integrieren.

8.1.2 Woche 2

		Datum	Tätigkeiten, Bemerkung
Woche 2 (20.03.2006 - 24.03.2006)		Montag, 20.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Sitzungsprotokoll 3 erstellen. ? Ablaufdiagramm in MS Visio erstellen. <i>Realisierung:</i> ? Entwicklung Setup Assistant HTA. ? Anpassungen an Login Form HTA. <i>Diverses:</i> ? Sitzung 3: Besprechung
		Dienstag, 21.03.2006	? Schule (BM).
		Mittwoch, 22.03.2006	? Schule (BS).
		Donnerstag, 23.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Vorgehensweise der HTA Entwicklung dokumentieren. ✎ Entwicklungsvorgehen für Setup Assistant dokumentieren. ✎ Entwicklungsvorgehen für Login Form dokumentieren. ? Anpassungen an Ablaufdiagramm vornehmen; einbinden in Dokumentation. ? Erstellen des Problemprotokolls. <i>Realisierung:</i> ? Entwicklung Setup Assistant HTA. ? Anpassungen an den Factory Components; Factory HTA.
		Freitag, 24.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Quellenverzeichnis erstellen; Quellen organisieren und referenzieren. ? Anpassungen an den Factory Components dokumentieren. ? Integration von Unity dokumentieren. ? Überarbeitung der Dokumentation. <i>Realisierung:</i> ? Integration von Unity. ? Entwicklung Setup Assistant; Setup Modus „NewInstallation“.

8.1.3 Woche 3

		Datum	Tätigkeiten, Bemerkung
Woche 3 (27.03.2006 - 31.03.2006)		Montag, 27.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Vorlage für Testprotokolle erstellen. <i>Test:</i> ? Testserie 1 – Komponententest 1. <i>Realisierung:</i> ? Entwicklung Setup Assistant HTA; Setup Modus „Reinstallation“ und „Clean Up“.
		Dienstag, 28.03.2006	? Schule (BM).
		Mittwoch, 29.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Dokumentationsanhang aktualisieren. ? Integration der Netzwerktreiber dokumentieren. <i>Test:</i> ? Testserie 1 – Komponententest 2. ? Testserie 1 – Integrationstest. <i>Realisierung:</i> ? Entwicklung Setup Assistant HTA; Setup Modus „Clean Up“.
		Donnerstag, 30.03.2006	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen. ? Projektauswertung dokumentieren. ? Überarbeitung der Dokumentation. ? Entwicklung von Setup Assistant dokumentieren. ? Testserien dokumentieren. <i>Realisierung:</i> ? Entwicklung Setup Assistant HTA. <i>Test:</i> ? Testserie 1 – Sicherheitstest. ? Testserie 1 – Systemtest. ? Testserie 2 – Benutzertest.
		Freitag, 31.03.2006 ¹	<i>Dokumentation:</i> ? Tages- / Wochenprotokoll nachführen und kontrollieren. ? Abbildungs-, Tabellen und Quellenverzeichnis aktualisieren. ? Überarbeitung der Dokumentation. ? Projektauswertung dokumentieren. ? Erfahrungsbericht verfassen. <i>Test:</i> ? Testserie 3 - Akzeptanztest durch Kunden. <i>Diverses:</i> ? Sitzung 4: Projektübergabe während Akzeptanztest <i>Präsentation:</i> ? Präsentation erstellen. ? Live Demonstration vorbereiten.

¹ Da ich zu Projektbeginn auf Kundeninformationen warten musste, konnte ich die geplanten 80 Arbeitsstunden für dieses Projekt nicht planmässig bis Donnerstag, 30.03.2006 nutzen. Der Abgabetermin wurde nach Absprache mit den Experten auf Freitag, 31.03.2006 verschoben.

8.2 Tagesprotokolle

8.2.1 Tagesprotokoll vom Montag, 13.03.2006

Zeitübersicht:

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	0.15	80.00	Tages- / Wochenprotokoll nachführen.	
1.00	0.15	79.85	Projektplanung	Zeitaufwände abschätzen, Priorisierung.
0.75	1.15	78.85	Soll-Zeitplan erstellen.	Siehe Soll-Zeitplan .
0.15	1.90	78.10	Projektordner einrichten.	
0.25	2.05	77.95	Dokumentationsvorlage anpassen.	
1.00	2.30	77.70	Sitzung 1: Kickoff Meeting	
0.50	3.30	76.70	Sitzungsprotokoll 1 erstellen.	Siehe Sitzungsprotokoll 1 .
3.80	3.80	76.20		

Fortschrittsanzeige:

Verwendete Zeit: 4.75%

Verbleibende Zeit: 95.25%



8.2.2 Tagesprotokoll vom Dienstag, 14.03.2006

Zeitübersicht:

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.00	3.80	76.20	-	Schule (BM).
0.00	3.80	76.20		

Fortschrittsanzeige:

Verwendete Zeit: 4.75%

Verbleibende Zeit: 95.25%



¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.

² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.

³ Arbeitsstunden verbleibend bis Projektende.

8.2.3 Tagesprotokoll vom Mittwoch, 15.03.2006

Zeitübersicht:

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	3.80	76.20	Tages- / Wochenprotokoll nachführen.	
0.25	3.95	76.05	Aufgabenstellung aus WO übertragen.	
1.25	4.20	75.80	Preinstallation Environments evaluieren.	
1.25	5.45	74.55	Evaluation des Preinstallation Environments dokumentieren.	
0.15	6.70	73.30	Projektauftrag in Teilaufgaben dividieren.	
0.33	6.85	73.15	Auftragsdivision dokumentieren	
0.50	7.18	72.82	Analysieren des Gesamtsystems.	
0.50	7.68	72.32	Analyse des Gesamtsystems dokumentieren.	
1.00	8.18	71.82	Installationsmethoden evaluieren.	
0.75	9.18	70.82	Evaluation der Installationsmethoden dokumentieren.	
0.50	9.93	70.07	RAMDisk-Produkte evaluieren.	
0.75	10.43	69.57	Evaluation der RAMDisk-Produkte dokumentieren.	
7.38	11.18	68.82		

Fortschrittsanzeige:

Verwendete Zeit: 14%

Verbleibende Zeit: 86%



8.2.4 Tagesprotokoll vom Donnerstag, 16.03.2006

Zeitübersicht:

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	11.18	68.82	Tages- / Wochenprotokoll nachführen.	
0.15	11.33	68.67	Tabelle der Projektkosten ergänzen.	
0.33	11.48	68.52	Entwicklungsumgebung einrichten.	
0.75	11.81	68.19	Entwicklungsumgebung für Windows PE dokumentieren.	
0.33	12.56	67.44	Windows PE / ISO Image erstellen.	
0.50	12.89	67.11	Erstellungsvorgang Windows PE und ISO Image dokumentieren.	
1.50	13.39	66.61	Entwicklung der Factory Components.	
0.33	14.89	65.11	Erstellung und Integration der Scripting Components.	
1.00	15.22	64.78	Dokumentation und Integration der Shutdown-Software.	
2.00	16.22	63.78	Sitzung 2: Expertenbesuch.	
0.50	18.22	61.78	Sitzungsprotokoll 2 erstellen.	Siehe Sitzungsprotokoll 2.
7.54	18.72	61.28		

Fortschrittsanzeige:

Verwendete Zeit: 23.4%

Verbleibende Zeit: 76.6%



¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.

² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.

³ Arbeitsstunden verbleibend bis Projektende.

8.2.5 Tagesprotokoll vom Freitag, 17.03.2006

Zeitübersicht:

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	18.72	61.28	Tages- / Wochenprotokoll nachführen.	
0.50	18.87	61.13	Integration der Scripting Components dokumentieren.	
2.50	19.37	60.63	Entwicklung und Integration der Factory Components dokumentieren.	
0.25	21.87	58.13	Importieren des Dokumentationsteils „Tastaturlayout anpassen“ aus Lehrlingsdokumentation.	[Quelle 1.01]
0.25	22.12	57.88	Tastaturlayout von Windows PE anpassen.	
0.25	22.37	57.63	Importieren des Dokumentationsteils „RAMDisk integrieren“ aus Lehrlingsdokumentation.	[Quelle 1.01]
1.00	22.62	57.38	Entwicklung der Factory Components; Factory HTA.	
2.50	23.62	56.38	Anpassungen an Login Form HTA.	
1.00	26.12	53.88	Überarbeitung der Dokumentation.	
8.40	27.12	52.88		

Fortschrittsanzeige:

Verwendete Zeit: 33.9%

Verbleibende Zeit: 66.1%



8.2.6 Tagesprotokoll vom Montag, 20.03.2006

Zeitübersicht:

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	27.12	52.88	Tages- / Wochenprotokoll nachführen.	
0.50	27.27	52.73	Detailliertes Ablaufdiagramm in MS Visio erstellen.	
5.00	27.77	52.23	Entwicklung Setup Assistant HTA.	
1.00	32.77	47.23	Anpassungen an Login Form HTA.	
1.00	33.77	46.23	Sitzung 3: Besprechung	
0.50	34.77	45.23	Sitzungsprotokoll 3 erstellen.	Siehe Sitzungsprotokoll 3.
8.15	35.27	44.73		

Fortschrittsanzeige:

Verwendete Zeit: 44.1%

Verbleibende Zeit: 55.9%



¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.

² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.

³ Arbeitsstunden verbleibend bis Projektende.

8.2.7 Tagesprotokoll vom Dienstag, 21.03.2006**Zeitübersicht:**

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.00	35.27	44.73	-	Schule (BM).
0.00	35.27	44.73		

Fortschrittsanzeige:

Verwendete Zeit: 44.1%

Verbleibende Zeit: 55.9%

**8.2.8 Tagesprotokoll vom Mittwoch, 22.03.2006****Zeitübersicht:**

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.00	35.27	44.73	-	Schule (BS).
0.00	35.27	44.73		

Fortschrittsanzeige:

Verwendete Zeit: 44.1%

Verbleibende Zeit: 55.9%

¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.³ Arbeitsstunden verbleibend bis Projektende.

8.2.9 Tagesprotokoll vom Donnerstag, 23.03.2006

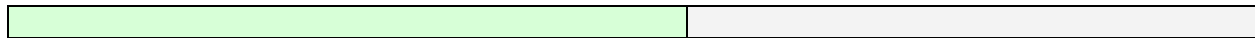
Zeitübersicht:

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	35.27	44.73	Tages- / Wochenprotokoll nachführen.	
2.00	35.42	44.58	Vorgehensweise der HTA Entwicklung dokumentieren.	
5.00	37.42	42.58	Entwicklung Setup Assistant HTA.	
0.50	42.42	37.58	Anpassungen an den Factory Components; Factory HTA.	
0.25	42.92	37.08	Anpassungen an Ablaufdiagramm vornehmen; einbinden in Dokumentation.	
0.33	43.17	36.83	Erstellen des Problemprotokolls.	
8.23	43.50	36.50		

Fortschrittsanzeige:

Verwendete Zeit: 54.4%

Verbleibende Zeit: 45.6%



8.2.10 Tagesprotokoll vom Freitag, 24.03.2006

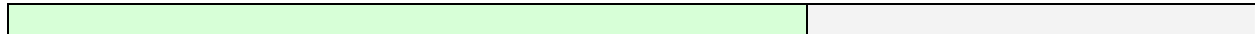
Zeitübersicht:

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	43.50	36.50	Tages- / Wochenprotokoll nachführen.	
1.00	43.65	36.35	Anpassungen an den Factory Components dokumentieren.	
0.33	44.65	35.35	Integration von Unity.	
0.50	44.98	35.02	Integration von Unity dokumentieren.	
5.00	45.48	34.52	Entwicklung Setup Assistant HTA.	Setup Modus "NewInstallation". Siehe Quellenverzeichnis .
0.75	50.48	29.52	Quellenverzeichnis erstellen; Quellen organisieren und referenzieren.	
7.73	51.23	28.77		

Fortschrittsanzeige:

Verwendete Zeit: 64.0%

Verbleibende Zeit: 36.0%



¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.

² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.

³ Arbeitsstunden verbleibend bis Projektende.

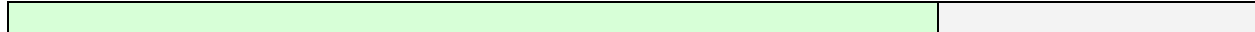
8.2.11 Tagesprotokoll vom Montag, 27.03.2006**Zeitübersicht:**

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	51.23	28.77	Tages- / Wochenprotokoll nachführen.	
1.00	51.38	28.62	Vorlage für Testprotokolle erstellen.	
0.25	52.38	27.62	Testserie 1 – Komponententest 1.	Siehe Komponententest 1 .
7.00	52.63	27.37	Entwicklung Setup Assistant HTA.	Setup Modus „Reinstallation“ und „Clean Up“.
8.40	59.63	20.37		

Fortschrittsanzeige:

Verwendete Zeit: 74.5%

Verbleibende Zeit: 25.5%

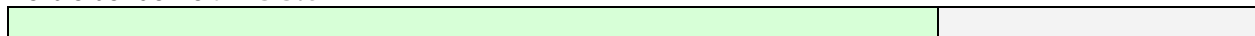
**8.2.12 Tagesprotokoll vom Dienstag, 28.03.2006****Zeitübersicht:**

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.00	59.63	20.37	-	Schule (BM).
0.00	59.63	20.37		

Fortschrittsanzeige:

Verwendete Zeit: 74.5%

Verbleibende Zeit: 25.5%

¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.³ Arbeitsstunden verbleibend bis Projektende.

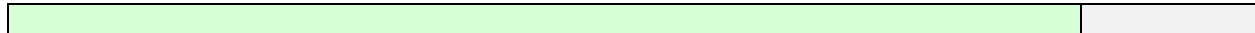
8.2.13 Tagesprotokoll vom Mittwoch, 29.03.2006**Zeitübersicht:**

Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	59.63	20.37	Tages- / Wochenprotokoll nachführen.	
0.25	59.78	20.22	Dokumentationsanhang aktualisieren.	
0.50	60.03	19.97	Testserie 1 – Komponententest 2.	Siehe Komponententest 2.
0.50	60.53	19.47	Testserie 1 – Integrationstest.	Siehe Integrationstest.
1.00	61.03	18.97	Testserien dokumentieren.	
5.25	62.03	17.97	Entwicklung Setup Assistant HTA.	Setup Modus „Clean Up“.
1.50	67.28	12.72	Integration der Netzwerktreiber dokumentieren.	Siehe Problem 4.
9.15	68.78	11.22		

Fortschrittsanzeige:

Verwendete Zeit: 86.0%

Verbleibende Zeit: 14.0%

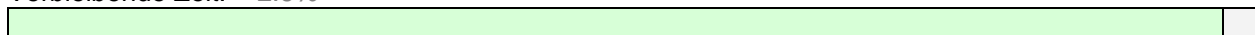
**8.2.14 Tagesprotokoll vom Donnerstag, 30.03.2006****Zeitübersicht:**

Std. Heute	Std. Verg.	Std. Verbl.	Ausgeführte Arbeit	Bemerkung
0.15	68.78	11.22	Tages- / Wochenprotokoll nachführen.	
0.50	68.93	11.07	Projektauswertung dokumentieren.	
1.00	69.43	10.57	Überarbeitung der Dokumentation.	Überprüfung von Rechtschreibung, Grammatik, Logik, Textaufbau und Style.
5.00	70.43	9.57	Entwicklung Setup Assistant HTA.	Setup Modus „Clean Up“.
1.50	75.43	4.57	Entwicklung von Setup Assistant dokumentieren.	
0.50	76.93	3.07	Testserie 1 – Sicherheitstest.	Siehe Sicherheitstest.
0.25	77.43	2.57	Testserie 1 – Systemtest.	Siehe Systemtest.
0.25	77.68	2.32	Testserie 2 – Benutzertest.	Siehe Benutzertest.
9.15	77.93	2.07		

Fortschrittsanzeige:

Verwendete Zeit: 97.4%

Verbleibende Zeit: 2.6%

¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.³ Arbeitsstunden verbleibend bis Projektende.

8.2.15 Tagesprotokoll vom Freitag, 31.03.2006**Zeitübersicht:**

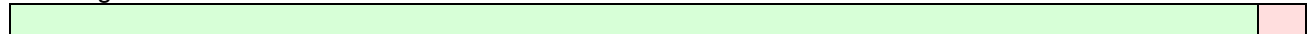
Std. ¹ Heute	Std. ² Verg.	Std. ³ Verbl.	Ausgeführte Arbeit	Bemerkung
0.25	77.93	2.07	Tages- / Wochenprotokoll nachführen und kontrollieren	
0.33	78.18	1.82	Abbildungs-, Tabellen und Quellenverzeichnis aktualisieren.	
1.00	78.51	1.49	Überarbeitung der Dokumentation.	
1.00	79.51	0.49	Projektauswertung dokumentieren.	
0.75	80.51	-0.51	Erfahrungsbericht verfassen.	
0.50	81.26	-1.26	Testserie 3 - Akzeptanztest durch Kunden.	Siehe Akzeptanztest .
0.50	81.76	-1.76	Sitzung 4: Projektübergabe	
1.50	82.26	-2.26	Präsentation erstellen.	
1.00	83.76	-3.76	Live Demonstration vorbereiten	
6.83	84.76	-4.76		

Fortschrittsanzeige:

Verwendete Zeit: 100%

Verbleibende Zeit: 0%

Überzogene Zeit: 6%

¹ Arbeitsstunden für ausgeführte Arbeit bzw. Arbeitsstundentotal heute.² Arbeitsstunden, welche seit Projektbeginn verstrichen sind.³ Arbeitsstunden verbleibend bis Projektende.

8.3 Sitzungsprotokolle

8.3.1 Sitzung 1: Kickoff Meeting

Nr	1
Datum / Zeit	Montag, 13.März 2006, 15.00 - 16.00 Uhr.
Ort	Schindler Informatik AG, Konferenzraum "Rigi".
Anwesende	<ul style="list-style-type: none"> ? Patrick Gehrig ? Stefan Epp ? Thomas Galliker
Pendenzen	<ul style="list-style-type: none"> ? Aufgabenstellung zusammen besprechen. ? Kundenansicht und -anforderung besprechen. ? Missverständnisse im Vorhinein eliminieren. ? Planung besprechen.
Protokoll	<ul style="list-style-type: none"> ? Zu Beginn der Sitzung zeigte ich den Anwesenden mein Planungsdiagramm. Der Kunde Stefan Epp erwähnte dabei, dass er vom 16.3.-24.3.06 nicht im Geschäft ist, aber von Michael Jenny vertreten wird. ? Die Aufgabenstellung wurde schrittweise abgearbeitet. Einige Lösungsansätze wurden diskutiert. Teilweise wurden bereits erste potenzielle Fehlerquellen festgestellt. ? Folgendes stellte sich bei der gemeinsamen Interpretation der Aufgabenstellung heraus: <ul style="list-style-type: none"> o Grundlage für die Arbeiten soll eine Preinstallation Plattform bieten. Der Produktwahl muss eine Evaluation zugrunde liegen. Wichtigstes Evaluationskriterium: Kurze Startzeit. o Für Schulungen werden temporäre EDU-Benutzerprofile benötigt. Diese sollen in einer AD OU angelegt werden. o Als "Software-Set" bezeichnet der Auftraggeber die Zusammenfassung von mehreren Software-Paketen in einer Unity-OU. o Es soll grundsätzlich zwei verschiedene Installationsszenarien geben, die der Kunde wählen kann: Modus "NewInstallation": <ul style="list-style-type: none"> ✍ System wird komplett neu installiert. ✍ Software wird entsprechend dem gewählten Software-Set installiert. ✍ EDU-Benutzerprofil muss in AD neu erstellt werden. ✍ Asset Informationen werden übernommen, sofern vorhanden. Andernfalls müssen die Asset Infos vom Supporter ergänzt werden können. o Modus "ReInstallation": <ul style="list-style-type: none"> ✍ System wird nicht neu aufgesetzt. ✍ Neue Software wird entsprechend dem gewählten Software-Set installiert. Bereits installierte Software muss deinstalliert werden. ✍ EDU-Benutzerprofil muss in AD neu erstellt werden. Lokales Profil wird gelöscht. ✍ Asset Informationen werden übernommen, sofern vorhanden. Andernfalls müssen die Asset Infos vom Supporter ergänzt werden können. ? Zusätzlich kam zum Ausdruck, dass die Benutzeroberfläche vom Medium gestartet werden soll. ? Stefan Epp erwähnte, dass "Krankheit" ein weiterer Risikofaktor des Projekts darstellen könne ✍ wird in der Dokumentation entsprechend ergänzt.
Nächste Sitzung	Montag, 20.März 2006, 10.00 - 11.00 Uhr, Schindler Informatik AG, Konferenzraum "Rigi".

8.3.2 Sitzung 2: Expertenbesuch

Nr	2
Datum / Zeit	Donnerstag, 16.März 2006, 08.00 - 10.00 Uhr.
Ort	Schindler Informatik AG, Konferenzraum "Mönch".
Anwesende	<ul style="list-style-type: none"> ? Reto Burger, Hauptexperte ? Ursula Reinhard, Expertin ? Patrick Gehrig ? Thomas Galliker
Gesprächsinhalt	<p>? Bewertung der Arbeit Nach einer gegenseitigen Begrüssungsrunde zeigte uns Herr Burger die Bewertungsvorlage, mit welcher die Projektarbeit nach verschiedenen Kriterien bewertet werden wird. Die gesetzten Termine wurden zur Vergewisserung nochmals zusammen angesehen.</p> <p>? Planung Nachdem die ersten Fragen geklärt wurden, konnte ich den Experten meine Projektplanung zeigen. Herr Burger brachte die konstruktive Kritik, dass die Dokumentationsphase in meinem Gantt-Diagramm ein grösseres Spektrum abdecken könne. Die für die Dokumentation geplanten Stunden würden ja schliesslich auf verschiedene Posten (wie z.B. Analyse, Realisierung, usw.) aufgeteilt.</p> <p>? Analyse Auch beim Betrachten der bereits dokumentierten Analysephase konnten einige Fragen beantwortet werden. Fragen befassten sich in diesem Gespräch ausschliesslich über die Formalität der Dokumentation als über die technische und/oder grammatikalische Korrektheit des Textes.</p> <p>? Präsentation Zum Schluss wurden einige Merkpunkte zur Präsentation ausgesprochen. Herr Burger informierte Patrick Gehrig, dass er während der Präsentation stillschweigend zuschauen muss und dass ein Eingreifen seinerseits Abzug geben könne. Der punktuell vorgeschlagene Präsentationsinhalt wurde von den Experten gutgeheissen.</p>
Nächste Sitzung	Montag, 27.März 2006, 10.00 - 11.00 Uhr, Schindler Informatik AG, Konferenzraum "Rigi".

8.3.3 Sitzung 3: Besprechung

Nr	3
Datum / Zeit	Montag, 20.März 2006, 10.00 - 10.30 Uhr.
Ort	Schindler Informatik AG, Konferenzraum „Rigi“.
Anwesende	? Michael Jenny ? Patrick Gehrig ? Thomas Galliker
Gesprächsinhalt	? Einleitung Nach einer kurzen Begrüssung zeigte ich den Anwesenden verschiedene Ansichten des erstellten Visio-Diagramms und kommentierte diese kurz. Schliesslich forderte ich den Kunden auf, mir genaue Informationen über die verschiedenen Anwendungsfälle der in diesem Projekt zu entwickelnder Anwendung („Setup Assistant“). ? Definieren der Setup Modi Der Kunde erläuterte einige „Probleme“, welche er mit dieser Anwendung gelöst haben möchte. Ich machte mir jeweils Gedanken über die technische Machbarkeit dieser detaillierten Anforderungen. Zudem musste ich jeweils abschätzen ob die Wünsche den Zeitrahmen des Projekts überziehen würden. Nach einem anspruchsvollen Gespräch konnten wir sämtliche Anwendungsfälle in Form eines Diagramms aufzeichnen. (Das bestehende Diagramm wird zu einem späteren Zeitpunkt aktualisiert). ? Gestaltung der Benutzerschnittstelle nach Schindler CI Zum Schluss erklärte ich dem Kunden, dass ich die Benutzerschnittstelle nach dem neuen Corporate Identity (CI) - Standard von Schindler gestalten werde, sofern er keine Einwände dagegen hat. Der Kunde begrüsst diesen Vorschlag.
Entscheidungen	? Gestaltung der Benutzerschnittstelle nach Schindler CI ? Definierte Setup Modi: 1.NewInstallation, 2.ReInstallation, 3.Clean Up
Nächste Sitzung	Montag, 27.März 2006, 10.00 - 11.00 Uhr, Schindler Informatik AG, Konferenzraum "Rigi".

Sitzung 4: Projektübergabe

Nr	4
Datum / Zeit	Donnerstag, 30.März 2006, 10.00 - 11.30 Uhr.
Ort	Schindler Informatik AG, Konferenzraum „Rigi“.
Anwesende	? Stefan Epp ? Michael Jenny ? Patrick Gehrig ? Thomas Galliker
Gesprächsinhalt	? Einführung in Setup Assistant Zu Beginn wurde dem Kunden eine kurze Einführung in Setup Assistant gegeben. Sämtliche Möglichkeiten und Einstellungen der Benutzeroberfläche wurden vorgeführt und erklärt. ? Akzeptanztest durch den Auftraggeber Im Anschluss an die Einführung konnte der Auftraggeber testen, ob das Produkt die Anforderungen erfüllt. Das Formular des Akzeptanztests wurde ausgedruckt und durch den Kunden ausgefüllt und visiert.

8.4 Problemprotokoll

Nachfolgende Tabellen enthalten einige Problemfälle, dessen Lösungen nur durch Fremdhilfe und/oder intensiver Lösungssuche gefunden werden konnten.

8.4.1 Problem 1 – HTA Interface Refresh Problem

Nr	1
Problem	MS HTA Interface Refresh-Problem; Benutzeroberfläche wird beim Ausführen umfangreicher Funktionen längere Zeit blockiert und nicht refreshed.
Lösungsvorgehen	Internet Recherchen (Google, MSDN, ...)
Lösung	Microsoft bezeichnet das HTA Interface Refresh Problem nicht als Fehler. Es handelt sich um eine konzeptionelle Eigenschaft von HTA. Es gibt drei Möglichkeiten, wie das Interface erneuert werden kann: 1. durch die Ausgabe einer „MsgBox“ 2. durch das Setzen der Explorer-Statuszeile „window.status“ 3. durch das Initiieren eines Timeouts „SetTimeout()“.
Fremdhilfe	[Quelle 2.17]

8.4.2 Problem 2 – Mking.cmd Problem

Nr	2
Problem	Prozess zum Erstellen von Windows PE via "Mking.cmd" bricht ab.
Lösungsvorgehen	Debuggen von "Mking.cmd".
Lösung	Die verwendete Windows 2003 Server Dateien waren nicht von einer OEM-CD, sondern von einer firmeninternen Netzwerkfreigabe. Es fehlten einige Ordner in diesem Verzeichnis, was "Mking.cmd" dazu brachte, die Betriebssystem Quelle als ungültig zu deklarieren.
Fremdhilfe	-

8.4.3 Problem 3 - Partitionierungsproblem

Nr	3
Problem	Schwer nachvollziehbares, rekursives Verhalten der Funktion „Part()“ der Anwendung „Setup Assistant“.
Lösungsvorgehen	Debuggen des VBScript Codes, vereinfachen des Codes zur Findung der Fehlerursache.
Lösung	Verursacher des Problems war die Timeout-Funktion „SetTimeout()“, welche ich als Workaround für das HTA Interface Refresh Problem eingesetzt hatte.
Fremdhilfe	Patrick Gehrig (Fachausbilder)

8.4.4 Problem 4 – Fehlender NIC-Treiber

Nr	4
Problem	Keine Netzwerk-Konnektivität in Windows PE.
Lösungsvorgehen	IP Konfiguration mit dem Befehl „ipconfig“ prüfen. Fehlermeldung: „An internal error occurred: The request is not supported. Please contact Microsoft Product Support Services for further help. Additional information: Unable to query host name.“
Lösung	Siehe Netzwerktreiber integrieren .
Fremdhilfe	OPK Hilfe, siehe [Quelle 1.05] .

8.4.5 Problem 5 – CD-RW Problem

Nr	5
Problem	Bluescreen beim Start von Windows PE nach erfolgter Treiberintegration.
Lösungsvorgehen	ISO Image nochmals erstellen; CD nochmals brennen.
Lösung	Windows PE hatte scheinbar Probleme beim Lesen des verwendeten CD-RW Mediums. Nach dem Brennen auf einen leeren CD-Rohling konnte Windows PE fehlerfrei gestartet werden.
Fremdhilfe	-

8.4.6 Problem 6 – Unity Asset Informationen

Nr	6
Problem	Unity Upgrade Assistant läuft nicht automatisch durch.

Lösungsvorgehen	Debuggen von Setup Assistant ↗ Ausgabe der AssetFile Pfadinformationen. Unity Server ↗ Überprüfung der Asset Konfigurationsdateien.
Lösung	Setup Assistant speicherte die AssetFile-Pfadinformationen in eine falsche Variable. Dabei wurde beim Neuerstellen der Asset Informationen auf dem Server jeweils eine Asset-Datei namens ".INI" erstellt. Auffallend war auch, dass plötzlich sämtliche Testsysteme dieselben Informationen anzeigen in Setup Assistant (dies, weil sie alle dieselbe Konfiguration ".INI" benutzten).
Fremdhilfe	-

8.4.7 Problem 7 – Latenzzeiten

Nr	7
Problem	HTA Login hat teilweise sehr hohe Latenzzeiten, was sich negativ auf die Ergonomie der Software auswirkt.
Lösungsvorgehen	-
Lösung	Das verzögerte Verhalten von HTA ist auf die langsamen Zugriffszeiten der eingesetzten CD/DVD-Medien zurückzuführen. Auf die Lösung dieses Latenzproblems wird in Absprache mit dem Fachausbildner verzichtet, da diese konzeptionelle Änderungen am Projekt verlangen würde.
Fremdhilfe	Patrick Gehrig (Fachausbildner)

9 Glossar

ADO - ActiveX Data Objects

ActiveX Schnittstelle zur Abfrage von ADO-Datenquellen.

Weitere Informationen siehe [\[Quelle 2.35\]](#).

Asset

Als Asset wird unter Unity ([↗ Unity](#)) ein Client Computer bezeichnet. Die Konfiguration eines Client Computers befindet sich in einem so genannten „Asset File“ (auch „Asset Configuration“ genannt). Diese Konfiguration wird durch den in der Softwareverteilungsumgebung eindeutige „Asset Tag“ ([↗ Asset Tag](#)) identifiziert.

Asset Tag

Eindeutige Kennzeichnung einer Client Konfiguration ([↗ Asset](#)). Die Kennzeichnung wird aus der Hardware des entsprechenden Client Computers ausgelesen und setzt sich wie folgt zusammen: Hersteller~Typ~Modell~Seriennummer. Beispiel: IBM~2668F7G~THINKPAD T43~L3VYDLH.

Debugging

Methode zur Lokalisierung von Fehlern in Programmen und Scripts.

Weitere Informationen siehe [\[Quelle 2.36\]](#).

HTA - Hypertext Application

HTML Applikationen (kurz HTA) bieten eine gute Möglichkeit, grafische Benutzerschnittstellen (GUI) für Scripts zu kreieren. Hypertext Applikationen sind sehr stark verwandt mit den herkömmlichen HTML Dateien.

Weitere Informationen siehe [\[Quelle 2.21\]](#) und [\[Quelle 2.22\]](#).

Integrationstest

Teststufe, bei welcher die Zusammenarbeit (Interaktion) zwischen mehreren Komponenten eines Gesamtsystems geprüft wird.

Weitere Informationen siehe [\[Quelle 2.19\]](#).

Komponententest

Teststufe, bei welcher eine einzelne Komponente eines Gesamtsystems geprüft wird. Auch genannt Modul-Test oder Unit-Test.

Weitere Informationen siehe [\[Quelle 2.20\]](#).

OEM – Original Equipment Manufacturer

Ein OEM bezeichnet einen Hersteller, welcher Produkte anderer Firmen einkauft, unverändert in eigene Produkte integriert und anschliessend als Gesamtprodukt an Endkunden weiterverkauft.

Der Begriff "OEM" existiert jedoch nicht nur in der Informatikbranche. Auch in der Automobil- und Elektronikbranche hat der Begriff weite Verbreitung.

Weitere Informationen siehe [\[Quelle 2.30\]](#).

OPK – OEM Preinstallation Kit

Ein OEM bezeichnet einen Hersteller, welcher Produkte anderer Firmen einkauft, unverändert in eigene Produkte integriert und anschliessend als Gesamtprodukt an Endkunden weiterverkauft.

Der Begriff "OEM" existiert jedoch nicht nur in der Informatikbranche. Auch in der Automobil- und Elektronikbranche hat der Begriff weite Verbreitung.

Hilfe und weitere Informationen sind zu finden unter [Anhang F](#).

PE – Preinstallation Environment

Ein Preinstallation Environment (PE) ist ein leichtgewicht Windows Betriebssystem, welches aus den Quelldateien von Windows XP SP1, SP2 oder Windows 2003 Server SP1 erstellt werden kann. Preinstallation Environments sind keine gewöhnliche Betriebssysteme, und können nicht direkt beim Softwarehändler gekauft werden. Es handelt sich um ein, mit entsprechender Software hergestelltes, auf eigene Bedürfnisse zugeschnittenes "Miniatur-Windows".

Diese im Funktionsumfang stark eingeschränkten Preinstallation Environments wurden hauptsächlich für OEM Hersteller und Systemadministratoren konzipiert.

Dank dem geringen Speicherbedarf und den modularen Erweiterungsmöglichkeiten bieten Preinstallation Environments einen grossen Anwendungsbereich, welcher sich von der systematischen Verteilung von Microsoft Betriebssystemen bis hin zu Diagnose- und Recovery-Zwecken erstreckt.

Weitere Informationen siehe [\[Quelle 2.12\]](#), [\[Quelle 2.13\]](#), [\[Quelle 2.14\]](#).

Prototyping-Verfahren

Bezeichnet eine Entwicklungsmethode, in der während der Entwicklungsphase stetig Komponenten-, Integrations- und Systemtests durchgeführt werden.

Weitere Informationen siehe [\[Quelle 2.29\]](#).

RAMDisk

Eine RAMDisk (auch "RAMDrive" genannt) ist ein virtueller Datenträger im Arbeitsspeicher eines Computers. RAMDisks reservieren im Arbeitsspeicher ("Random Access Memory", kurz "RAM") den konfigurierten Speicherplatzbedarf und stellen diesen in Form eines virtuellen Laufwerks unter Windows zur Benutzung zur Verfügung. Daten einer RAMDisk werden vollumfänglich im RAM gehalten, bis das System heruntergefahren wird.

RAMDisks zeichnen sich durch hohe Lese- und Schreibgeschwindigkeiten sowie einem hohen Flexibilitätsgrad aus. Als Nachteil (oder besser: Nebeneffekt) kann der Verlust aller in der RAMDisk gespeicherten Daten betrachtet werden, wenn das System ausgeschaltet wird.

Einige Anwendungsbeispiele:

- ? *Cache-Laufwerk für Applikationen (z.B. temporäre Internetdateien).*
- ? *Lese-/Schreibbares Laufwerk in Preinstallation Environments.*
- ? *Audio-/Video Bearbeitung.*

Weitere Informationen siehe [\[Quelle 2.27\]](#).

Registry

Die Registrierung (engl. "Registry") stellt eine zentrale Informationssammlung von Windows dar. In dieser hierarchisch aufgebauten Datenbank können Informationen, sowohl vom Betriebssystem, als auch von Programmen, gespeichert und gelesen werden. Zur Darstellung der Informationen stellt Microsoft den Registrierungseditor "Regedit" zur Verfügung.

Weitere Informationen siehe [\[Quelle 2.25\]](#).

Slipstreaming

Methode zur Aktualisierung einer Betriebssystemquelle. Durch diese Methode können Updates (Hotfixes, Patches oder Service Packs) in eine Windows 2000/XP Betriebssystemquelle eingebunden werden, ohne dass diese nach der Installation von Windows separat installiert werden müssen.

Weitere Informationen siehe [\[Quelle 2.26\]](#).

Systemtest

Teststufe, bei welcher das System als Ganzes geprüft wird. Diese Stufe wird erreicht, sobald sämtliche Vorstufen (Komponententest, Integrationstest,...) positiv verlaufen sind.

Weitere Informationen siehe [\[Quelle 2.28\]](#).

Unity

Softwareverteilungs- und Management-Software. Die Software bietet umfangreiche Möglichkeiten zur Verteilung und Verwaltung von Windows Installer Paketen. Im Wesentlichen gehören folgende Komponenten zu Unity:

- ? Unity Package Manager; Software zur Verwaltung von Windows Installer Paketen.
- ? Unity Deployment Manager; Software zur Verwaltung von Konfigurationen, operative Schnittstelle der Softwareverteilung.
- ? Unity Closed Client; Agent zur Steuerung von Unity Clients.

Weitere Informationen siehe [\[Quelle 1.08\]](#).

VBS - VisualBasicScript

Visual Basic Script (kurz "VBS" oder "VBScript") ist eine von Microsoft entwickelte, in WSH eingebundene objektorientierte Scriptsprache. Visual Basic Script hat syntaktische Ähnlichkeiten zu Visual Basic (VB) und Visual Basic Application (VBA).

Weitere Informationen siehe [\[Quelle 2.31\]](#).

VGA - Video Graphics Array

Als VGA wird ein Video Adapter bezeichnet, welcher eine Bildauflösung von mindestens 640 x 480 Pixel hat. VGA ist der Video Display Standard für Hardware während des Startvorgangs von Windows Betriebssystemen.

Weitere Informationen siehe [\[Quelle 2.32\]](#).

WMI - Windows Management Instrumentation

Über WMI können Hardware-Informationen eines Windows Computers ausgelesen werden. WMI ist integraler Bestandteil von Windows 2000, XP und 2003. Für Windows 9x und NT4 gibt es entsprechende Updates.

Weitere Informationen siehe [\[Quelle 2.33\]](#).

WSH - Windows Script Host

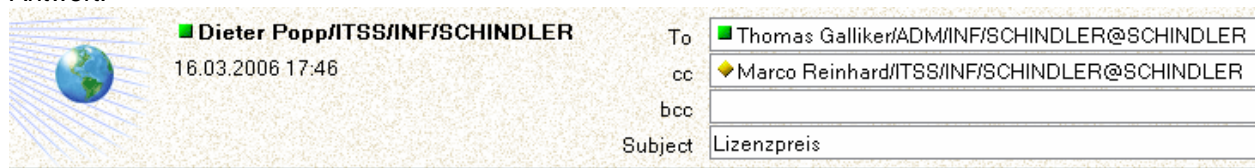
Windows Script Host (oftmals auch "Windows Scripting Host" genannt) bietet unter Windows eine ActiveX Schnittstelle für Scriptsprachen. WSH kann als direkte Ablösung der Batch-Programmierung angesehen werden. Standardmässig unterstützte Scriptsprachen sind JScript und VBScript (~~VS~~ *VBScript* - *VisualBasicScript*). Weitere Scriptsprachen lassen sich nachinstallieren.

Weitere Informationen siehe [\[Quelle 2.34\]](#).

10 Anhang

10.1 Anhang A – Software Assurance Lizenzkosten

Auf die Anfrage der Software Assurance (SA) Lizenzkosten für Windows Server 2003 erhielt ich folgende Antwort:



Hoi Thomas,

ein Windows 2003 Server Lizenz mit SA kostet Fr. 1582.- ohne Mwst.

Gruss
Dieter

Abbildung 17: Mail von D.Popp: Windows Server 2003 SA Lizenzkosten.

10.2 Anhang B – Quellcode

10.2.1 Login Form

10.2.1.1 Login.hta

```
<html>
<head>
<meta http-equiv="Content-Language" content="de-ch">

<title>HTA Login Script Application</title>

<HTA:APPLICATION
    ID="objHTALogin"
    APPLICATIONNAME="HTA Login Script Application"
    SCROLL="no"
    SINGLEINSTANCE="no"
    WINDOWSTATE="maximize"
    BORDER="none"
    innerBorder="none"
    maximizeButton="false"
    minimizeButton="false"
>

<SCRIPT>
var shutdownKey=1

function checkKey()
{
    //alert(window.event.keyCode)

    //Press ENTER to login
    if (window.event.keyCode == 13)
    {
        Login()
    }
    if (window.event.keyCode == 27)
    {
        if (shutdownKey == 2)
        {
            shutdownKey=1
            Shutdown()
        }
        else
        {
            shutdownKey = shutdownKey+1
        }
    }
}
</SCRIPT>

</head>

<SCRIPT Language="VBScript">

Dim ScriptRoot, ConfigFile
Dim iniTitle, BackgroundColor
Dim AuthenticationMethod, AuthorizedGroups
Dim LDAPProvider, LDAPPath, GlobalCatalog
Dim logonUsername, logonDomain, logonPassword
Dim Domain, DomainController
Dim RamDisk
Dim SourceShare
Dim SourceShareSetup
Dim System32Dir

Sub Window_Onload
    document.body.style.cursor = "wait"
    GetVariables()
    AuthorizedGroups=GetAuthorizedGroups()
    Window.Focus()
    Call SetFocus()
    document.body.style.cursor = "default"
End Sub

Sub Reload
    Location.Reload(True)
```

```

End Sub

Function GetVariables
    Dim objFSO, WshShell
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    ScriptRoot = objFSO.GetAbsolutePathName(".")
    ConfigFile = ScriptRoot & "\config.ini"

    Set WshShell = CreateObject("WScript.Shell")
    System32Dir = WshShell.ExpandEnvironmentStrings("%SystemRoot%\system32")

    iniTitle = GetINIStr("General", "Title", "Error in script!", ConfigFile)
    BackgroundColor = GetINIStr("General", "BackgroundColor", "#004E98", ConfigFile)

    RamDisk = GetINIStr("RAMDisk", "RAMDiskLetter", "Z:", ConfigFile)

    GlobalCatalog = GetINIStr("LDAP", "GlobalCatalog", "", ConfigFile)
    AuthenticationMethod = Ucase(GetINIStr("Authentication", "AuthenticationMethod", "",
ConfigFile))
    DefaultDomain = Ucase(GetINIStr("Authentication", "DefaultDomain", "", ConfigFile))
    LDAPPath = GetLDAPPath(Domain)
End Function

Function SetFocus
    'Enable input fields
    Form1.txtUsername.disabled = False
    Form1.txtPassword.disabled = False

    If Form1.txtUsername.value="" Then
        Form1.txtUsername.Focus()
    Else
        Form1.txtPassword.Focus()
    End If
End Function

Function GetLDAPPath(Domainname)
    Dim tmp, returnvalue
    tmp = Split(Domainname, ".")
    For i=0 To Ubound(tmp)
        If i<Ubound(tmp) Then
            returnvalue = returnvalue & "DC=" & tmp(i) & ","
        Else
            returnvalue = returnvalue & "DC=" & tmp(i)
        End If
    Next
    GetLDAPPath = returnvalue
End Function

Function GetDomains
    Dim fso, f, line, linevalue, Count, OptionEntry, OptionEntries
    Const Reading=1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.OpenTextFile(ConfigFile, Reading)

    Count=0
    Do Until f.AtEndOfStream
        line = f.ReadLine
        If InStr(line, "Domain" & (Count+1)) Then
            Count = Count+1
        End If
    Loop

    If Count=0 Then
        'msgbox "No domain configured in config.ini!"
        txtInfo.innerHTML = "No domain configured. Check config.ini!"
    ElseIf Count=1 Then
        txtMasterDomain.InnerHTML = GetINIStr("LDAP", "Domain1", "error", ConfigFile)
        &".& GlobalCatalog
    ElseIf Count>1 Then
        For i=1 To Count
            OptionName = GetINIStr("LDAP", "Domain"&i, "error", ConfigFile)
            &".& GlobalCatalog
            OptionValue = GetINIStr("LDAP", "Domain"&i & "DC", "error",
ConfigFile)
            OptionEntries = OptionEntries & "<option
value='&i&'>&OptionName&</option>"
        Next
        txtMasterDomain.InnerHTML = "<SELECT size='1' class='optionList'
name='comboDomain'
onChange='SetDomainValue(Form1.comboDomain.value)'>&OptionEntries&</SELECT>"
        Call SelectDefaultDomain()

```



```

        End If
    End Function

Function SelectDefaultDomain
    Dim DefaultDomainNr : DefaultDomainNr = GetINIString("Authentication", "DefaultDomain",
"error", ConfigFile)
    Form1.comboDomain.Value = DefaultDomainNr
    SetDomainValue(Form1.comboDomain.value)
End Function

Function SetDomainValue(DomainNr)
    Domain = GetINIString("LDAP", "Domain"&DomainNr, "error", ConfigFile)
    DomainController = GetINIString("LDAP", "Domain"&DomainNr&"DC", "error", ConfigFile)
    SourceShare = GetINIString("LDAP", "Domain"&DomainNr&"SourceShare", "error", ConfigFile)
    SourceShareSetup = SourceShare &"\"& GetINIString("LDAP", "Domain"&DomainNr&"htaSetup",
"error", ConfigFile)
End Function

Function GetDomainValue(DomainNr)
    GetDomainValue = GetINIString("LDAP", "Domain"&DomainNr, "error", ConfigFile)
End Function

Function GetAuthorizedGroups
    Dim GroupString, GroupArray
    GroupString = GetINIString("Authentication", "AuthorizedGroups", "", ConfigFile)
    GroupArray = Split(GroupString, ";")

    'For each group in GroupArray
    ' msgbox group
    'Next
    GetAuthorizedGroups=GroupArray
End Function

Function GetINIString(Section, KeyName, Default, FileName)
    Dim INIContents, PosSection, PosEndSection, sContents, Value, Found

    'Get contents of the INI file As a string
    INIContents = GetFile(FileName)

    'Find section
    PosSection = InStr(1, INIContents, "[" & Section & "]", vbTextCompare)
    If PosSection>0 Then
        'Section exists. Find end of section
        PosEndSection = InStr(PosSection, INIContents, vbCrLf & "[")
        '?Is this last section?
        If PosEndSection = 0 Then PosEndSection = Len(INIContents)+1

        'Separate section contents
        sContents = Mid(INIContents, PosSection, PosEndSection - PosSection)

        If InStr(1, sContents, vbCrLf & KeyName & "=", vbTextCompare)>0 Then
            Found = True
            'Separate value of a key.
            Value = SeparateField(sContents, vbCrLf & KeyName & "=", vbCrLf)
        End If
    End If
    If isempty(Found) Then Value = Default
    GetINIString = Value
End Function

'Separates one field between sStart And sEnd
Function SeparateField(ByVal sFrom, ByVal sStart, ByVal sEnd)
    Dim PosB: PosB = InStr(1, sFrom, sStart, 1)
    If PosB > 0 Then
        PosB = PosB + Len(sStart)
        Dim PosE: PosE = InStr(PosB, sFrom, sEnd, 1)
        If PosE = 0 Then PosE = InStr(PosB, sFrom, vbCrLf, 1)
        If PosE = 0 Then PosE = Len(sFrom) + 1
        SeparateField = Mid(sFrom, PosB, PosE - PosB)
    End If
End Function

'File functions
Function GetFile(ByVal FileName)
    Dim FS: Set FS = CreateObject("Scripting.FileSystemObject")
    'Go To windows folder If full path Not specified.
    If InStr(FileName, ":\") = 0 And Left (FileName,2)<>"\\\" Then
        FileName = FS.GetSpecialFolder(0) & "\" & FileName
    End If
    On Error Resume Next

```

```

    GetFile = FS.OpenTextFile(FileName).ReadAll
End Function

Function MoveAnswerFile(AnswerFileSource)
    Dim AnswerFileDest, objFileCopy, objNewFile, objFSO, WshShell, WshSysEnv

    Set WshShell = CreateObject("WScript.Shell")

    AnswerFileDest = WshShell.ExpandEnvironmentStrings("%TEMP%\unattend.txt")

    'copying file from cd-rom to TEMP folder
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    Set objFileCopy = objFSO.GetFile(AnswerFileSource)
    objFileCopy.Copy (AnswerFileDest)

    'setting file attribute to "normal"
    Set objNewFile = objFSO.GetFile(AnswerFileDest)
    objNewFile.Attributes = 0

    MoveAnswerFile = AnswerFileDest
End Function

Function Reboot()
    Dim WshShell, CmdDir, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    CmdDir = Left(WshShell.ExpandEnvironmentStrings("%COMSPEC%"),
InStrRev(WshShell.ExpandEnvironmentStrings("%COMSPEC%"), " \"))
    Cmd = CmdDir & "reboot.cmd"

    'msgbox Cmd
    WshShell.Run(Cmd), 1, True
End Function

Function Login()
txtInfo.innerHTML = "Connecting..."

    If InStr(Form1.txtUsername.value, "@") <> 0 Then
        ' if the username is in correct form "domain\username":
        tmpDomain = Mid(Form1.txtUsername.value, InStr(Form1.txtUsername.value, "@")+1)
        If InStr(tmpDomain, GlobalCatalog) = 0 Then
            tmpDomain = tmpDomain & "." & GlobalCatalog
        End If
        Username = tmpDomain & "\" &
Left(Form1.txtUsername.value, InStr(Form1.txtUsername.value, "@")-1)
    Else
        ' if the username is not in correct form (e.g. username only):
        Username = GetDomainValue(Form1.comboDomain.value) & "." & GlobalCatalog & "\" &
Form1.txtUsername.value
    End If

    Password = Form1.txtPassword.value

    'Disable input fields
    Form1.txtUsername.disabled = True
    Form1.txtPassword.disabled = True

    If Username="" OR Password="" Then
        txtInfo.innerHTML = "Authentication error.<br>Check username and password."
        SetFocus()
        Exit Function
    End If

    'Refresh HTA Interface -----
    window.status = ""
    '-----

Select Case AuthenticationMethod

Case "NATIVE":
'=====
    'Native Mode does not authenticate username/password against active directory
    'The user is authenticated if it is permitted to connect the share

    Call SetupInitialization(Username, Password)

Case "LDAP":
'=====
    Const ADS_SECURE_AUTHENTICATION = 1
    Dim rs, openDS, objOU, LDAPPath, objUser, DomainCN, UserCN, objMemberOf, IsAuthorized

```

```

Dim equalsIndex, commaIndex, Groupname

DomainCN = "dc=inf,dc=schindler,dc=com"
LDAPPath = "LDAP://" & Domain & "/cn=users," & DomainCN
UserCN = "LDAP://CN=" & Username & ",OU=Users,OU=Prod," & DomainCN

Set oConn = CreateObject("ADODB.Connection")
Set oComm = CreateObject("ADODB.Command")

oConn.Provider = "AdsDSOObject"

' Set the binding options for the search.
oConn.Properties("ADSI Flag") = ADS_SECURE_AUTHENTICATION

If Username <> vbNullString AND Username <> "" Then
    oConn.Properties("User ID") = Username
    oConn.Properties("Password") = Password
    oConn.Properties("Encrypt Password") = True
End If

'On Error Resume Next
oConn.Open
oComm.ActiveConnection = oConn

'Format search criteria using SQL syntax
'commandString = "SELECT memberOf, samAccountName, givenName, sn, AdsPath FROM
'LDAP://" & DomainController & "/OU=Users,OU=Prod,DC=inf,DC=schindler,DC=com" WHERE samAccountName =
' " & Username & "'
commandString = "SELECT memberof FROM 'GC://schindler.com' WHERE
sAMAccountname=' " & Username & "'

'msgbox commandstring
oComm.CommandText = commandString

' Execute the query.
Set rs = oComm.Execute
objMemberOf=rs("memberof")

IsAuthorized=false
if err.number <> 0 then
    'msgbox "Authentication error (code "&err.number&"). Please check username and
password."
    txtInfo.innerHTML = "Authentication error (code "&err.number&").<br>Check
username and password."

    Form1.txtPassword.value = ""
    err.clear
    Exit Function
Else
    For Each objGroup in objMemberOf
        If IsAuthorized=false Then
            equalsIndex = InStr(objGroup,"=")
            commaIndex = InStr(objGroup,",")
            Groupname = Mid(objGroup,(equalsIndex)+1,(commaIndex-equalsIndex)-
1)

            'msgbox Groupname
            For Each iniGroupEntry in AuthorizedGroups
                If Groupname=iniGroupEntry Then
                    IsAuthorized=true
                End If
            Next
        End If
    Next

    If IsAuthorized=true Then
        err.clear
        Call SetupInitialization(Username,Password)
    Else
        'msgbox "Authorization error (code "&err.number&"). "&Username&" is not
authorized."
        txtInfo.innerHTML = "Authorization error (code
"&err.number&").<br>"&Username&" is not authorized."

        Form1.txtPassword.value = ""
        Exit Function
    End If
End If
End Select
End Function

```

```

Function SetupInitialization(iUsername,iPassword)
    Dim returnvalue

    logonUsername = iUsername
    logonDomain = iDomain
    logonPassword = iPassword

    If IsNetworkAvailable() Then
        returnvalue = ConnectShare(iUsername,iPassword)
        If returnvalue = 0 Then
            Form1.txtUsername.value = ""
            Form1.txtPassword.value = ""
            Call CopyFiles(SourceShareSetup,RamDisk&"\htaSetup")
            txtInfo.innerHTML = iUsername&" is authenticated successfully"
            Call RunSetupGUI(RamDisk&"\htaSetup")
        Else
            txtInfo.innerHTML = iUsername&" is not authorized."
            Form1.txtPassword.value = ""
        End If
    Else
        txtInfo.innerHTML = "Connection Error. "&vblf&" Please check IP connectivity and
NIC drivers."
    End If
    SetFocus()
End Function

Function GetIPAddress()
    Dim value : value=""
    Set objWMIService = GetObject("winmgmts:\\.\\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM
Win32_NetworkAdapterConfiguration",,48)

    On Error Resume Next
    For Each objItem in colItems
        If objItem.Index = 1 Then
            value = Join(objItem.IPAddress, ",")
        End If
    Next
    On Error Goto 0

    GetIPAddress=value
End Function

Function IsNetworkAvailable()
    Dim returnvalue : returnvalue=False
    If GetIPAddress()<>" Then
        'msgbox "IP Address is available"
        returnvalue=True
    Else
        'msgbox "No IP Address available"
        returnvalue=False
    End If
    IsNetworkAvailable = returnvalue
End Function

Function ConnectShare(iUsername,iPassword)
    Dim WshShell, Cmd, returnvalue
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir&"\Net.exe use "&SourceShareSetup&" /user:"&iUsername&" "&iPassword
    'msgbox Cmd

    returnvalue = WshShell.Run(Cmd, 0, true)
    ConnectShare = returnvalue
End Function

Function CopyFiles(Source, Destination)
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = "xcopy "&Source&"\* "&Destination&" /I /S /Y"
    'msgbox Cmd

    Call WshShell.Run(Cmd,0,True)
End Function

Function RunSetupGUI(Path)
    Dim WshShell, Cmd, Args
    Dim errorcode
    Set WshShell = CreateObject("WScript.Shell")

```

```

        txtInfo.innerHTML = "Please wait..."
        'msgbox domain
        Args = "/arg:" & Domain & " /arg:" & SourceShare
        Cmd = System32Dir & "\mshta.exe " & Path & "\Setup.hta " & Args
        'msgbox Cmd

        Call WshShell.Run(Cmd,1,False)
        WshShell.AppActivate("HTA Login Script Application")
        Window.Close()
    End Function

Function Shutdown()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir & "\ShutdownPE.exe shutdown"
    'msgbox Cmd

    Call WshShell.Run(Cmd,0,False)
End Function

Function Reboot()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir & "\ShutdownPE.exe restart"
    'msgbox Cmd

    Call WshShell.Run(Cmd,0,False)
End Function

</SCRIPT>
<link rel="stylesheet" type="text/css" href="gfx/stylesheet.css">
<form method="POST" name="Form1" action="--WEBBOT-SELF--">
<body style="background-color: #004E98" onkeypress='checkKey()''>
    <script type="text/vbscript">
        Call GetVariables()
        document.body.style.cssText="background -
color: "&BackgroundColor
    </script>

<table border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="100%" id="AutoNumber9" height="100%">
    <tr>
        <td width="33%" height="100%">&nbsp;</td>
        <td width="33%" height="100%">
            <div style="position:relative; top:0px; left:0px; z-index:1;
width:374; height:270">
                <b>
                    </b>

                    <div id="Title" class="singleLineHeadlineNormal"
style="position:absolute; top:20px; left:120px; width:275px; height:37px; z-index:2">
                        <p align="left">
                    </div>
                    <div id="Logo" style="position:absolute; top:20px;
left:20px; width:57px; height:33px; z-index:2">
                        
                    </div>

                    <div id="txtInfo" class="mainContent"
style="position:absolute; top:63px; left:120px; width:275px; height:33px; z-index:2">
                    </div>

                    <div style="position:absolute; top:130; left:116; width:182;
height:20; z-index:2">
                        <p align="left">
                            <input type="text" class="inputField" name="txtUsername"
size="37" style="width:200px;">
                        </div>

                    <div class="mainContent" style="position:absolute; top:130;
left:9; width:76; height:20; z-index:2">
                        <p align="left">
                            User Name:
                        </div>

```

```

height:20; z-index:2">
        <div style="position:absolute; top:155; left:116; width:189;
            <p align="center">
                <input type="password" class="inputField"
name="txtPassword" style="width:200px;">
            </p>
        </div>

        <div class="mainContent" style="position:absolute; top:155;
left:9; width:76; height:20; z-index:2">
            <p align="left">
                Password:
            </p>
        </div>

height:20; z-index:2">
        <div style="position:absolute; top:180px; left:116px; width:150;
            <p align="center">
                <div id="txtMasterDomain">&nbsp;</div>
            </p>
        </div>

        <div class="mainContent" style="position:abso lute; top:180;
left:9; width:83; height:20; z-index:2">
            <p align="left">
                Unity Master:
            </p>
        </div>

height:24; z-index:2">
        <div style="position:absolute; top:215; left:116; width:24;
            <p align="left">
                <a href="#" style="text-decoration: none"
onClick="Login()">
                <button name="btnLogin" class="formButton" style="width:
70;">OK</button>
            </a>
        </div>

        <div style="position:absolute; top:215; left:206; width:43;
            <p align="left">
                <a href="#" style="text-decoration: none"
onClick="Shutdown()">
                <button name="btnShutdown" class="formButton" style="width:
70;">Shutdown</button>
            </a>
        </div>
    </div>

    <script type="text/vbscript">
        Call GetVariables()
        Call GetDomains()
        Form1.txtUsername.value = GetINIStrng("Authentication",
"DefaultUsername", "", ConfigFile)
        Title.innerHTML = GetINIStrng("General", "Title", "error",
ConfigFile)
    </script>

</td>
<td width="34%" height="100%">&nbsp;</td>
</tr>
</table>
</body>
</form>
</html>

```

10.2.1.2 Config.ini

```

[General]
Title=Login Form
BackgroundColor=#004E98

[Authentication]
DefaultUsername=
DefaultDomain=1
AuthenticationMethod=NATIVE

[LDAP]
GlobalCatalog=schindler.com

Domain1=inf
Domain1DC=infw0001

```

```
Domain1SourceShare=\\infw0226\ui_prod$  
Domain1htaSetup=Tools\htaSetup
```

10.2.2 Factory Components

10.2.2.1 Startnet.cmd

```
@echo off
COLOR 1F
ECHO Please wait...

REM - INSTALL WSH COMPONENT //////////////////////////////////////
regsvr32 %SystemRoot%\System32\jscript.dll /S
regsvr32 %SystemRoot%\System32\scroobj.dll /S
regsvr32 %SystemRoot%\System32\scrrun.dll /S
regsvr32 %SystemRoot%\System32\vbscript.dll /S
regsvr32 %SystemRoot%\System32\shext.dll /S
regsvr32 %SystemRoot%\System32\shom.ocx /S
regsvr32 %SystemRoot%\System32\mlang.dll /S
%SystemRoot%\System32\rundll32.exe setupapi,InstallHinfSection DefaultInstall 132 WSH.inf

REM - INSTALL ADO COMPONENTS //////////////////////////////////////
regsvr32 %SystemRoot%\System32\MSCTF.dll /s
regsvr32 %SystemRoot%\System32\CLBCATQ.DLL /s
regsvr32 %SystemRoot%\System32\comsvcs.dll /s
regsvr32 %SystemRoot%\System32\colbact.DLL /s
regsvr32 %SystemRoot%\System32\dssenh.dll /s
regsvr32 %SystemDrive%\Program Files\Common Files\System\Ole db\oledb32.dll" /S
regsvr32 %SystemDrive%\Program Files\Common Files\System\Ole db\OLEDB32R.DLL" /S
regsvr32 %SystemDrive%\Program Files\Common Files\System\Ole db\sqloledb.dll" /S
regsvr32 %SystemDrive%\Program Files\Common Files\System\ado\msadhr15.dll" /S
regsvr32 %SystemDrive%\Program Files\Common Files\System\ado\msado15.dll" /S

REM - INSTALL HTA COMPONENTS //////////////////////////////////////
%SystemRoot%\System32\mshta.exe /register
regsvr32 %SystemRoot%\System32\asctrls.ocx /S
regsvr32 %SystemRoot%\System32\plugin.ocx /S
regsvr32 %SystemRoot%\System32\actxprxy.dll /S
regsvr32 %SystemRoot%\System32\atl.dll /S
regsvr32 %SystemRoot%\System32\cryptdlg.dll /S
regsvr32 %SystemRoot%\System32\ddrawex.dll /S
regsvr32 %SystemRoot%\System32\dispex.dll /S
regsvr32 %SystemRoot%\System32\dxtrans.dll /S
regsvr32 %SystemRoot%\System32\hlink.dll /S
regsvr32 %SystemRoot%\System32\iedkcs32.dll /S
regsvr32 %SystemRoot%\System32\iepeers.dll /S
regsvr32 %SystemRoot%\System32\iesetup.dll /S
regsvr32 %SystemRoot%\System32\imgutil.dll /S
regsvr32 %SystemRoot%\System32\inseng.dll /S
regsvr32 %SystemRoot%\System32\itircl.dll /S
regsvr32 %SystemRoot%\System32\itss.dll /S
regsvr32 %SystemRoot%\System32\licmgr10.dll /S
%SystemRoot%\System32\rundll32.exe setupapi,InstallHinfSection reg 132 mshtml.inf
regsvr32 %SystemRoot%\System32\mshtml.dll /S
regsvr32 %SystemRoot%\System32\msrating.dll /S
regsvr32 %SystemRoot%\System32\mstime.dll /S
regsvr32 %SystemRoot%\System32\sendmail.dll /S
%SystemRoot%\System32\rundll32.exe setupapi,InstallHinfSection DefaultInstall 132 HTA.inf

REM - START FACTORY COMPONENTS //////////////////////////////////////
mshta.exe %SystemRoot%\System32\htaFactory\Factory.hta

REM - START LOGIN FORM //////////////////////////////////////
Z:
cd Z:\htaLogin
start Z:\htaLogin\Login.hta

CLS
```

10.2.2.2 Factory.hta

```
<html>
<head>
<meta http-equiv="Content-Language" content="de-ch">
<title>HTA Factory</title>

<HTA:APPLICATION
    ID="objHTASetup"
    APPLICATIONNAME="HTA Factory"
    ICON = ""
    SCROLL="no"
    WINDOWSTATE="maximize"
```



```

BORDER="none"
BORDERSTYLE="none"
INNERBORDER="no"
MAXIMIZEBUTTON="no"
MINIMIZEBUTTON="no"
CAPTION="no"
CONTEXTMENU="no"
NAVIGABLE="yes"
ICON=" "
SCROLL="no" SCROLLFLAT="no"
SELECTION="no"
SHOWINTASKBAR="no"
SINGLEINSTANCE="no"
SYSMENU="yes"
VERSION="1.0"
WINDOWSTATE="normal"
>
<link rel="stylesheet" type="text/css" href="gfx/stylesheet.css">
</head>

<SCRIPT Language="VBScript">

    //////////////////////////////////////
    // GLOBAL VARIABLES
    //////////////////////////////////////
    //////////////////////////////////////
Dim ConfigFile
Dim iniTitle, iniMessage
Dim System32Dir

    //////////////////////////////////////
    // INTERNAL FUNCTIONS
    //////////////////////////////////////
    //////////////////////////////////////
Sub Window_Onload
    On Error Resume Next
    document.body.style.cursor = "wait"

    GetVariables()
    InitPE()
    CopyLoginApp()

    document.body.style.cursor = "default"
    On Error Goto 0
    window.close()
End Sub

    //////////////////////////////////////
Function GetVariables
    Dim objFSO, ScriptFilename, cmdline, WshShell

    cmdline = Replace(objHTASetup.CommandLine, "\"", "")
    cmdline = Left( cmdline, InStr(cmdline, ".hta ") + 4 )
    ScriptRoot = Left(cmdline, InStrRev(cmdline, "\") - 1)
    ConfigFile = ScriptRoot & "\config.ini"

    Set WshShell = CreateObject("WScript.Shell")
    System32Dir = WshShell.ExpandEnvironmentStrings("%SystemRoot%\system32")
End Function

    //////////////////////////////////////
Function InitPE()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir & "\factory.exe -winpe"
    'msgbox Cmd

    Call WshShell.Run(Cmd, 0, True)
End Function

    //////////////////////////////////////
Function CopyLoginApp()
    Dim fso, sourceFolder, destFolder

    sourceFolder = System32Dir & "\htalogin"
    destFolder = "Z:\htalogin"

    Set fso = CreateObject("Scripting.FileSystemObject")
    fso.CopyFolder sourceFolder, destFolder, True

```

```

End Function

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

</SCRIPT>

<body style="background-color: #FFFFFF">
<form method="POST" action="" name="Form1">
<TABLE class=mainTable id="table1" width="916">
  <TBODY>
    <TR>
      <TD class=mainTableCellTopAlign width=20 height=3></TD>
      <TD class=mainTableCellTopAlign width=82></TD>
      <TD class=mainTableCellTopAlign width=15></TD>
      <TD class=mainTableCellTopAlign width=26></TD>
      <TD class=mainTableCellTopAlign width=57></TD>
      <TD class=mainTableCellTopAlign width=247></TD>
      <TD class=mainTableCellTopAlign width=15></TD>
      <TD class=mainTableCellTopAlign width=82></TD>
      <TD class=mainTableCellTopAlign width=15></TD>
      <TD class=mainTableCellTopAlign width=82></TD>
      <TD class=mainTableCellTopAlign width=15 colspan="2"></TD>
      <TD class=mainTableCellTopAlign width=82></TD>
      <TD class=mainTableCellTopAlign width=15></TD>
      <TD class=mainTableCellTopAlign width=82></TD>
      <TD class=mainTableCellTopAlign width=15></TD>
      <TD class=mainTableCellTopAlign width=121></TD>
      <TD class=mainTableCellTopAlign width=23></TD>
      <TD class=mainTableCellTopAlign width=35></TD>
      <TD class=mainTableCellTopAlign width=1></TD>
      <TD class=mainTableCellTopAlign width=-114></TD>
    </TR>
    <TR>
      <TD class=mainTableCellTopAlign></TD>
      <TD class=metaNavigationCell colSpan=10 height=20>&nbsp;</TD>
      <TD class=metaNavigationCell colSpan=10 height=20>&nbsp;</TD>
    </TR>
    <TR>
      <TD class=mainTableCellTopAlign></TD>
      <TD class=mainTableCellTopAlign>
        <IMG height=72 alt=""
          src="gfx/logo-schindler.gif" width=82 border=0></TD>
      <TD class=mainTableCellTopAlign colSpan=2></TD>
      <TD class=mainTableCellBottomAlign colSpan=16>
        <DIV id=Title class=singleLineHeadlineNormal>Welcome to Windows PE 2005</DIV>
        <DIV id=Message class=doubleLineHeadlineSecondLine>Setup Assistant is preparing. Please
be patient...</DIV></TD>
      </TR>
    <TR>
      <TD class=mainTableCellTopAlign colSpan=21 height=9></TD></TR>
    <TR>
      <TD class=mainTableCellTopAlign colSpan=21 height=2></TD></TR>
    <TR>
      <TD class=mainTableCellTopAlign height=14></TD>
      <TD class=mainTableTwinStripes colSpan=20></TD>
    </TR>

    <TD class=mainTableCellTopAlign></TD>
    <TD class=mainTableCellTopAlign></TD></TR>

    <TD class=mainTableCellTopAlign></TD>
    <TD class=mainTableCellTopAlign></TD></TR>
  </TBODY></TABLE>
</form>
</body>
</html>

```

10.2.3 Setup Assistant

10.2.3.1 Setup.hta

```

<html>
<head>
<meta http-equiv="Content-Language" content="de-ch">
<title>HTA Setup</title>

<HTA:APPLICATION
    ID="objHTASetup"
    APPLICATIONNAME="HTA Setup"
    ICON = ""
    SCROLL="no"
    WINDOWSTATE="maximize"
    BORDER="none"
    BORDERSTYLE="none"
    INNERBORDER="no"
    MAXIMIZEBUTTON="no"
    MINIMIZEBUTTON="no"
    CAPTION="no"
    CONTEXTMENU="no"
    NAVIGABLE="yes"
    ICON=""
    SCROLL="no" SCROLLFLAT="no"
    SELECTION="no"
    SHOWINTASKBAR="no"
    SINGLEINSTANCE="no"
    SYSMENU="yes"
    VERSION="1.0"
    WINDOWSTATE="normal"
>
<link rel="stylesheet" type="text/css" href="gfx/stylesheet.css">

</head>

<SCRIPT Language="VBScript">

    //////////////////////////////////////
    // GLOBAL VARIABLES //////////////////////////////////////
    //////////////////////////////////////
    Dim ScriptRoot, ConfigFile, System32Dir
    Dim iniTitle, iniSubtitle, iniCopyRightNotice
    Dim iniSetupModel, iniSetupMode2, iniSetupMode3
    Dim iniSoftwareOU, iniComputer, iniOU, iniOSSource, iniUnattendFile, iniQuickFormat,
    iniDefaultKeyboard
    Dim iniLocalProfilesPath, iniServerProfilesFolder, iniEduUsers, sEduUsers(), EduUsersCount
    Dim iniMethod1Label, iniMethod2Label, iniDefaultMethod
    Dim uShare, uConfigShare, uGroupsIni, uAssetTag, uAssetFile, uResourceDomain, uAccountDomain ,
    uOU, uComputername, uKeyboard
    Dim sDisk, sPart, sDriveLetter, sPartSize, sInstallMethod
    Dim IsClchecked

    //////////////////////////////////////
    // INTERNAL FUNCTIONS //////////////////////////////////////
    //////////////////////////////////////
    Sub Window_Onload
        On Error Resume Next
        document.body.style.cursor = "wait"

        GetVariables()

        FillOptionKeyboard()
        FillOptionSoftware()

        If CheckUnityAssetFile()=True Then
            If GetDisks()=True Then
                'default selection
                Form1.R1(1).Checked = True
                'visibility
                Form1.R1(0).Disabled = False
                Form1.R1(1).Disabled = False
                Form1.R1(2).Disabled = False

                R1_OnClick()
            Else
                'default selection
                Form1.R1(0).Checked = True

```

```

        'visibility
        Form1.R1(0).Disabled = True
        Form1.R1(1).Disabled = True
        Form1.R1(2).Disabled = True

        R1_OnClick()
    End If
Else
    GetDisks()
    'default selection
    Form1.R1(0).Checked = True
    'visibility
    Form1.R1(0).Disabled = True
    Form1.R1(1).Disabled = True
    Form1.R1(2).Disabled = True

    R1_OnClick()

End If

Main.style.visibility = "visible"
document.body.style.cursor = "default"
On Error Goto 0
End Sub

////////////////////////////////////
Function GetVariables
    Dim objFSO, ScriptFilename, cmdline

    'Getting the folder path of Scriptroot and the file path of ConfigFile
    cmdline = Replace(objHTASetup.CommandLine, "", "")
    cmdline = Left( cmdline, InStr(cmdline, ".hta ") + 4 )
    ScriptRoot = Left(cmdline, InStrRev(cmdline, "\") - 1)
    ConfigFile = ScriptRoot & "\config.ini"

    'Loading Unity variables...
    cmdline = Replace(objHTASetup.CommandLine, "", "")
    cmdline = Mid( cmdline, InStr(cmdline, ".hta ") + 5 )
    Arguments = Split(cmdline, "/arg:")

    uAccountDomain = Arguments(1)
    uResourceDomain = Arguments(1)

    uShare = Arguments(2)
    uConfigShare = uShare & "\Config"
    uGroupsIni = uConfigShare & "\UnityGroups.ini"

    OSSHare = SourceShare & "\" & GetINIStrng("Unity", "OS1Path", "", ConfigFile) & "\i386"

    'Reading configuration values
    iniTitle = GetINIStrng("General", "Title", "error", ConfigFile)
    iniSubtitle = GetINIStrng("General", "Subtitle", "error", ConfigFile)
    iniSetupModel = GetINIStrng("General", "SetupModelLabel", "error", ConfigFile)
    iniSetupMode2 = GetINIStrng("General", "SetupMode2Label", "error", ConfigFile)
    iniSetupMode3 = GetINIStrng("General", "SetupMode3Label", "error", ConfigFile)
    iniCopyRightNotice = GetINIStrng("General", "CopyRightNotice", "error", ConfigFile)
    iniSoftwareOU = GetINIStrng("Unity", "SoftwareOU", "", ConfigFile)
    iniOSSource = uShare & "\" & GetINIStrng("Unity", "OSSource", "", ConfigFile)
    iniUnattendFile = uShare & "\" & GetINIStrng("Unity", "UnattendFile", "", ConfigFile)
    iniMethod1Label = GetINIStrng("Software", "Method1Label", "", ConfigFile)
    iniMethod2Label = GetINIStrng("Software", "Method2Label", "", ConfigFile)
    iniDefaultMethod = GetINIStrng("Software", "DefaultMethod", "", ConfigFile)
    iniDefaultKeyboard = GetINIStrng("Keyboard", "DefaultKeyboard", "", ConfigFile)
    iniOU = GetINIStrng("ActiveDirectory", "OU" & GetINIStrng("ActiveDirectory",
"DefaultOU", "", ConfigFile) & "Path", "", ConfigFile)
    iniQuickFormat = GetINIStrng("DiskFormat", "QuickFormat", "", ConfigFile)

    'Reading Unity Asset Information
    iniComputer = GetUnityAssetValue("Computer")

    Set WshShell = CreateObject("WScript.Shell")
    System32Dir = WshShell.ExpandEnvironmentStrings("%SystemRoot%\system32")

End Function

////////////////////////////////////
Function SelectSetupModel()
    'format information

```

```

        lblFormatOption.innerHTML = "(Recreating and formatting all partitions on disk
        "&sDisk&")"

        'Enable and clear OptionPartsize
        Form1.OptionPartsize.Disabled = False
        clearOptions(Form1.OptionPartsize)

        'Automatically select a predefined partition size
        FillOptionPartsize()
        SelectOptionPartsize( GetDiskSize(sDisk) )

        'asset information handling
        Form1.C1.checked = True
        Form1.C1.disabled = True
        lblAssetInformation.innerHTML = "Create new asset file"

        Form1.txtComputername.disabled = False
        Form1.OptionKeyboard.disabled = False

        lblEDUPprofiles.innerHTML = "Educational Profiles"
        EDUPprofiles.innerHTML = "(N/A)"

        Form1.R2(0).Disabled = True
        Form1.R2(1).Disabled = True
        Form1.R2(1).Checked = True

End Function

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Function SelectSetupMode2()
    'format information
    lblFormatOption.innerHTML = "(Formatting partition "&GetDriveLetter(sDisk, sPart)&" on
    disk "&sDisk&" only)"

    'Disable and clear OptionPartsize
    Form1.OptionPartsize.Disabled = True
    clearOptions(Form1.OptionPartsize)

    'Fill OptionPartsize with currently selected partition size
    Set objOption = Document.createElement("OPTION")
    Dim Partsize : Partsize = GetPartSize(Form1.OptionDisk.Value, Form1.OptionPart.Value)
    objOption.Text = Round( Partsize / 1000 )&" GB"           'Text in GigaByte
    objOption.Value = Partsize                               'Value in
MegaByte
    Form1.OptionPartsize.Add(objOption)

    'asset information handling
    Form1.C1.checked = True
    Form1.C1.disabled = False
    lblAssetInformation.innerHTML = "Inherit Asset Information"

    Form1.txtComputername.value = GetUnityAssetValue("Computer")
    Form1.txtComputername.disabled = True

    lblEDUPprofiles.innerHTML = "Educational Profiles"
    EDUPprofiles.innerHTML = "(N/A)"

    Form1.OptionKeyboard.value = GetUnityAssetValue("Keyboard")
    Form1.OptionKeyboard.disabled = True

    Form1.R2(0).Disabled = False
    Form1.R2(1).Disabled = False
    Form1.R2(0).Checked = True

    Form1.OptionSoftware.value = GetUnityAssetValue("OU")

End Function

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Function SelectSetupMode3()
    'format information
    lblFormatOption.innerHTML = "(No filesystem changes)"

    'Disable and clear OptionPartsize
    Form1.OptionPartsize.Disabled = True
    clearOptions(Form1.OptionPartsize)

    'Fill OptionPartsize with currently selected partition size
    Set objOption = Document.createElement("OPTION")
    Dim Partsize : Partsize = GetPartSize(Form1.OptionDisk.Value, Form1.OptionPart.Value)
    objOption.Text = Round( Partsize / 1000 )&" GB"           'Text in GigaByte

```

```

        objOption.Value = Partsize                                     'Value in
MegaByte
        Form1.OptionPartsize.Add(objOption)

        'asset information handling
        Form1.C1.checked = True
        Form1.C1.disabled = True
        lblAssetInformation.innerHTML = "Inherit Asset Information"

        Form1.txtComputername.value = GetUnityAssetValue("Computer")
        Form1.txtComputername.disabled = True

        lblEDUPProfiles.innerHTML = "Educational Profiles"
        Call LoadEDUPProfiles()

        Form1.OptionKeyboard.value = GetUnityAssetValue("Keyboard")
        Form1.OptionKeyboard.disabled = True

        Form1.R2(0).Disabled = False
        Form1.R2(1).Disabled = False
        Form1.R2(0).Checked = True

        Form1.OptionSoftware.value = GetUnityAssetValue("OU")

End Function

////////////////////////////////////
Function RunSetupModel()
    'changing view
    Main.innerHTML = "<table class='dataTable' width='100%>"_
        &"<tr class='bgColor2' bg='#CCCCCC' fs='1'>"_
        &"<td class='dataTableHeaderCell' colspan='2'><div class='mainContentSmall
mainContentBold'>Please wait... </div></td>"_
        &"</tr><tr id='Row1'>"_
        &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx1'></td>"_
        &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep1' class='mainContentSmall'></div></td></td>"_
        &"</tr><tr id='Row2'>"_
        &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx2'></td>"_
        &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep2' class='mainContentSmall'></div></td></td>"_
        &"</tr><tr id='Row3'>"_
        &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx3'></td>"_
        &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep3' class='mainContentSmall'></div></td></td>"_
        &"</tr><tr id='Row4'>"_
        &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx4'></td>"_
        &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep4' class='mainContentSmall'></div></td></td>"_
        &"</tr><tr id='Row5'>"_
        &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx5'></td>"_
        &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep5' class='mainContentSmall'></div></td></td>"_
        &"</tr>"_
        &"<tr bg='#FFAAAA'>"_
        &"<td style='border-top: 1px solid #000000;' width='4%'>&nbsp;</td>"_
        &"<td style='border-top: 1px solid #000000;' width='95%'>&nbsp;</td>"_
        &"</tr>"_
    &"</table>"

    Form1.Cx1.disabled = True
    Form1.Cx2.disabled = True
    Form1.Cx3.disabled = True
    Form1.Cx4.disabled = True
    Form1.Cx5.disabled = True

    SetupStep1.innerHTML = "Creating new asset file"
    SetupStep2.innerHTML = "Creating partitions"
    SetupStep3.innerHTML = "Formatting partitions"
    SetupStep4.innerHTML = "Starting Windows Setup"
    SetupStep5.innerHTML = "Restarting Computer"

    '-----
    window.status = ""
    'Creating new asset file
    Row1.style.backgroundColor = "#EEEEEE"

```

```

SetupStep1.innerHTML = "Creating new asset file on "&uShare

Call CreateUnityAssetFile()

'Setting asset information
Call UpdateUnityAssetFile("Computer", uComputername)
Call UpdateUnityAssetFile("LoginID", "Administrator")
Call UpdateUnityAssetFile("UniqueID", uAssetTag)
Call UpdateUnityAssetFile("AccountDomain", uAccountDomain)
Call UpdateUnityAssetFile("ResourceDomain", uResourceDomain)
Call UpdateUnityAssetFile("OU", uOU)
Call UpdateUnityAssetFile("ResourceOU", iniOU&"@"&uResourceDomain)
Call UpdateUnityAssetFile("InstallShare", uShare)
Call UpdateUnityAssetFile("Keyboard", uKeyboard)
Call UpdateUnityAssetFile("AutoStart", "TRUE")
Call UpdateUnityAssetFile("AutoEnd", "TRUE")
Call UpdateUnityAssetFile("Finish", "TRUE")

SetupStep1.innerHTML = "Asset file created successfully"
Row1.style.backgroundColor = "#FFFFFF"
Form1.Cx1.checked = True

'-----
'Partitioning
window.status = ""
Row2.style.backgroundColor = "#EEEEEE"
DriveLetter1 = "C:"
SetupStep2.innerHTML = "Creating partition "&DriveLetter1
Call Partitioning(sDisk, 0, sPartSize, DriveLetter1)

DriveLetter2 = "D:"
SetupStep2.innerHTML = "Creating partition "&DriveLetter2
Call Partitioning(sDisk, 1, 0, DriveLetter2)

SetupStep2.innerHTML = "Partitions created successfully"
Row2.style.backgroundColor = "#FFFFFF"
Form1.Cx2.checked = True

'-----
'Formatting
window.status = ""
Row3.style.backgroundColor = "#EEEEEE"

DiskLabel1 = GetINIStrng("DiskFormat", "DefaultDiskLabel1", "Local Disk", ConfigFile)
SetupStep3.innerHTML = "Formatting partition "&DriveLetter1&" ("&DiskLabel1&")
Call Format(DriveLetter1, DiskLabel1)

DiskLabel2 = GetINIStrng("DiskFormat", "DefaultDiskLabel2", "Local Disk", ConfigFile)
SetupStep3.innerHTML = "Formatting partition "&DriveLetter2&" ("&DiskLabel2&")
Call Format(DriveLetter2, DiskLabel2)

SetupStep3.innerHTML = "Partitions formatted successfully"
Row3.style.backgroundColor = "#FFFFFF"
Form1.Cx3.checked = True

'-----
'Starting Windows Unattended Setup
window.status = ""
Row4.style.backgroundColor = "#EEEEEE"
SetupStep4.innerHTML = "Windows Setup is preparing"

Call StartWindowsSetup()

SetupStep4.innerHTML = "Windows Setup has prepared successfully"
Row4.style.backgroundColor = "#FFFFFF"
Form1.Cx4.checked = True

'-----
'Restarting computer
window.status = ""
SetupStep5.innerHTML = "Restarting Computer"
Row5.style.backgroundColor = "#EEEEEE"
Call Reboot()
Row5.style.backgroundColor = "#FFFFFF"
Form1.Cx5.checked = True

End Function

////////////////////////////////////
Function RunSetupMode2()
    'changing view

```

```

Main.innerHTML = "<table class='dataTable' width='100%>"_
    &"<tr class='bgColor2' bg='#CCCCCC' fs='1'>"_
    &"<td class='dataTableHeaderCell' colspan='2'><div class='mainContentSmall
mainContentBold>Please wait... </div></td>"_
    &"</tr><tr id='Row1'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx1'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep1' class='mainContentSmall'></div></td></td>"_
    &"</tr><tr id='Row2'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx2'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep2' class='mainContentSmall'></div></td></td>"_
    &"</tr><tr id='Row3'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx3'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep3' class='mainContentSmall'></div></td></td>"_
    &"</tr><tr id='Row4'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx4'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%><div
id='SetupStep4' class='mainContentSmall'></div></td></td>"_
    &"</tr>"_
    &"<tr bg='#FFAAAA'>"_
    &"<td style='border-top: 1px solid #000000;' width='4%'>&nbsp;</td>"_
    &"<td style='border-top: 1px solid #000000;' width='95%'>&nbsp;</td>"_
    &"</tr>"_
    &"</table>"

Form1.Cx1.disabled = True
Form1.Cx2.disabled = True
Form1.Cx3.disabled = True
Form1.Cx4.disabled = True

SetupStep1.innerHTML = "Updating asset information"
SetupStep2.innerHTML = "Formatting system partition"
SetupStep3.innerHTML = "Starting Windows Setup"
SetupStep4.innerHTML = "Restarting Computer"

'-----
'Updating asset information
window.status = ""
Row1.style.backgroundColor = "#EEEEEE"
SetupStep1.innerHTML = "Updating asset information on "&uShare

Call CreateUnityAssetFile()

'updating asset information
Call UpdateUnityAssetFile("Computer", uComputername)
Call UpdateUnityAssetFile("LoginID", "Administrator")
Call UpdateUnityAssetFile("UniqueID", uAssetTag)
Call UpdateUnityAssetFile("AccountDomain", uAccountDomain)
Call UpdateUnityAssetFile("ResourceDomain", uResourceDomain)
Call UpdateUnityAssetFile("OU", uOU)
Call UpdateUnityAssetFile("ResourceOU", iniOU&"@"&uResourceDomain)
Call UpdateUnityAssetFile("InstallShare", uShare)
Call UpdateUnityAssetFile("Keyboard", uKeyboard)
Call UpdateUnityAssetFile("AutoStart", "TRUE")
Call UpdateUnityAssetFile("AutoEnd", "TRUE")
Call UpdateUnityAssetFile("Finish", "TRUE")

SetupStep1.innerHTML = "Asset information updated successfully"
Row1.style.backgroundColor = "#FFFFFF"
Form1.Cx1.checked = True

'-----
'Formatting
window.status = ""
Row2.style.backgroundColor = "#EEEEEE"

DriveLetter = GetDriveLetter(sDisk, 0)
DiskLabel = GetINIStr("DiskFormat", "DefaultDiskLabel1", "Local Disk", Config File)
SetupStep2.innerHTML = "Formatting partition "&DriveLetter&" ("&DiskLabel&)"
Call Format(DriveLetter, DiskLabel)

SetupStep2.innerHTML = "Partition formatted successfully"
Row2.style.backgroundColor = "#FFFFFF"
Form1.Cx2.checked = True

```



```

'-----
'Starting Windows Unattended Setup
window.status = ""
Row3.style.backgroundColor = "#EEEEEE"
SetupStep3.innerHTML = "Windows Setup is preparing"

Call StartWindowsSetup()

SetupStep3.innerHTML = "Windows Setup has prepared successfully"
Row3.style.backgroundColor = "#FFFFFF"
Form1.Cx3.checked = True

'-----
'Restarting computer
window.status = ""
SetupStep4.innerHTML = "Restarting Computer"
Row4.style.backgroundColor = "#EEEEEE"
Call Reboot()
Row4.style.backgroundColor = "#FFFFFF"
Form1.Cx4.checked = True

End Function

////////////////////////////////////
Function RunSetupMode3()
'changing view
Main.innerHTML = "<table class='dataTable' width='100%'>"_
    &"<tr class='bgColor2' bg='#CCCCCC' fs='1'>"_
    &"<td class='dataTableHeaderCell' colspan='2'><div class='mainContentSmall
mainContentBold'>Please wait... </div></td>"_
    &"</tr><tr id='Row1'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx1'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%'><div
id='SetupStep1' class='mainContentSmall'></div></td></td>"_
    &"</tr><tr id='Row2'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx2'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%'><div
id='SetupStep2' class='mainContentSmall'></div></td></td>"_
    &"</tr><tr id='Row3'>"_
    &"<td class='dataTableCellMiddle' align='left'><input type='checkbox' value='1'
name='Cx3'></td>"_
    &"<td class='dataTableCellMiddle' style='text-align: left' width='95%'><div
id='SetupStep3' class='mainContentSmall'></div></td></td>"_
    &"</tr></tr>"_
    &"<tr bg='#FFAAAA'>"_
    &"<td style='border-top: 1px solid #000000;' width='4%'>&nbsp;</td>"_
    &"<td style='border-top: 1px solid #000000;' width='95%'>&nbsp;</td>"_
    &"</tr>"_
    &"</table>"

Form1.Cx1.disabled = True
Form1.Cx2.disabled = True
Form1.Cx3.disabled = True

SetupStep1.innerHTML = "Cleaning local profiles"
SetupStep2.innerHTML = "Cleaning profiles on server"
SetupStep3.innerHTML = "Restarting Computer"

'-----
'Updating software config
Call UpdateSoftwareSet()

'updating asset information
Call UpdateUnityAssetFile("OU", uOU)

'-----
'cleaning local profiles
window.status = ""
Row1.style.backgroundColor = "#EEEEEE"

Call CleanEDUPProfilesLocal()

SetupStep1.innerHTML = "Local profiles cleaned successfully"
Row1.style.backgroundColor = "#FFFFFF"
Form1.Cx1.checked = True

```

```

'-----
'cleaning server profiles
window.status = ""
Row2.style.backgroundColor = "#EEEEEE"

Call CleanEDUPprofilesServer()

SetupStep2.innerHTML = "Profiles on server cleaned successfully"
Row2.style.backgroundColor = "#FFFFFF"
Form1.Cx2.checked = True

'-----
'restarting computer
SetupStep3.innerHTML = "Restarting Computer"
Row3.style.backgroundColor = "#EEEEEE"
Call Reboot()
Row3.style.backgroundColor = "#FFFFFF"
Form1.Cx3.checked = True
End Function

////////////////////////////////////
Function CheckUnityAssetFile()
    Dim fso, returnvalue
    Set fso = CreateObject("Scripting.FileSystemObject")

    Set objWMIService = GetObject("winmgmts:\\.\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_ComputerSystemProduct",,48)

    For Each objItem in colItems
        If objItem.Version = "" Then
            '[VENDOR]~[TYPE]~[SERIAL]
            uAssetTag = objItem.Vendor&"~"&objItem.Name&"~"&objItem.IdentifyingNumber
        Else
            '[VENDOR]~[TYPE]~[VERSION]~[SERIAL]
            uAssetTag =
objItem.Vendor&"~"&objItem.Name&"~"&objItem.Version&"~"&objItem.IdentifyingNumber
        End If
    Next

    uAssetTag = Ucase(uAssetTag)
    uAssetFile = uConfigShare&"\"&uAssetTag&".INI"
    'msgbox uAssetFile
    If (fso.FileExists(uAssetFile))=True Then
        'function returns "TRUE", if uAssetFile already exists
        returnvalue=True

    Else
        'function returns "FALSE", if uAssetFile does not exist yet
        returnvalue=False
    End If
CheckUnityAssetFile=returnvalue
End Function

////////////////////////////////////
Function CreateUnityAssetFile()
    'Recreating asset information file (!)
    Dim fso, tf
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set tf = fso.CreateTextFile(uAssetFile, True)
    tf.Close
End Function

////////////////////////////////////
Function UpdateUnityAssetFile(iField, iValue)
    Call WriteINIString("RunUnity",iField, iValue, uAssetFile)
End Function

////////////////////////////////////
Function GetUnityAssetValue(Key)
    GetUnityAssetValue = GetINIString("RunUnity", Key, "", uAssetFile)
End Function

////////////////////////////////////
Function UpdateSoftwareSet
    Dim PackagesDefault : PackagesDefault = GetINISections("OU - <default>", uGroupsIni)
    Dim PackagesGroup : PackagesGroup = GetINISections("OU - "&uOU, uGroupsIni)
    Dim PackagesAsset : PackagesAsset = GetINISections("Packages" ,uAssetFile)
    Dim SetFlag : SetFlag = False

```

```

        If sInstallMethod = 1 Then
            For i=1 To Ubound(PackagesAsset)-1
                tmpPA = PackagesAsset(i)
                tmpPAi = InStr(tmpPA,"=")-1
                If tmpPAi>0 Then
                    Call WriteINIString("Packages", Left(tmpPA,tmpPAi), "TOREMOVE",
uAssetFile)
                End If
            Next

            For i=1 To Ubound(PackagesDefault)-1
                tmpPD = PackagesDefault(i)
                tmpPDi = InStr(tmpPD,"=")-1
                If tmpPDi >=0 Then
                    If InStr(Ucase(Mid(tmpPD,tmpPDi+2)),"TOINSTALL")>0 Then
                        Call WriteINIString("Packages", Left(tmpPD,tmpPDi),
Mid(tmpPD,tmpPDi+2), uAssetFile)
                    End If
                End If
            Next

            For i=1 To Ubound(PackagesGroup)-1
                tmpPG = PackagesGroup(i)
                tmpPGi = InStr(tmpPG,"=")-1
                If tmpPGi >=0 Then
                    Call WriteINIString("Packages", Left(tmpPG,tmpPGi),
Mid(tmpPG,tmpPGi+2), uAssetFile)
                End If
            Next

        Else
            For i=1 To Ubound(PackagesAsset)-1
                tmpPA = PackagesAsset(i)
                tmpPAi = InStr(tmpPA,"=")-1
                If tmpPAi>0 Then
                    Call WriteINIString("Packages", Left(tmpPA,tmpPAi), "TOREMOVE",
uAssetFile)
                End If
            Next
        End If
    End Function

    //////////////////////////////////////
Function GetDisks()
    Dim strArray(), DiskDriveCount, Temp, Count, lastDiskNumber, i

    Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_DiskDrive",,48)

    Count=0
    For Each objItem In colitems
        Count=Count+1
    Next

    ReDim Preserve strArray(Count,2)

    Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_DiskDrive",,48)

    i=1
    For Each objItem In colitems

        Set objOption = Document.createElement("OPTION")
        objOption.Text = "Disk "&objItem.Index&": "& objItem.Caption &
("&round(objItem.Size/1000000000) & " GB)"
        objOption.Value = objItem.Index

        strArray(i,1) = objOption.Text
        strArray(i,2) = objOption.Value

        Form1.OptionDisk.Add(objOption)
        i=i+1
    Next

    'Sorting OptionDisk
    Sort strArray, Form1.OptionDisk

    'default selection
    Form1.OptionDisk.value = 0

    Getdisks=OptionDisk_OnChange()
End Function

```

```

////////////////////////////////////
Function GetPart(disknumber)
    Dim Temp, PartCount, DriveLetter

    Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_DiskPartition WHERE Name
LIKE '%" & disknumber & "%'", , 48)

    'remove OptionPart-Items
    For Each objOption in Form1.OptionPart.Options
        objOption.RemoveNode
    Next

    PartCount = 0
    For Each objItem In colItems
        'Add Partitions to OptionPart
        Set objOption = Document.createElement("OPTION")
        DriveLetter = GetDriveLetter(disknumber, Mid(objItem.Name, Len(objItem.Name), 1))
        objOption.Text = "Partition " & DriveLetter & " (" &
Round(objItem.Size/1000000000) & " GB)"
        objOption.Value = Mid(objItem.Name, Len(objItem.Name), 1)
        Form1.OptionPart.Add(objOption)
        PartCount = PartCount+1
    Next

    If PartCount > 0 Then
        OptionPartDIV.style.visibility = "visible"
        Form1.OptionPart.value = 0
        OptionPart_OnChange()
        GetPart=True
    Else
        OptionPartDIV.style.visibility = "hidden"
        GetPart=False
    End If
End Function

////////////////////////////////////
Function GetPartSize(disknumber, partnumber)
    Dim Returnvalue

    Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_DiskPartition WHERE Name
LIKE '%" & disknumber & "%'&partnumber&'", , 48)

    For Each objItem In colItems
        'objItem.Size contains the partitions size in Bytes. 1'000'000 Bytes = 1
MegaByte.
        Returnvalue = Round( objItem.Size / 1000000 )
    Next

    'Returnvalue => Partition size in MegaByte
    GetPartSize = Returnvalue
End Function

////////////////////////////////////
Function GetDriveLetter(Disknumber, Partnumber)
    Disknumber = CInt(Disknumber)
    Partnumber= CInt(Partnumber)

    Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_LogicalDiskToPartition", , 48)

    For Each objItem In colItems
        Dim Disk, Part, Driveletter
        Disk = CInt( Left(Mid(objItem.Antecedent, InStr(objItem.Antecedent, ",") -1), 1) )
        Part = CInt( Right(Left(objItem.Antecedent, Len(objItem.Antecedent)-1), 1) )

        'Debug Information:
        'msgbox "Disk #" & disk & " Part #" & part & " letter=" & Left(Mid(objItem.Dependent,
Len(objItem.Dependent)-2), 2)

        If Disk=Disknumber AND Part=Partnumber Then
            Driveletter = Left(Mid(objItem.Dependent, Len(objItem.Dependent)-2), 2)
        End If
    Next
    GetDriveLetter=Driveletter
End Function

////////////////////////////////////
Function GetDiskSize(disknumber)

```

```

Dim returnvalue
If disknumber="" Then
    disknumber=0
End If

Set objWMIService = GetObject("winmgmts:\\localhost\root\CIMV2")
Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_DiskDrive WHERE
Index=" & disknumber, , 48)

For Each objItem In colItems
    returnvalue = round(objItem.Size/1000000)
Next
GetDiskSize = returnvalue
End Function

////////////////////////////////////
Function Partitioning(disknumber, partnumber, partsize, diskletter)
    Dim PartScriptFile : PartScriptFile = ScriptRoot & "\Part.txt"
    disknumber = Cint(disknumber)
    partnumber = Cint(partnumber)

    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Dim fso, f
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.CreateTextFile(PartScriptFile, True)

    If disknumber=0 AND partnumber=0 Then
        f.WriteLine("select disk " & disknumber)
        f.WriteLine("clean")
        f.WriteLine("create partition primary size=" & partsize)
        f.WriteLine("assign letter=" & diskletter)
        f.WriteLine("active")
    Else
        f.WriteLine("select disk " & disknumber)
        f.WriteLine("create partition primary")
        f.WriteLine("assign letter=" & diskletter)
    End If

    f.Close

    Cmd = System32Dir & "\DISKPART.EXE /s " & PartScriptFile
    'msgbox Cmd

    Call WshShell.Run(Cmd, 0, True)

    fso.DeleteFile PartScriptFile
End Function

////////////////////////////////////
Function Format(iDrive, iLabel)
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    If Lcase(iniQuickFormat)=Lcase("Yes") Then
        'Quick Format
        Cmd = "FORMAT.COM " & iDrive & " /V:" & iLabel & " /FS:NTFS /Q /X /Y"
    Else
        'Normal Format
        Cmd = "FORMAT.COM " & iDrive & " /V:" & iLabel & " /FS:NTFS /X /Y"
    End If

    'msgbox Cmd
    Set objShell = CreateObject("WScript.Shell")
    objShell.Run(Cmd), 0, True
End Function

////////////////////////////////////
Function StartWindowsSetup()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = iniOSSource & "\WINNT32.EXE /syspart:" & GetDriveLetter(sDisk, sPart) &
/s:" & iniOSSource & " /unattend:" & iniUnattendFile
    'msgbox Cmd

    Set objShell = CreateObject("WScript.Shell")
    objShell.Run(Cmd), 0, True
End Function

```

```

////////////////////////////////////
Function OptionDisk_OnChange()
    sDisk = Form1.OptionDisk.value
    OptionDisk_OnChange=GetPart(sDisk)
End Function

////////////////////////////////////
Function OptionPart_OnChange()
    sPart = Form1.OptionPart.value
    sDriveLetter = GetDriveLetter(sDisk, sPart)
    label2.innerHTML = "Partition Size (" & sDriveLetter & ")"
    R1_OnClick()
End Function

////////////////////////////////////
Function R1_OnClick()
    If Form1.R1(0).Checked Then
        SelectSetupModel()
    End If
    If Form1.R1(1).Checked Then
        SelectSetupMode2()
    End If
    If Form1.R1(2).Checked Then
        SelectSetupMode3()
    End If
End Function

////////////////////////////////////
Function C1_OnClick()
    If Form1.C1.checked Then
        R1_OnClick()
    Else
        Form1.txtComputername.disabled = False
        Form1.OptionKeyboard.disabled = False
    End If
End Function

////////////////////////////////////
Function Start()
    'validating input values
    If InputValidation() Then
        'setting temporary variables
        uComputername = Form1.txtComputername.value
        uKeyboard = Form1.OptionKeyboard.value
        uOU = Form1.OptionSoftware.value
        sPartSize = Form1.OptionPartsize.value
        If Form1.C1.checked Then
            IsC1checked = True
        Else
            IsC1selected = False
        End If

        ReDim sEduUsers(Int(EduUsersCount)-1)
        For i=0 To Ubound(sEduUsers)
            If Form1.C2(i).Checked Then
                sEduUsers(i) = Form1.C2(i).value
            End If
        Next

        If Form1.R2(0).Checked Then
            sInstallMethod = 1
        Else
            sInstallMethod = 2
        End If

        'SETUP MODE 1 -----
        If Form1.R1(0).Checked Then
            document.body.style.cursor = "wait"

            'refreshing UI
            Main.innerHTML = " "
            window.status = " "

            RunSetupModel()

            document.body.style.cursor = "default"
        End If
        'SETUP MODE 2 -----
        If Form1.R1(1).Checked Then
            document.body.style.cursor = "wait"

```

```

        'refreshing UI
        Main.innerHTML = ""
        window.status = ""

        RunSetupMode2()

        document.body.style.cursor = "default"
        Exit Function
    End If
    'SETUP MODE 3 -----
    If Form1.R1(2).Checked Then
        document.body.style.cursor = "wait"

        'refreshing UI
        Main.innerHTML = ""
        window.status = ""

        RunSetupMode3()

        document.body.style.cursor = "default"
        Exit Function
    End If
End If
End Function

////////////////////////////////////
Function clearOptions(inpList)
    Dim objOption
    For Each objOption in inpList.Options
        objOption.RemoveNode
    Next
End Function

////////////////////////////////////
Function FillOptionPartsize
    Dim fso, f, IsSection, line, linevalue
    Dim objOption, OptionsCount

    clearOptions(Form1.OptionPartsize)

    Const Reading=1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.OpenTextFile(ConfigFile, Reading)

    OptionsCount=0
    Do Until f.AtEndOfStream
        line = f.ReadLine
        If InStr(line, "[PartSizeChoice]") Then
            OptionsCount = OptionsCount+1
        End If
    Loop

    For i=1 To OptionsCount
        Set objOption = Document.createElement("OPTION")
        For Each Row in GetINISections("PartSizeChoice"&i,ConfigFile)
            If InStr(Row,"Text") Then
                objOption.Text = Mid(Row, InStr(Row, "=")+1)
            End If
            If InStr(Row,"SysPartSize") Then
                objOption.Value = Mid(Row, InStr(Row, "=")+1)
            End If
        Next
        Form1.OptionPartsize.Add(objOption)
    Next
End Function

////////////////////////////////////
Function FillOptionKeyboard
    Dim objOption, OptionsCount, strArray()

    clearOptions(Form1.OptionKeyboard)

    Languages = GetINISections("Keyboard", uConfigShare&"\UnitySystem.ini")

    OptionsCount=Ubound(Languages)
    ReDim strArray(OptionsCount,2)

    for i=1 To OptionsCount
        Set objOption = Document.createElement("OPTION")
        objOption.Text = Replace(Left( Languages(i), InStr(Languages(i), "=") ), "=", "")
    
```

```

        objOption.Value = objOption.Text
        strArray(i,1) = objOption.Text
        strArray(i,2) = objOption.Value
        Form1.OptionKeyboard.Add(objOption)
    Next

    Sort strArray, Form1.OptionKeyboard

    'Default selection
    Form1.OptionKeyboard.value = iniDefaultKeyboard

End Function

////////////////////////////////////
Function SelectOptionPartsize(DiskSize)
    Dim fso, f, line, Count, OptionsCount
    Dim MinDiskSpace, MaxDiskSpace, SysPartSize

    Const Reading=1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.OpenTextFile(ConfigFile, Reading)

    OptionsCount=0
    Do Until f.AtEndOfStream
        line = f.ReadLine
        If InStr(line, "[PartSizeChoice]") Then
            OptionsCount = OptionsCount+1
        End If
    Loop

    For i=1 To OptionsCount
        Set objOption = Document.createElement("OPTION")
        For Each Row in GetINISections("PartSizeChoice"&i,ConfigFile)
            If InStr(Row,"MinDiskSpace") Then
                MinDiskSpace = Mid(Row, InStr(Row, "=")+1)
            End If

            If InStr(Row,"MaxDiskSpace") Then
                MaxDiskSpace = Mid(Row, InStr(Row, "=")+1)
            End If

            If InStr(Row,"SysPartSize") Then
                SysPartSize = Mid(Row, InStr(Row, "=")+1)
            End If
        Next

        If Int(DiskSize) >= Int(MinDiskSpace) AND Int(DiskSize) <= Int(MaxDiskSpace) Then
            Form1.OptionPartsize.value = SysPartSize
            Exit Function
        ElseIf i=Count Then
            Form1.OptionPartsize.value = SysPartSize
            Exit Function
        End If

        'Debug Information:
        'msgbox "min:"&MinDiskSpace & " max:"&MaxDiskSpace & " =>
        SysPartSize:"&SysPartSize
    Next
End Function

////////////////////////////////////
Function FillOptionSoftware
    Dim objOption, OptionsCount, strArray(), SoftwareSets

    clearOptions(Form1.OptionSoftware)

    Dim fso, f, IsSection, line, linevalue
    Const Reading=1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.OpenTextFile(uGroupsIni, Reading)

    Do Until f.AtEndOfStream
        line = f.ReadLine
        If InStr(line, "[OU - "&iniSoftwareOU) Then
            Set objOption = Document.createElement("OPTION")
            objOption.Text = Replace(Replace(line,"[OU - ",""),"],"","")
            objOption.Value = objOption.Text
            Form1.OptionSoftware.Add(objOption)
        End If
    Loop
End Function

```



```

////////////////////////////////////
Function CleanEDUPProfilesLocal
    Dim FSO

    For i=0 To Ubound(sEduUsers)
        If sEduUsers(i)<>" Then
            Set FSO = CreateObject("Scripting.FileSystemObject")
            FSO.DeleteFolder iniLocalProfilesPath&"\"&sEduUsers(i), True
        End If
    Next
End Function

////////////////////////////////////
Function CleanEDUPProfilesServer
    Dim FSO, WshShell, ProfileFolder, ProfileSubFolder

    For i=0 To Ubound(sEduUsers)
        If sEduUsers(i)<>" Then
            Set FSO = CreateObject("Scripting.FileSystemObject")
            iniServerProfilesFolder = GetINIString("Profiles", "ServerProfilesPath",
            "", ConfigFile)
            Set ProfileFolder =
            FSO.GetFolder(iniServerProfilesFolder&"\"&sEduUsers(i)&"$")

            'deleting all files in profile folder
            FSO.DeleteFile ProfileFolder&"\*.*", True

            'deleting all subfolder in profile folder
            For Each ProfileSubFolder in ProfileFolder.SubFolders
                Select Case True
                    Case (ProfileSubFolder.name = "Notes")
                        'deleting subfolders of the "Notes" directory
                        For Each Folder in ProfileSubFolder.SubFolders
                            FSO.DeleteFolder Folder, True
                        Next

                        'deleting files of the "Notes" directory
                        For Each File in ProfileSubFolder.Files
                            If Lcase(File.Name)<>Lcase(sEduUsers(i)&".id") Then
                                FSO.DeleteFile File, True
                            End If
                            If (lcase(File.Name)<>"names.nsf") Then
                                FSO.DeleteFile File, True
                            End If
                        Next
                    Case Else
                        FSO.DeleteFolder ProfileSubFolder, True
                End Select
            Next
        End If
    Next
End Function

////////////////////////////////////
Function LoadEDUPProfiles()
    Dim WshShell, FSO, tmpEDUPProfiles
    Set WshShell = CreateObject("WScript.Shell")

    iniLocalProfilesPath = GetDriveLetter(sDisk, sPart) &"\"& GetINIString("Profiles",
    "LocalProfilesPath", "", ConfigFile)
    iniEduUsers = GetINIString("Profiles", "EDUUsers", "", ConfigFile)
    iniEduUsers = Split(iniEduUsers, ";")
    EduUsersCount = 0

    For Each EduUser in iniEduUsers
        Set FSO = CreateObject("Scripting.FileSystemObject")
        If FSO.FolderExists(iniLocalProfilesPath&"\"&EduUser) Then
            EduUsersCount = EduUsersCount+1
            tmpEDUPProfiles = tmpEDUPProfiles &"<table border='0' width='100%'
id='table4' cellspacing='0' cellpadding='0'>"_
                                                                    &"<tr><td
width='1%'>"_
                                                                    &"<input
type='checkbox' name='C2' value='\"&EduUser&"' style='margin-left: -4'></td>"_
                                                                    &"<td><div
class='mainContentSmall' style='margin-left: 2'>Clean profile \"&EduUser&"</div></td>"_
                                                                    &"</tr>"_
                                                                    &"</table>"

            tmpEDUPProfiles = tmpEDUPProfiles &"<br>"
        End If
    Next
End Function

```

```

Next

If tmpEDUPProfiles<>" Then
    EDUPProfiles.innerHTML = tmpEDUPProfiles & "<input style='visibility:hidden;'
type='checkbox' name='C2' value='0' style='margin-left: -4'>"
End If
End Function

////////////////////////////////////
Function Reload()
    Location.Reload(True)
End Function

////////////////////////////////////
Function Shutdown()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir & "\ShutdownPE.exe shutdown"
    'msgbox Cmd

    Call WshShell.Run(Cmd,0,False)
End Function

////////////////////////////////////
Function Reboot()
    Dim WshShell, Cmd
    Set WshShell = CreateObject("WScript.Shell")

    Cmd = System32Dir & "\ShutdownPE.exe restart"
    'msgbox Cmd

    Call WshShell.Run(Cmd,0,False)
End Function

////////////////////////////////////
Function InputValidation()
    Dim IsValid : IsValid = True
    Dim msgboxTitle : msgboxTitle = "Input Validation"

    'OptionDisk Validation
    '(last chance to stop, if the target system doesn't have any local disks installed)
    If Form1.OptionDisk.value = "" Then
        Form1.OptionDisk.focus()
        MsgBox GetINIString("InputValidation", "Message1", "Error!", ConfigFile) ,
VBExclamation, msgboxTitle
        IsValid = False
        Exit Function
    End If

    'txtComputername Validation
    If Form1.txtComputername.value = "" Then
        Form1.txtComputername.focus()
        MsgBox GetINIString("InputValidation", "Message2", "Error!", ConfigFile) ,
VBExclamation, msgboxTitle
        IsValid = False
        Exit Function
    End If

    If Len(Form1.txtComputername.value)<>8 Then
        Form1.txtComputername.focus()
        MsgBox GetINIString("InputValidation", "Message3", "Error!", ConfigFile) ,
VBExclamation, msgboxTitle
        IsValid = False
        Exit Function
    End If

    InputValidation=IsValid
End Function

////////////////////////////////////
// EXTERNAL FUNCTIONS
////////////////////////////////////

Sub WriteINIString(Section, KeyName, Value, FileName)
    Dim INIContents, PosSection, PosEndSection

    'Get contents of the INI file As a string
    INIContents = GetFile(FileName)

```

```

'Find section
PosSection = InStr(1, INIContents, "[" & Section & "]", vbTextCompare)
If PosSection>0 Then
    'Section exists. Find end of section
    PosEndSection = InStr(PosSection, INIContents, vbCrLf & "[")
    '?Is this last section?
    If PosEndSection = 0 Then PosEndSection = Len(INIContents)+1

    'Separate section contents
    Dim OldsContents, NewsContents, Line
    Dim sKeyName, Found
    OldsContents = Mid(INIContents, PosSection, PosEndSection - PosSection)
    OldsContents = split(OldsContents, vbCrLf)

    'Temp variable To find a Key
    sKeyName = LCase(KeyName & "=")

    'Enumerate section lines
    For Each Line In OldsContents
        If LCase(Left(Line, Len(sKeyName))) = sKeyName Then
            Line = KeyName & "=" & Value
            Found = True
        End If
        NewsContents = NewsContents & Line & vbCrLf
    Next

    If isempty(Found) Then
        'key Not found - add it at the end of section
        NewsContents = NewsContents & KeyName & "=" & Value
    Else
        'remove last vbCrLf - the vbCrLf is at PosEndSection
        NewsContents = Left(NewsContents, Len(NewsContents) - 2)
    End If

    'Combine pre-section, new section And post-section data.
    INIContents = Left(INIContents, PosSection-1) & _
        NewsContents & Mid(INIContents, PosEndSection)
else'if PosSection>0 Then
    'Section Not found. Add section data at the end of file contents.
    If Right(INIContents, 2) <> vbCrLf And Len(INIContents)>0 Then
        INIContents = INIContents & vbCrLf
    End If
    INIContents = INIContents & "[" & Section & "]" & vbCrLf & _
        KeyName & "=" & Value
end if'if PosSection>0 Then
WriteFile FileName, INIContents
End Sub

Function WriteFile(ByVal FileName, ByVal Contents)

    Dim FS: Set FS = CreateObject("Scripting.FileSystemObject")
    'On Error Resume Next

    'Go To windows folder If full path Not specified.
    If InStr(FileName, ":\") = 0 And Left (FileName,2)<>"\" Then
        FileName = FS.GetSpecialFolder(0) & "\" & FileName
    End If

    Dim OutStream: Set OutStream = FS.OpenTextFile(FileName, 2, True)
    OutStream.Write Contents
End Function

Function GetINIString(Section, KeyName, Default, FileName)
    Dim INIContents, PosSection, PosEndSection, sContents, Value, Found

    'Get contents of the INI file As a string
    INIContents = GetFile(FileName)

    'Find section
    PosSection = InStr(1, INIContents, "[" & Section & "]", vbTextCompare)
    If PosSection>0 Then
        'Section exists. Find end of section
        PosEndSection = InStr(PosSection, INIContents, vbCrLf & "[")
        '?Is this last section?
        If PosEndSection = 0 Then PosEndSection = Len(INIContents)+1

        'Separate section contents
        sContents = Mid(INIContents, PosSection, PosEndSection - PosSection)

        If InStr(1, sContents, vbCrLf & KeyName & "=", vbTextCompare)>0 Then

```

```

        Found = True
        'Separate value of a key.
        Value = SeparateField(sContents, vbCrLf & KeyName & "=", vbCrLf)
    End If
End If
If isempty(Found) Then Value = Default
GetINIString = Value
End Function

Function GetINISections(Section, FileName)
    Dim INIContents, PosSection, PosEndSection, sContents, Value()

    'Get contents of the INI file As a string
    INIContents = GetFile(FileName)

    'Find section
    PosSection = InStr(1, INIContents, "[" & Section & "]", vbTextCompare)
    If PosSection > 0 Then
        'Section exists. Find end of section
        PosEndSection = InStr(PosSection, INIContents, vbCrLf & "[")
        '?Is this last section?
        If PosEndSection = 0 Then
            PosEndSection = Len(INIContents)+1
        End If

        'Separate section contents
        sContents = Mid(INIContents, PosSection, PosEndSection - PosSection)
    End If
    GetINISections = Split(sContents, vbCrLf)
End Function

'Separates one field between sStart And sEnd
Function SeparateField(ByVal sFrom, ByVal sStart, ByVal sEnd)
    Dim PosB: PosB = InStr(1, sFrom, sStart, 1)
    If PosB > 0 Then
        PosB = PosB + Len(sStart)
        Dim PosE: PosE = InStr(PosB, sFrom, sEnd, 1)
        If PosE = 0 Then PosE = InStr(PosB, sFrom, vbCrLf, 1)
        If PosE = 0 Then PosE = Len(sFrom) + 1
        SeparateField = Mid(sFrom, PosB, PosE - PosB)
    End If
End Function

Function GetFile(ByVal FileName)
    Dim FS: Set FS = CreateObject("Scripting.FileSystemObject")
    'Go To windows folder If full path Not specified.
    If InStr(FileName, ":\") = 0 And Left (FileName,2)<>"\\\" Then
        FileName = FS.GetSpecialFolder(0) & "\" & FileName
    End If
    On Error Resume Next

    GetFile = FS.OpenTextFile(FileName).ReadAll
End Function

////////////////////////////////////
Sub Sort(inpArray(), inpList)
    Dim intRet
    Dim intCompare
    Dim intLoopTimes
    Dim strTemp

    For intLoopTimes = 1 To UBound(inpArray,1)
        For intCompare = LBound(inpArray,1) To UBound(inpArray,1) - 1
            intRet = StrComp(inpArray(intCompare,1), inpArray(intCompare + 1,1), vbTextCompare)
            If intRet = 1 Then
                ' String1 is > than String2
                ' Move Dimension 1 of inpArray
                strTemp = inpArray(intCompare,1)
                inpArray(intCompare,1) = inpArray(intCompare + 1,1)
                inpArray(intCompare + 1,1) = strTemp

                ' Move Dimension 2 of inpArray
                strTemp = inpArray(intCompare,2)
                inpArray(intCompare,2) = inpArray(intCompare + 1,2)
                inpArray(intCompare + 1,2) = strTemp
            End If
        Next
    Next

    Dim objOption
    For Each objOption in inpList.Options

```

[illegible]

```

<TD class=mainTableCellTopAlign colSpan=21 height=9></TD></TR>
<TR>
<TD class=mainTableCellTopAlign colSpan=21 height =2></TD></TR>
<TR>
<TD class=mainTableCellTopAlign height=14></TD>
<TD class=mainTableTwinStripes colSpan=19></TD>
<TD class=mainTableCellTopAlign></TD></TR>
<TR>
<TD class=mainTableCellTopAlign></TD>
<TD class=mainTableCellTopAlign colSpan=19>
  <IMG height=19 alt="" src="gfx/blank.gif" width=1 border=0></TD>
<TD class=mainTableCellTopAlign></TD></TR>
<TR>
<TD class=mainTableCellTopAlign></TD>
<TD class=mainTableCellTopAlign colSpan=3>
  &nbsp;</TD>
<TD class=mainTableCellTopAlign>&nbsp;</TD>
<TD class=mainTableContentCell colSpan=11>

  <div id="Main" style="visibility:hidden;">
    <table class="dataTable" width="100%">
      <tr class="bgColor2" bg="#CCCCC" fs="1">
        <td class="dataTableHeaderCell" colspan="2"><div class="mainContentSmall
mainContentBold">
          Setup Mode </div></td>
      </tr><tr>
        <td class="dataTableCellMiddle" align="left">
          <input type="radio" value="1" name="R1" onclick="R1_OnClick"></td>
        <td class="dataTableCellMiddle" style="text-align: left" width="95%"><div
id="SetupModel" class="mainContentSmall"><code></code>SetupModelLabel<code></code></div></td></tr><tr>
        <td class="dataTableCellMiddle" align="left">
          <input type="radio" value="2" name="R1" onclick="R1_OnClick"></td>
        <td class="dataTableCellMiddle" style="text-align: left" width="95%"><div
id="SetupMode2" class="mainContentSmall"><code></code>SetupMode2Label<code></code></div></td></tr><tr>
        <td class="dataTableCellMiddle" align="left">
          <input type="radio" value="3" name="R1" onclick="R1_OnClick"></td>
        <td class="dataTableCellMiddle" style="text-align: left" width="95%"><div
id="SetupMode3" class="mainContentSmall"><code></code>SetupMode3Label<code></code></div></td></tr>
      </tr>
      <tr bg="#FFAAAA">
        <td style="border-top: 1px solid #000000;" width="4%">&nbsp;</td>
        <td style="border-top: 1px solid #000000;" width="95%">&nbsp;</td>
      </tr>
    </table>

    <table class="dataTable" width="100%">
      <tr class="bgColor2" bg="#CCCCC" fs="1">
        <td class="dataTableHeaderCell" colspan="2"><div id="title1"
class="mainContentSmall mainContentBold">
          Disk Configuration</div></td>
      </tr><tr>
        <td class="dataTableCellMiddle" align="left"><div id="label1"
class="mainContentSmall">Harddisks / Partitions</div></td>
        <td class="dataTableCellMiddle" style="text-align: left" width="79%">
          <table border="0" width="100%" id="table2" cellspacing="0"
cellpadding="0">
            <tr>
              <td width="1%"><div id="OptionDiskDIV"
class="mainContentSmall">
                <select class="optionList" style="border: 1px solid #FFFFFF;" size="1"
name="OptionDisk" onChange="OptionDisk_OnChange">
                  </select></div></td>
              <td>
                <div id="OptionPartDIV" class="mainContentSmall">
                  <select class="optionList" style="border: 1px solid #FFFFFF;" size="1"
name="OptionPart" onChange="OptionPart_OnChange">
                    </select></div>
                </td>
            </tr>
          </table>
        </td>
      </tr><tr>
        <td class="dataTableCellMiddle" align="left"><div id="label2"
class="mainContentSmall">
          Partition Size</div></td>
        <td class="dataTableCellMiddle" width="79%">
          <table border="0" width="100%" id="table4" cellspacing="0"

```

```

cellpadding="0">
        <tr>
            <td width="1%">
                <select class="optionList" style="border: 1px solid #FFFFFF;" size="1"
name="OptionPartsSize">
                </select></td>
            <td>
                <div id="lblFormatOption" class="mainContentSmall" style="margin -
left: 2"></div>
            </td>
        </tr>
    </table>
</td>
</tr><tr bg="#FFAAAA">
    <td style="border-top: 1px solid #000000;" width="21%">&nbsp;</td>
    <td style="border-top: 1px solid #000000;" width="78%">&nbsp;</td>
</tr>
</table>

<table class="dataTable" width="100%">
    <tr class="bgColor2" bg="#CCCCCC" fs="1">
        <td class="dataTableHeaderCell" colspan="2"><div class="mainContentSmall
mainContentBold">
            Windows Setup</div></td>
        </tr><tr>
            <td class="dataTableCellMiddle" align="left"><div class="mainContentSmall"
id="lblAssetInformation">
                lblAssetInformation</div></td>
            <td class="dataTableCellMiddle" style="text-align: left" width="79%">
                <input type="checkbox" name="C1" value="0" style="margin-left: -4"></td>
        </tr><tr>
            <td class="dataTableCellMiddle" align="left"><div
class="mainContentSmall">Computer
                Name</div></td>
            <td class="dataTableCellMiddle" style="text-align: left" width="79%">
                <input type="text" class="inputField" name="txtComputername"
style="width:200px;"></td>
        </tr><tr>
            <td class="dataTableCellMiddle" align="left"><div class="mainContentSmall">
                Keyboard Layout</div></td>
            <td class="dataTableCellMiddle" style="text-align: left" width="79%">
                <select class="optionList" style="border: 1px solid #FFFFFF;" size="1"
name="OptionKeyboard">
                </select></td>
        </tr><tr>
            <td class="dataTableCellTop" align="left"><div id="lblEDUProfiles"
class="mainContentSmall">
                Educational Profiles</div></td>
            <td class="dataTableCellMiddle" style="text-align: left" width="79%">
                <div id="EDUProfiles" class="mainContentSmall">
                    (N/A)</div></td>
        </tr><tr bg="#FFAAAA">
            <td width="21%">&nbsp;</td>
            <td width="78%">&nbsp;</td>
        </tr>
        <tr bg="#FFAAAA">
            <td width="99%" colspan="2">
                <p align="right">
                </td>
            </tr>
        </table>

        <table class="dataTable" width="100%">
            <tr class="bgColor2" bg="#CCCCCC" fs="1">
                <td class="dataTableHeaderCell" colspan="3"><div class="mainContentSmall
mainContentBold">
                    Software</div></td>
                </tr><tr>
                    <td class="dataTableCellMiddle" align="left"><div id="label1"
class="mainContentSmall">
                        Installation method</div></td>
                    <td class="dataTableCellMiddle" style="text-align: left" width="3%">
                        <input type="radio" value="1" name="R2" style="margin-left: -4"></td>
                    <td class="dataTableCellMiddle" style="text-align: left" width="76%">
                        <div id="InstallMethod1" class="mainContentSmall"></div></td>
                </tr><tr>
                    <td class="dataTableCellMiddle" align="left">&nbsp;</td>
                    <td class="dataTableCellMiddle" style="text-align: left" width="3%">
                        <input type="radio" value="2" name="R2" style="margin-left: -4"></td>

```

```
 </div></td> </tr><tr>   |  Thomas Galliker Schindler Informatik AG  Seite 96 von 104  31.03.2006 |
```



```

CopyRightNotice.innerHTML = iniCopyRightNotice
InstallMethod1.innerHTML = iniMethod1Label
InstallMethod2.innerHTML = iniMethod2Label
Form1.R2(iniDefaultMethod-1).checked = True
Form1.txtComputername.value = iniComputer
</script>
</form>
</body>
</html>

```

10.2.3.2 Config.ini

```
[General]
Title=Welcome to Setup Assistant
Subtitle=This wizard will guide you through the setup process
SetupModelLabel=<u>New</u>Installation
SetupMode2Label=<u>Re</u>Installation
SetupMode3Label=Clean Up
CopyRightNotice=©2006 Schindler Informatik Ltd.

[Unity]
ConfigurationFolder=Config
GroupsInformation=Config\UnityGroups.ini
SystemInformation=Config\UnitySystem.ini
OSSource=Images\dsksetup\winxpsp2\I386
UnattendFile=Images\dsksetup\winxpsp2\unattend.txt
SoftwareOU=EDU_INF

[Software]
DefaultMethod=1
Method1Label=Remove unused software before installing new software
Method2Label=Completely remove <u>all</u> existing software before installing new software

[Profiles]
EDUUsers=EDUINFCE;EDUINFDA;EDUINFFA;EDUINFJA;EDUINFILA;EDUINFLU;EDUINFMA;EDUINFMI;EDUINFSA
LocalProfilesPath=Documents and Settings
ServerProfilesPath=\\infv0001

[Keyboard]
DefaultKeyboard=Swiss German

[ActiveDirectory]
DefaultOU=2
OU1Path=Test\Workstations
OU2Path=Prod\Workstations

[PartSizeChoice1]
MinDiskSpace=0
MaxDiskSpace=30000
SysPartSize=10000
Text=10 GB

[PartSizeChoice2]
MinDiskSpace=30001
MaxDiskSpace=60000
SysPartSize=12000
Text=12 GB

[PartSizeChoice3]
MinDiskSpace=60001
MaxDiskSpace=100000
SysPartSize=15000
Text=15 GB

[PartSizeChoice4]
MinDiskSpace=100001
MaxDiskSpace=999999
SysPartSize=20000
Text=20 GB

[DiskFormat]
QuickFormat=Yes
DefaultDiskLabel1=System
DefaultDiskLabel2=Data

[InputValidation]
Message1=No disk selected! Please select a disk.
Message2=Please enter a Computer Name.
Message3=Computer Name must contain exactly 8 character.
```

10.3 Anhang C – RAMDisk Konfigurationsdatei

Tabelle 29: RAMDisk Konfigurationsdatei "ramdisk.inf"

[Version]
Signature="\$WINDOWS NT\$"
[AddReg]
HKLM,"ControlSet001\Control\Class\{FFA1C341-4539-11D3-B88D-00C04FAD5172}", "Class", 0000000000, "RamDisk"
HKLM,"ControlSet001\Control\Class\{FFA1C341-4539-11D3-B88D-00C04FAD5172}", " ", 0000000000, "RAMDisk"
HKLM,"ControlSet001\Control\Class\{FFA1C341-4539-11D3-B88D-00C04FAD5172}\0001", "ProviderName", 0000000000, "QSoft"
HKLM,"ControlSet001\Control\Class\{FFA1C341-4539-11D3-B88D-00C04FAD5172}\0001", "MatchingDeviceId", 0000000000, "ramdriv"
HKLM,"ControlSet001\Control\Class\{FFA1C341-4539-11D3-B88D-00C04FAD5172}\0001", "DriverDesc", 0000000000, "Ramdisk [QSoft]"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "ClassGUID", 0000000000, "{FFA1C341-4539-11D3-B88D-00C04FAD5172}"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "Class", 0000000000, "RamDisk"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "ConfigFlags", 0x00010001, 0x4
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "Mfg", 0000000000, "QSoft"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "HardwareID", 0000000000, "ramdriv"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "Service", 0000000000, "Ramdriv"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "DeviceDesc", 0000000000, "Ramdisk [QSoft]"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "Capabilities", 0x00000000
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999", "Driver", 0000000000, "{FFA1C341-4539-11D3-B88D-00C04FAD5172}\0001"
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999\Control", "DisableCount", 0x00010001, 0x0
HKLM,"ControlSet001\Enum\Root\UNKNOWN\9999\Control", 0000000000, "ActiveService", "Ramdriv"
HKLM,"ControlSet001\Services\Ramdriv", "Type", 0x00010001, 0x1
HKLM,"ControlSet001\Services\Ramdriv", "Start", 0x00010001, 0x3
HKLM,"ControlSet001\Services\Ramdriv", "ErrorControl", 0x00010001, 0x1
HKLM,"ControlSet001\Services\Ramdriv", "DisplayName", 0000000000, "Ramdisk [QSoft]"
HKLM,"ControlSet001\Services\Ramdriv", "ImagePath", 0x00020000, "system32\drivers\ramdrv.sys"
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "BreakOnEntry", 0x00010001, 0x0
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "DebugLevel", 0x00010001, 0x5
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "DebugComp", 0x00010001, 0xffffffff
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "DiskSize", 0x00010001, 0x02000000
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "DriveLetter", 0000000000, "Z:"
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "RootDirEntries", 0x00010001, 0x200
HKLM,"ControlSet001\Services\Ramdriv\Parameters", "SectorsPerCluster", 0x00010001, 0x2
HKLM,"ControlSet001\Services\Ramdriv\Enum", "0", 0000000000, "Root\UNKNOWN\9999"
HKLM,"ControlSet001\Services\Ramdriv\Enum", "Count", 0x00010001, 0x1
HKLM,"ControlSet001\Services\Ramdriv\Enum", "NextInstance", 0x00010001, 0x1

Zur Konfiguration der Diskgrösse dieser RAMDisk können muss der hexadezimale Zahlenwert des RegistryEintrags "DiskSize" angepasst werden. Nachfolgende Tabelle zeigt die möglichen Diskgrössen und die jeweils dazugehörigen hexadezimalen Werte. Der markierte Eintag referenziert mit dem in der letzteren Tabelle markierten Registry Wert.

Tabelle 30: Hexadezimale "DiskSize"-Werte für RAMDisk.

Registry Wert	RAMDisk Grösse
0x00400000	4 MByte
0x00800000	8 MByte
0x01000000	16 MByte
0x02000000	32 MByte
0x04000000	64 MByte

10.4 Anhang D – Ist-Zeitplan

<Diese Seite wird durch den Ausdruck von MS Project Ist-Zeitplan ersetzt>


10.5 Anhang E – Detailliertes Ablaufdiagramm

<Diese Seite wird durch den Ausdruck von MS Visio Ablaufdiagramm ersetzt>




10.6 Anhang F – OPK Hilfe

Datei	Beschreibung
 WinPE Hilfe.zip	Windows Preinstallation Environment User's Guide.


10.7 Anhang G – Netzwerkanalyse

Datei	Beschreibung
 AuthCapture.cap	Netzwerkanalyse mit aufgezeichneter Kerberos-Authentifizierung.

10.8 Anhang H – Hypertext Anwendungen

Datei	Beschreibung
 Htlogin.zip	Hypertext Application "Login Form".
 Htsetup.zip	Hypertext Application "Setup Assistant".
 htaFactory.zip	Hypertext Application "Factory".

10.9 Anhang I – Präsentation

Datei	Beschreibung
 Präsentation.zip	PowerPoint Präsentation inkl. Live Demonstration.

10.10 Anhang J – Quellenverzeichnis

Diese Dokumentation ist mit Hilfe verschiedener Informationsquellen entstanden. Aus urheberrechtlichen Gründen – nicht zuletzt auch aus Anstandsgründen – werden in den nachfolgenden Tabellen sämtliche Quellen illustriert, welche im Zusammenhang mit der Entstehung dieser Projektarbeit eine Rolle spielten.

10.10.1 Generelle Quellen

Nummer	Beschreibung
[Quelle 1.01]	Quelle: LehlingsDB  Lehlingsdokumentationen Autor: Thomas Galliker Datum: 30.03.2006
[Quelle 1.02]	Quelle: http://www.qualitysoftware.tk http://www.ramdisk.tk Autor: Qsoft Datum: 30.03.2006
[Quelle 1.03]	Quelle: http://unattended.msfn.org/unattended.xp Autor: MSFN, Microsoft Software Forum Network; Eigentümer des Inhalts dieser Subdomain: Mike Wilson Datum: 30.03.2006
[Quelle 1.04]	Quelle: http://de.wikipedia.org/wiki/Softwaretest Autor: Enzyklopädie Wikipedia Datum: 30.03.2006
[Quelle 1.05]	Quelle: OPK Hilfe zu Windows PE (WinPE.chm), siehe Anhang E . Autor: Microsoft Corporation Datum: 30.03.2006

[Quelle 1.06]	Quelle: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp
	Autor: Microsoft Corporation; WMI Referenz
	Datum: 30.03.2006
[Quelle 1.07]	Quelle: http://www.im.iwi.unibe.ch/lehre/qualifikationsarbeiten/resource/data/LeitfadenWissArbeit_IM.doc
	Autor: Universität Bern, Institut für Wirtschaftsinformatik
	Datum: 30.03.2006
[Quelle 1.08]	Quelle: http://www.unitysite.com
	Autor: Unity Website
	Datum: 30.03.2006
[Quelle 1.09]	Quelle: http://www.ibm.com
	Autor: IBM Lenovo; Titelbild
	Datum: 30.03.2006

10.10.2 Referenzierte Quellen

Nummer	Beschreibung	Seite
[Quelle 2.01]	Quelle: http://www.nu2.nu/pebuilder/commercial	
	Autor: Nu2 Productions, Bart Lagerweij	
	Datum: 30.03.2006	
[Quelle 2.02]	Quelle: http://www.cts-bv.nl/pebuilder	
	Autor: CTS, Computer Technology Systems B.V.	
	Datum: 30.03.2006	
[Quelle 2.03]	Quelle: http://support.microsoft.com/kb/q257405	
	Autor: Microsoft Corporation; Ramdisk.sys sample driver for Windows 2000	
	Datum: 30.03.2006	
[Quelle 2.04]	Quelle: http://home.tiscali.be/ir006712/RAMDisk/ramdiskfree.htm	
	Autor: Qsoft RAMDisk; Freeware Version	
	Datum: 30.03.2006	
[Quelle 2.05]	Quelle: http://home.tiscali.be/ir006712/RAMDisk/ramdiskpro.htm	
	Autor: Qsoft RAMDisk; Pro/Extended Version	
	Datum: 30.03.2006	
[Quelle 2.06]	Quelle: http://www.microsoft.com/germany/technet/prodtechnol/windowsvista/expert/ximage.msp	
	Autor: Microsoft Corporation; Technet	
	Datum: 30.03.2006	
[Quelle 2.07]	Quelle: http://www.microsoft.com/germany/technet/datenbank/articles/600953.msp	
	Autor: Microsoft Corporation; Technet	
	Datum: 30.03.2006	
[Quelle 2.08]	Quelle: http://www.msfm.org/board/lofiversion/index.php/t46845.html	
	Autor: MSDN, Microsoft Software Forum Network; "writeable ramdisk for winpe2005"	
	Datum: 30.03.2006	
[Quelle 2.09]	Quelle: http://www.laboratoire-microsoft.org/articles/win/WinPE/1	
	Autor: Kazer, MGI Consultants, Architecte Systèmes	
	Datum: 30.03.2006	
[Quelle 2.10]	Quelle: http://support.microsoft.com/default.aspx?scid=kb;en-us;139177	
	Autor: Microsoft Corporation	
	Datum: 30.03.2006	
[Quelle 2.11]	Quelle: http://mystuff.clarke.co.nz/ShutdownPE.asp http://sideload.clarke.co.nz/download/ShutdownPE1.zip	
	Autor: David Clarke	
	Datum: 30.03.2006	
[Quelle 2.12]	Quelle: http://www.microsoft.com/licensing/programs/sa/support/winpe.msp	
	Autor: Microsoft Corporation; Windows PE	
	Datum: 30.03.2006	
[Quelle 2.13]	Quelle: http://www.microsoft.com/ireland/licensing/sa/sadsoc.asp	
	Autor: Microsoft Corporation; Software Assurance	

	Datum:	30.03.2006	
[Quelle 2.14]	Quelle:	http://de.wikipedia.org/wiki/Microsoft_Windows_PE	
	Autor:	Enzyklopädie Wikipedia; Microsoft Windows PE	
	Datum:	30.03.2006	
[Quelle 2.15]	Quelle:	http://www.nu2.nu/pebuilder	
	Autor:	Nu2 Productions, Bart Lagerweij	
	Datum:	30.03.2006	
[Quelle 2.16]	Quelle:	http://www-307.ibm.com/pc/support/site.wss/document.do?sitestyle=lenovo&Indo cid=MIGR-58618 http://www-307.ibm.com/pc/support/site.wss/license.do?filename=mobiles/77ra05ww.exe	
	Autor:	IBM Lenovo; Support & Downloads	
	Datum:	30.03.2006	
[Quelle 2.17]	Quelle:	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndude/html/dude02262001.asp	
	Autor:	Microsoft Developer Network, MSDN; „Updating the Display During Lengthy Operations“	
	Datum:	30.03.2006	
[Quelle 2.18]	Quelle:	http://visualbasicscript.com/m_24421/mpage_1/key_tm.htm	
	Autor:	VisualBasicScript Forum; „Edit INI via VBS“	
	Datum:	30.03.2006	
[Quelle 2.19]	Quelle:	http://de.wikipedia.org/wiki/Integrationstest	
	Autor:	Enzyklopädie Wikipedia; Integrationstest	
	Datum:	30.03.2006	
[Quelle 2.20]	Quelle:	http://de.wikipedia.org/wiki/Komponententest	
	Autor:	Enzyklopädie Wikipedia; Komponententest	
	Datum:	30.03.2006	
[Quelle 2.21]	Quelle:	http://www.microsoft.com/technet/scriptcenter/hubs/htas.msp	
	Autor:	Microsoft Technet; HTA Developers Center	
	Datum:	30.03.2006	
[Quelle 2.22]	Quelle:	http://msdn.microsoft.com/library/default.asp?url=/workshop/author/hta/overview/htaoverview.asp	
	Autor:	Microsoft Developer Network, MSDN; „Introduction to HTML“	
	Datum:	30.03.2006	
[Quelle 2.23]	Quelle:	http://www.microsoft.com/downloads/details.aspx?FamilyId=231D8143-F21B-4707-B583-AE7B9152E6D9&displaylang=en	
	Autor:	Microsoft HTA Helpomatic 1.01	
	Datum:	30.03.2006	
[Quelle 2.24]	Quelle:	http://www.microsoft.com/downloads/details.aspx?familyid=09DFC342-648B-4119-B7EB-783B0F7D1178&displaylang=en	
	Autor:	Microsoft Scriptomatic 2.0	
	Datum:	30.03.2006	
[Quelle 2.25]	Quelle:	http://de.wikipedia.org/wiki/Registry	
	Autor:	Enzyklopädie Wikipedia; Windows Registrierung	
	Datum:	30.03.2006	
[Quelle 2.26]	Quelle:	http://de.wikipedia.org/wiki/Slipstreaming	
	Autor:	Enzyklopädie Wikipedia; Slipstreaming	
	Datum:	30.03.2006	
[Quelle 2.27]	Quelle:	http://de.wikipedia.org/wiki/Ramdisk	
	Autor:	Enzyklopädie Wikipedia; RAMDisk	
	Datum:	30.03.2006	
[Quelle 2.28]	Quelle:	http://de.wikipedia.org/wiki/Systemtest	
	Autor:	Enzyklopädie Wikipedia; Systemtest	
	Datum:	30.03.2006	
[Quelle 2.29]	Quelle:	http://de.wikipedia.org/wiki/Prototyping_%28Softwareentwicklung%29	
	Autor:	Enzyklopädie Wikipedia; Prototyping-Verfahren	
	Datum:	30.03.2006	

[Quelle 2.30]	Quelle:	http://de.wikipedia.org/wiki/Original_Equipment_Manufacturer	
	Autor:	Enzyklopädie Wikipedia; OEM	
	Datum:	30.03.2006	
[Quelle 2.31]	Quelle:	http://de.wikipedia.org/wiki/VBScript	
	Autor:	Enzyklopädie Wikipedia; VBScript	
	Datum:	30.03.2006	
[Quelle 2.32]	Quelle:	http://de.wikipedia.org/wiki/Video_Graphics_Array	
	Autor:	Enzyklopädie Wikipedia; Video Graphics Array	
	Datum:	30.03.2006	
[Quelle 2.33]	Quelle:	http://de.wikipedia.org/wiki/Windows_Management_Instrumentations	
	Autor:	Enzyklopädie Wikipedia; Windows Management Instrumentations	
	Datum:	30.03.2006	
[Quelle 2.34]	Quelle:	http://de.wikipedia.org/wiki/Windows_Script_Host	
	Autor:	Enzyklopädie Wikipedia; Windows Script Host	
	Datum:	30.03.2006	
[Quelle 2.35]	Quelle:	http://de.wikipedia.org/wiki/ActiveX_Data_Objects	
	Autor:	Enzyklopädie Wikipedia; ActiveX Data Objects	
	Datum:	30.03.2006	
[Quelle 2.36]	Quelle:	http://de.wikipedia.org/wiki/Debugging	
	Autor:	Enzyklopädie Wikipedia; Debugging	
	Datum:	30.03.2006	
[Quelle 2.37]	Quelle:	http://www.networkchemistry.com/products/packetyzer.php	
	Autor:	Network Chemistry; Packetyzer	
	Datum:	30.03.2006	