

Hochschule Magdeburg-Stendal
Fachbereich Ingenieurwissenschaften und Industriedesign (IWID)
Institut für Elektrotechnik

Bachelorarbeit

zur Erlangung des Grades eines "Bachelor of Engineering"
im Studiengang Elektrotechnik

**Thema: "Modernisierung von Steuerungssystem von Simatic S5
auf Simatic S7 der VSG1 Anlagen"**

Eingereicht von: Ngoc Anh Nguyen

Angefertigt für: Euroglas GmbH, Haldensleben

Matrikel: E 2009

Ausgabetermin: 30. Januar 2013

Abgabetermin: 10. April 2013

Schulischer Betreuer: Herr Prof. Dr.-Ing. Yongjian Ding

Betrieblicher Betreuer: Herr Sascha Schöne

Aufgabenstellung des Projekts

Mit Bekanntgabe der Abkündigungsdaten für die SIMATIC S5 Systeme wird es immer schwieriger und kostenintensiver, Service- und Supportangebote für die Systeme zu erhalten, Ersatzteile zu beschaffen und damit Anlagen zu erweitern. Somit sind auch erforderliche Qualifikationen für das Service- und Betriebspersonal sehr schwer aufrecht zu erhalten. Zudem ist das SIMATIC S5 System in seinem Funktionsumfang eingeschränkt und lässt erforderliche technische Ergänzungen zum Teil nicht zu.

Diesem Problem soll durch die Migration in ein SIMATIC S7 System entgegengewirkt werden. Die Vorteile des SIMATIC S7 Systems sind die hohe Wirtschaftlichkeit, die durch die anforderungsgerechte Funktionalität und Performance sowie Investitionssicherheit durch das Konzept der Totally Integrated Automation erreicht wird.

Die Aufgabestellung des Projekts ist die Modernisierung der VSG1 Steuerung von Simatic S5 auf Simatic S7. Die VSG1 Anlage ist eine Teilanlage zur Produktion von Verbundsicherheitsglas der Firma Euroglas GmbH in Haldensleben. Die Aufgabe der VSG1 Anlage besteht darin, Glasscheiben mit verschiedenen Größen und Dicken mithilfe eines Portalroboters aus Gestellen zu entnehmen und sie an den Kipptisch mit Rollenbeförderern für den Weitertransport und weitere Verarbeitungsschritte zu übergeben.

Im Zuge der Systemumstellung von S5 auf S7 sind sechs Steuerungssysteme nacheinander umzustellen. Es handelt sich dabei um die Linien VSG 1 bis 3 und MAG 1 bis 3. Das vorhandene Programm ist zu analysieren und das neue S7-Programm ist strukturiert zu erstellen. Bei der Migration, die eine Untermenge der Modernisierung beschreibt, erfolgt der schrittweise oder komplette Austausch installierter Komponenten oder Systeme „S5“ durch ein Nachfolgesystem „S7“. Dabei kommt es darauf an, in den Altanlagen enthaltenes Anlagenengineering zu bewahren. Das erhaltene Anlagenengineering wird im Anschluss ganz oder teilweise in das Engineering des neuen Systems überführt. Dabei verfolgt man das Ziel, die teilweise hohen getätigten Investitionen in das Engineering zu sichern.

Ehrenwörtliche Erklärung

Ich versichere, hiermit dass ich diese Abschlussarbeit ohne fremde Hilfe selbstständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

.....
Ort, Datum

.....
Unterschrift

Danksagung

Vielen Dank an alle, die mich während der Bearbeitung meines Bachelorthemas unterstützt haben.

Besonderer Dank gilt meinen Betriebsbetreuern Herrn Sascha Schöne und den Herrn Matthias Dettmer von der Firma Euroglas in Haldensleben für die Bereitstellung des Themas und deren Unterstützung während dieser Zeit. Ich bedanke mich ebenso bei meinem Hochschulbetreuer Prof. Ding von dem HS Magdeburg für die vielen Ratschläge.

Inhaltverzeichnis

1	Einleitung	1
2	Siemens Simatic	2
2.1	Einführung Simatic	2
2.2	Generationswechsel S5 auf S7.....	5
2.3	Migrationsmöglichkeiten	8
3	Allgemeine Anlagenbeschreibung	10
3.1	Achsenbeschreibung	12
3.2	Anlagenübersicht - Istzustand.....	13
3.3	Anlageübersicht - Sollzustand	14
3.4	Baugruppen Istzustand.....	15
3.5	Baugruppen Sollzustand nach Modernisierung	17
3.6	Hardwarestruktur	18
4	Projektmanagement.....	19
4.1	Einführung	20
4.2	Projektstrukturplan	21
4.3	Projektcontrolling	24
4.4	Risikomanagement	29
4.5	Dokumentation	35
4.6	Zusammenfassung Projektdurchführung	37
4.6.1	Analyse der Projektdurchführung	39
5	Software Überprüfung.....	42
5.1	Einführung	42
5.2	Allgemeine Softwaretestverfahren	43

5.3	Nomenklatur	44
5.3.1	IEEE Standard.....	44
5.3.2	ISO IEC 29119.....	45
5.4	Teststrategie	48
5.5	Testverfahren: Black-Box-Test und White-Box-Test	50
5.6	Teststufen bzw. Testzyklen	52
5.7	Testverlauf	54
5.7.1	Software Spezifikation des Projekts	55
5.7.2	Softwarestruktur.....	56
5.7.3	Programmtest	58
5.8	Zusammenfassung	63
6	Projektauswertung.....	64
6.1	Ergebnis	64
6.2	Defizite	64
7	Optimierungsmöglichkeiten.....	66
7.1	Riss -Erkennungssensor auf Rollenförderer +022.....	66
7.2	Intelligente Positionierung der Portalroboter und Kipptisch beim Übergabe ..	70
7.3	Vollständige Automatisierung.....	71
7.4	Mit „Blick in die Zukunft“ – vollständige Vernetzung der Anlage	72
7.4.1	Allgemein Busstruktur/Bustopologie	72
7.4.2	Anlagenbusstruktur	73
8	Literaturverzeichnis.....	76
9	Internetverzeichnis.....	77
10	Anhang	78

Abbildungsverzeichnis

Abbildung 2-1 Produktkündigung.....	5
Abbildung 2-2 Simatic Historie	6
Abbildung 2-3 Beispiel S5 Peripherie ET200.....	8
Abbildung 2-4 Beispiel Adapter Fronstecker S7-400.....	8
Abbildung 2-5 Beispiel Adapter Fronstecker S7-300.....	9
Abbildung 3-1 Achsenansicht Portalroboter (X-/Y-/Z-Achse).....	10
Abbildung 3-2 Achsenansicht Kipptisch (K-/L-Achse).....	11
Abbildung 3-3 Übersicht VSG1 Anlage-Istzustand.....	13
Abbildung 3-4 Übersicht VSG1 Anlage - Sollzustand	14
Abbildung 3-5 genutzte S5 Baugruppen (Schaltschrank Steuerung VSG1)	15
Abbildung 3-6 S7-Baugruppen	17
Abbildung 3-7 S5 Hardwarestruktur.....	18
Abbildung 4-1 Beispielmmodell eines PSP als Baumdiagramm bezogen auf durchgeführte Projekt.....	22
Abbildung 4-2 Kernbereich des Projektcontrolling	24
Abbildung 4-3 Beispiel Checklisteunterteilung/Meilensteine	28
Abbildung 4-4 Risikoklassifizierung	29
Abbildung 4-5 Mögliche Risiken des Projekts	31
Abbildung 4-6 Projektstruktur "Soll" Form.....	39
Abbildung 4-7 Projektstruktur "IST" Form	41
Abbildung 5-1 Programmstruktur-Beispiel.....	49

Abbildung 5-2 V-Modell.....	52
Abbildung 5-3 durchgeführte Testablauf.....	54
Abbildung 5-4 Programmstruktur aufgerufene Modulen Stand 12011_17.....	56
Abbildung 5-5 Beispiel WinCC Anlagenübersicht Version 12011_17.....	61
Abbildung 5-6 Beispiel WinCC Y-Achse Darstellung Version 12011_17.....	62
Abbildung 7-1 Schrittkette VSG1 bei Riss auf Glasplatte.....	66
Abbildung 7-2 Schrittkette VSG1 Optimierungsversuch 01.....	67
Abbildung 7-3 Schrittkette VSG1 Optimierungsversuch 02.....	68
Abbildung 7-4 Netzwerktopologie.....	72
Abbildung 7-5 derzeitige Busstruktur.....	74

Tabellenverzeichnis

Tabelle 1 Automatisierungspyramide	3
Tabelle 2 Konvertierungsrate	9
Tabelle 3 S5-Teileliste	16
Tabelle 4 S7-Teileliste	17
Tabelle 5 Möglichkeiten der Datengewinnung für das Projektcontrolling.....	26
Tabelle 6 Bewertungschema zwischen Schadenausmaß und Eintrittswahrscheinlichkeit.....	32
Tabelle 7 Vergleich Black-Box-Test und White Box-Test.....	51
Tabelle 8 Modulbeschreibung, Stand 12011_17.....	57
Tabelle 9 Vorteil und Nachteil der Optimierungsversuche	69

Abkürzungsverzeichnis

Abkürzung	Bedeutung
AWL	Anweisungsliste
E/A	Eingang/Ausgang
BMW	Bayerische Motoren Werke
DP	Dezentrale Peripherie
ET	Extension Terminal
evtl.	eventuell
Etc.	Et cetera (und so weiter)
FUP	Funktionsplan
HMI	Human Machine Interface
HW	Hardware
IEEE	Institute of Electrical and Electronic Engineers
KOP	Kontaktplan
MES	Manufacturing Execution System
OEM	Original-Equipment-Manufacturer
PSP	Projektstrukturplan
RFID	Radio-Frequency Identification
SPS	Speicherprogrammierbare Steuerung
SQAP	Software Quality Assurance Plan
SCMP	Software Configuration Management Plan
STD	Software Test Documentation
SRS	Software Requirements Specification
SUT	Standard for Software Unit Testing
SVVP	Software Validation & Verification Plan
SDD	Software Design Description
SPMP	Software Project Management Plan
SW	Software
UV1	Unterverteiler 1
UV2	Unterverteiler 2
UV3	Unterverteiler 3
UV4	Unterverteiler 4
UV5	Unterverteiler 5
TIA	Totally Integrated Automation
VPS	verbindungsprogrammierte Steuerung
VSG	Verbundsicherheitsglas

1 Einleitung

Euroglas¹ ist ein Flachglashersteller des europäischen Mittelstandes und ein Tochterunternehmen der Glas Trösch Holding AG aus der Schweiz. In Kooperation mit den Partnern Josef Nowak Glas GmbH & Co. KG, Sencoglas GmbH, Glaswerke Arnold GmbH & Co. KG, Press-Glas SA und Joh. Sprinz GmbH & Co. KG, betreibt Euroglas an vier Standorten Floatglaswerke (in Deutschland, Polen und Frankreich). Neben der Produktion von Floatglas, extra weißem Floatglas, Verbundsicherheitsglas, beschichtetem Wärme- und Sonnenschutzglas wird seit 2009 auch Glas für die Dünnschichtphotovoltaik gefertigt.

Die Bachelorarbeit wird im Rahmen eines Umbauprojekts angefertigt. Es geht dabei um die Steuerungsmodernisierung von S5 auf S7 der Teilanlagen Verbundsicherheitsglas (VSG1) der Euroglas GmbH in Haldensleben. Die Aufgabenstellung umfasst dabei die Projektplanung, die Software-Überprüfung, die Planung der Projektdurchführung sowie die Projektdokumentation. Diesbezüglich wird das S7 Programm von der Firma Symacon, mit Firmensitz in Barleben, programmiert und es erfolgt eine zusätzliche Überprüfung unsererseits hinsichtlich der notwendigen bzw. gewünschten Spezifikation.

Was sind die Ziele des Projekts?

Bei der Durchführung des Projekts sollten alle Arbeiten möglichst so durchgeführt werden, dass der jetzige Zustand mit minimalen Anforderungen wiederhergestellt werden kann. Aufgrund der Tatsache, dass dieser Umbau als Grundgerüst für die Modernisierung weiterer Teilanlagen dient, ist mit einem größeren Programmieraufwand zu rechnen. Unter Grundgerüst ist zu verstehen, dass die gesamte Programmierung modular gestaltet wird und somit eine verhältnismäßig einfache Integration in weiteren Anlagenabschnitten gegeben sein soll.

Dabei sollte die Software von Symacon komplett erneuert und auf unser Konzept einer modularen Programmierung angepasst werden. Eine genauere Beschreibung zu diesem Konzept wird in den folgenden Kapiteln oder im Anhang Lastenheft zu finden sein.

¹für weitere Information zu Euroglas GmbH siehe <http://www.euroglas.com/unternehmen/uebersicht-unternehmen-euroglas.html> oder Anhang_07_Euroglas_Imagebroschüre

2 Siemens Simatic

2.1 Einführung Simatic

In den Anfängen der programmierbaren Steuerung gab es die Simatic in der Version G mit Germanium-Einzelhalbleitern. Sie kam am 2. April 1958 als verbindungsprogrammierte Steuerung (VPS) Simatic auf den Markt. Am selben Tag wurde der Name Simatic beim Deutschen Patentamt als Warenzeichen der Firma Siemens-Schuckertwerke eingetragen. Simatic verbindet „Siemens“ und „Automatic“ als Kofferwort miteinander. Der Begriff wird für Produkte in der Automatisierungstechnik, der Leittechnik und der Manufacturing-Execution-Ebene genutzt.

Dabei besitzt der VPS verschiedene Modelle, die in den folgenden Jahren immer weiter entwickelt und verbessert wurden. Die Modelle N(1964) mit Silicium-Einzelhalbleitern, H, P und C(1973) als integrierte Schaltung folgten innerhalb der kommenden 15 Jahre. Der Übergang von der umständlichen VPS zur wesentlich effizienteren speicherprogrammierbaren Steuerung (SPS) gelang 1973 mit der Simatic S3 mit bescheidenem Erfolg. Im Jahr 1979 kam die Simatic S5 mit großem Erfolg auf den Markt. Bis 1995 wurde sie ständig weiterentwickelt und ist bis 2015 bei Siemens teilweise immer noch verfügbar. Zunächst wurden alle drei Darstellungsarten für das Anwenderprogramm konsequent angeboten: Anweisungsliste (AWL), Funktionsplan (FUP) und Kontaktplan (KOP). 1994 erfolgte die Einführung der heute aktuellen und ständig weiterentwickelten Simatic S7.

Sowohl die S5, als auch die S7 wird nach dem Konzept der Totally Integrated Automation (TIA) entwickelt. Dieses Konzept definiert das Zusammenspiel diverser automatisierungstechnischer Einzelkomponenten, Software-Tools und dazugehöriger Services zu einer durchgängigen Automatisierungslösung. Das Ziel des Konzepts der TIA besteht darin, ein automatisiertes System zu entwickeln bzw. das vorhandene System zu optimieren, die ohne äußere Einflüsse ordnungsgemäß ablaufen. Dabei unterteilen sie sich in vollautomatisiertes System, halbautomatisiertes System und manuelle Steuerung. Das einzelne System wird durch ihren dazugehörigen Automatisierungsgrad charakterisiert. Je höher der Automatisierungsgrad, desto niedriger ist der Anteil von äußeren Einflüssen, der benötigt wird, damit ein System einwandfrei funktionieren kann. Unter äußeren Einflüssen versteht man dabei den sogenannten Faktor Mensch.

Zu beachten ist es, dass die Instandhaltung und die Wartungsarbeit sowie die Reparatur von Verschleiß, die zwangsweise von Menschenhand durchgeführt werden, nicht unter dem Faktor Mensch für automatisiertes System berücksichtigt werden.

Es ist zu vermerken, dass die VSG1 Anlagen kein vollautomatisiertes System sind, sondern „nur“ ein halbautomatisiertes System. Vollautomatisiertes System bedeutet, dass die Anlage ohne menschliche Hilfe komplett funktionstüchtig ist. Somit spielt der Faktor Mensch keine Rolle. Unter einem halbautomatisiertem System hingegen versteht man, dass die Anlage teilweise äußere Einflüsse benötigt, um komplett funktionstüchtig zu sein.

Vermerk: Der Mensch ist ein unberechenbarer Faktor für ein automatisiertes System, der positiv oder aber auch negativ ausfallen kann.

Zusätzlich ermöglicht das Konzept der TIA die Durchgängigkeit über die vier Automatisierungsebenen der Automatisierungspyramide:

Ebene	Eingesetzte Systeme	Typische Aufgaben
Unternehmensebene	ERP	Produktionsgrobplanung, Bestellabwicklung
Betriebsleitebene	MES, MIS, LIMS	Produktionsfeinplanung, Produktionsdatenerfassung, KPI- Ermittlung; Material-Management, Qualitätsmanagement
(Prozess)-leitebene	Prozessleitsystem/HMI/SCADA	Bedienen und Beobachten, Rezeptverwaltung und Ausführung, Messwert Archivierung
Steuerungsebene	SPS	Steuerung, Regelung
Feldebene	Prozesssignale, Ein- /Ausgabemodule, Feldbluse	Schnittstelle zum technischen Produktionsprozess über Ein- und Ausgangssignale
Sensor-/Aktorebene	Parallelverdrahtung oder intelligente Systeme, wie z.B.: AS-Interface	Einfache und schnelle Datensammlung, meist binärer Signale

Tabelle 1 Automatisierungspyramide

Hierbei ist zu beachten, dass die oberen oder unteren Ebenen teilweise entfallen können, falls diese im jeweiligen Kontext unwichtig sind. Manchmal wird die Unternehmensebene und Betriebsleitebene als Managementebene zusammengefasst. Die Leitebene hat in bestimmten Branchen spezielle Namen, z. B. Prozessleitebene, Verkehrsleitebene, Gebäudeleitebene. Manche Autoren verstehen unter Leitebene die Zusammenfassung von Prozessleitebene und MES. Teilweise werden zusätzliche Ebenen aufgeführt, wie Zellenebene zwischen Steuerungsebene und Leitebene. Die Steuerungsebene wird auch als Automationsebene, Automatisierungsebene oder Prozessebene bezeichnet. Die Feldebene gliedert sich in Ein-/Ausgabebene bzw. Sensor-/Aktorebene.

Die Durchgängigkeit von TIA bietet den beteiligten Unternehmen in der Wertschöpfungskette eine Vereinfachung und Kosteneinsparung. Unter Wertschöpfungskette versteht man Original-Equipment-Manufacturer(OEM), Systemintegratoren, Planer und Endkunde.

Die Steuerungsebene wird im Rahmen der Projektdurchführung umgestaltet. Andere Ebenen der Automatisierungspyramiden bleiben vorerst weitestgehend unverändert. Dabei bleibt der Automatisierungsgrad des Systems im eigentlichen Sinne gleich, die Zuverlässigkeit und Verfügbarkeit des Systems bezogen auf ihren Software sollte durch die Modernisierung verbessert bzw. die Effektivität und Produktivität der Anlageerhöht werden.

2.2 Generationswechsel S5 auf S7



Was sind die Gründe von Step 5 auf Step 7 umzusteigen?

Aufgrund der Tatsache, dass in baldiger Zukunft keine S5 Ersatzteilkomponenten mehr verfügbar sein werden bzw. die Anforderungen an den Leistungsumfang der Anlagensteuerungen stetig steigen, ist eine Umrüstung unumgänglich.

Wussten Sie, dass im Jahr 2015 die S5-U Produkte 30 Jahre und die SINUMERIK 31 Jahre alt werden?

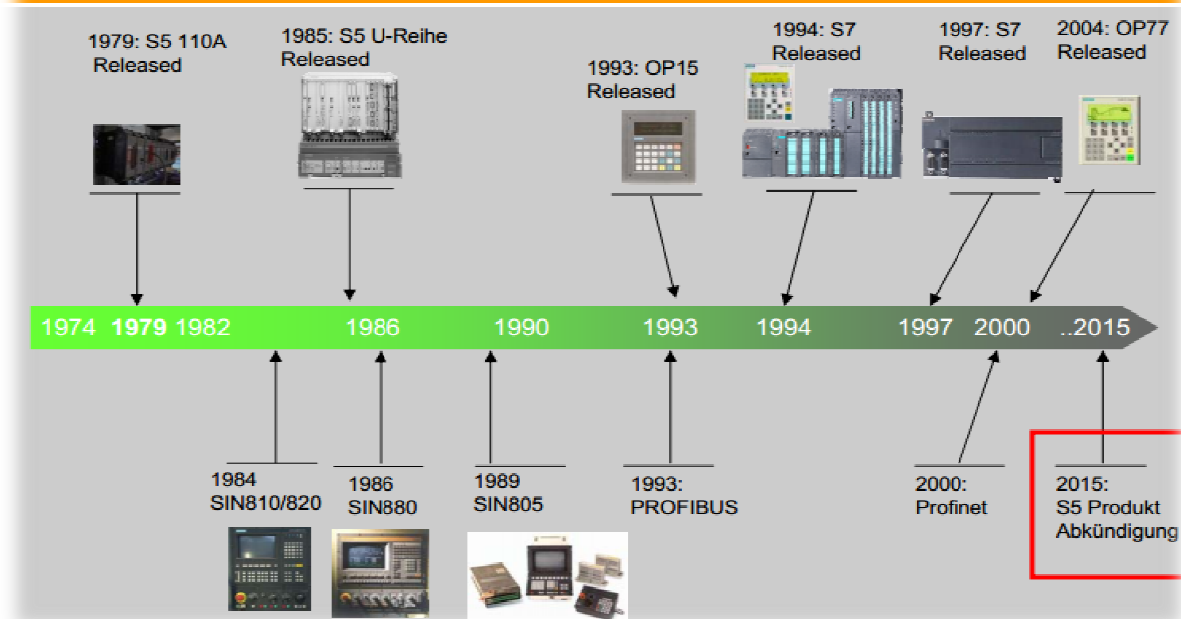


Abbildung 2-1 Produktkündigung

Des Weiteren zählen dazu die höhere Rechenleistung, verbesserte Diagnosefähigkeit sowie einfachere Programmierungsmöglichkeiten, um nur einige Verbesserungen des S7 gegenüber seinem Vorgänger zu nennen.

Siemens kündigte bereits vor längerer Zeit an, im Jahr 2015 das Produkt Simatic S5 aus ihrem Katalog komplett zu entfernen, sprich Reparatur und Ersatzteile sowie Technical Supportwürden entfallen.

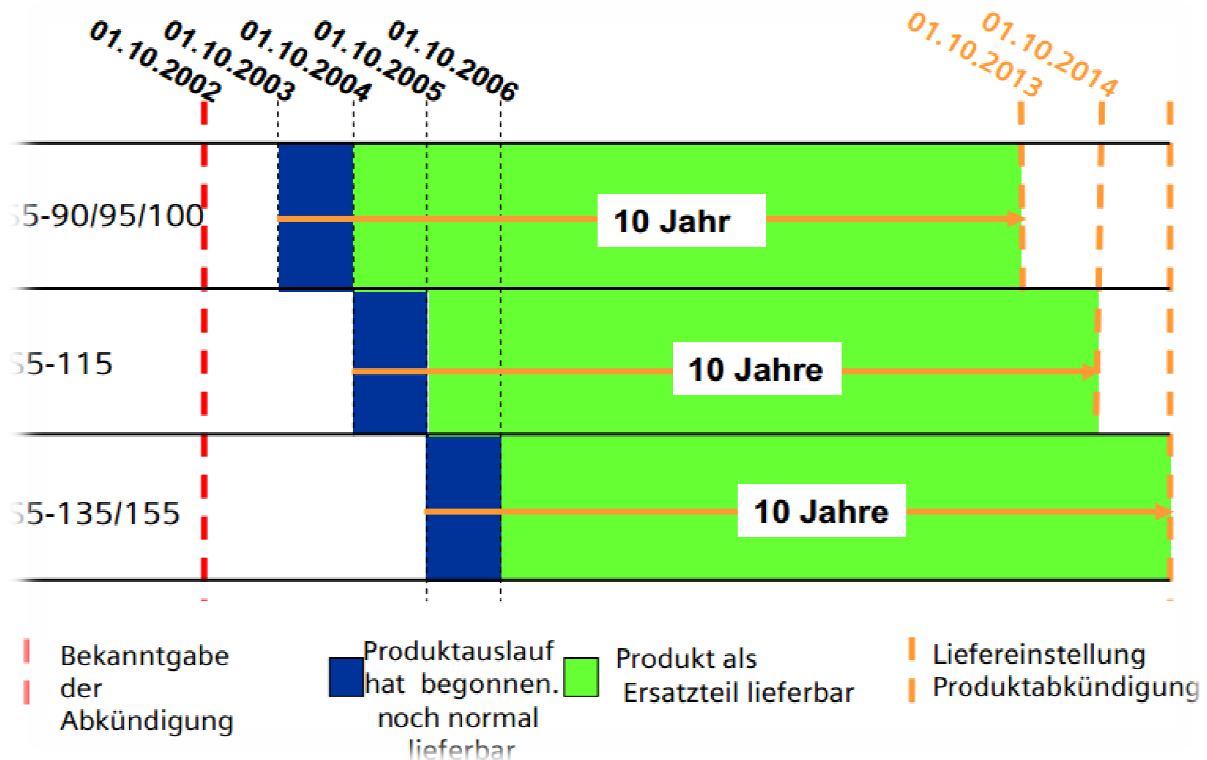


Abbildung 2-2 Simatic Historie

Zwar hat man zurzeit noch die Möglichkeit, genügend Ersatzteile vorzubestellen oder Support-Verträge für Reparatur und Ersatzteile abzuschließen, um seine bestehendes System weiter in Betrieb zuhalten, jedoch allein schon aus Kostengründen gibt es auf lange Sicht keine Alternativen.

Ein weiterer zu betrachtender Aspekt für den Wechsel ist die Verfügbarkeit des Know-Hows auf dem Markt, weilimmer weniger Techniker STEP 5 programmieren können. Die Behauptung, dass jeder, der in STEP 7 programmieren kann, dies auch in STEP 5 beherrscht, ist unbegründet. Die Programmierweise von STEP 5 und STEP 7 ist „ähnlich“, jedoch gibt es einige Feinheiten, die man bei der Nutzung der jeweiligen Version beachten muss. Ein Stichwort dazu wären z. B. die sogenannten „Schmiermerker“ von STEP 5.

Hier ist eine kurze Zusammenfassung der Vorteile einer Umrüstung von S5 auf S7:

- Kostenreduzierung bezüglich Ablauf der Anlage
 - Verfügbarkeit der S7 Ersatzteile
 - Reduzierung der Störungen und Stillstandzeiten durch verbesserte Diagnosemöglichkeit und Fernwartungsmöglichkeit
- Erhöhung des Bedienkomforts
- Vereinfachung der Programmierbarkeit
- Erhöhung der CPU Speicherkapazität
- Standardisierte Profibusschnittstelle
- Vereinfachte Möglichkeit zur Verknüpfung von Maschinen von unterschiedlichen Herstellern
- Erhaltung neuer gesetzlicher Anforderungen (Normen und Vorschriften)
- Erhöhung der Maschinenlebensdauer und Produktivität durch effiziente Steuerung
- Sicherung der Prozesssicherheit der Anlage
- MES (Manufacturing Execution System) Anbindungsmöglichkeit für Optimierung, automatische Produktplanung etc.

2.3 Migrationsmöglichkeiten

Unter dem Begriff Migration versteht man die Umrüstung von Simatic S5 auf Simatic S7 oder allgemein gesagt, das bestehende System wird gegen eine neuere Technologie ausgetauscht. Die Hardwareumrüstung kann dabei vollständig oder schrittweise erfolgen. Schrittweise insofern, dass die ältere S5 Peripherie prinzipiell auch S7 kompatibel ist.



Abbildung 2-3 Beispiel S5 Peripherie ET200

Außerdem kann die vorhandene Verkabelung der S5 Baugruppen unter Nutzung von Adaptern für die S7 Baugruppe weiter genutzt werden. Hierbei liegt der Vorteil darin, dass es keine aufwändige Umverdrahtung notwendig ist. Ein möglicher Fehler bei der Verdrahtung wird somit verhindert und, falls notwendig, kann das bestehende System mit wenig Aufwand wieder zurückgebaut werden. Durch die Nutzung von Adaptern ergibt insgesamt sich eine große Zeitersparnis, vor allem für die Inbetriebnahme.

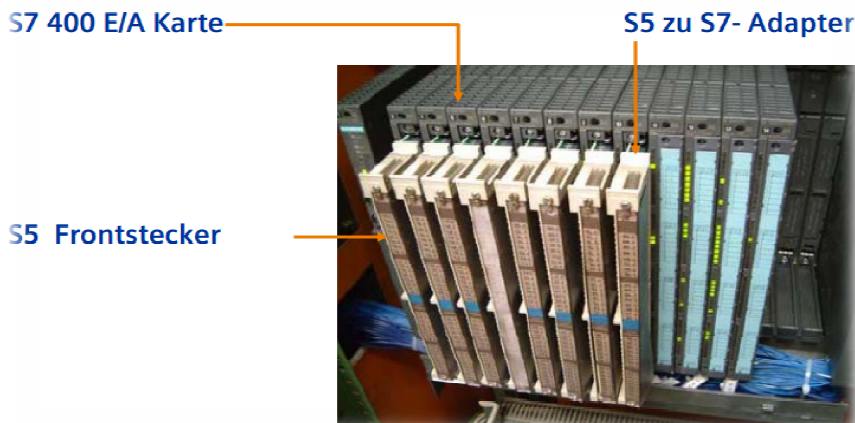


Abbildung 2-4 Beispiel Adapter Fronstecker S7-400

S7 300 E/A Karten

S7 300 DIN Adapter
Kit

S5 Frontstecker

S5 zu S7 Adapter

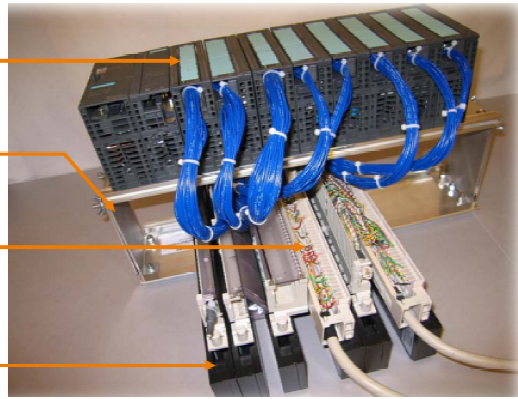


Abbildung 2-5 Beispiel Adapter Fronstecker S7-300

Für die Umrüstung in der Softwareebene bietet Siemens ein Konvertierungstool an, das in der Lage ist, Programmcode, Kommentare und Symbole von S5 auf S7 zu konvertieren. Die durchschnittliche Konvertierungsrate des Source Codes ist abhängig von den genutzten S5 Baugruppen. Nicht konvertierbar sind CPs, Profibus FMS, Zähler-Positionierung, HMI-Interface, direkte Speicherprogrammierung etc.

S5-Baugruppe	Konvertierungsrate in %
90/95/100U	90%
115U	80%
135U	70%
150U	60%
155U	60%

Tabelle 2 Konvertierungsrate

Auch wenn das Konvertierungstool eine mögliche Lösung darstellt, ist deren Nutzung dennoch nur bedingt brauchbar. Für kleineres Programm sollte das Konvertierungstool keine Probleme verbreiten. Jedoch für etwas komplexe Programme ist der zeitliche Aufwand für die Fehlersuche und Fehlerbehebung einfach zu intensiv. Dazu kommt noch die lückenhafte Konvertierung der Programmcodes, weil der zeitliche Aufwand für eine neue Entwicklung der Software wesentlich extensiver ist. Außerdem bietet der Simatic S7 eine erheblich bessere Möglichkeit, um sofort das Sachverhalten darzustellen, im Vergleich zu seinem Vorgänger Simatic S5. Ein Beispiel dazu wäre die Darstellung von Ablaufsteuerung in S7 Graph, was übersichtlicher als die Darstellung in FUP oder AWL ist.

3 Allgemeine Anlagenbeschreibung

Die Teilanlagen¹, auf die dieses modulare Steuerungskonzept angewendet werden soll, können aus mehreren Portalrobotern mit bis zu fünf Achsen bestehen, mehreren Rollenförderern, mehreren Rollenförderern in Kombination mit Riemenförderern und der dazugehörigen Aushebung sowie mehreren Lagerplätzen. In diesem Anlagenteil werden die Glasscheiben aus einem der Lagerplätze entnommen, zum Kipptisch gebracht und anschließend mit den Rollenbeförderern in Richtung Waschanlage für die weiteren Verarbeitungsprozesse transportiert.

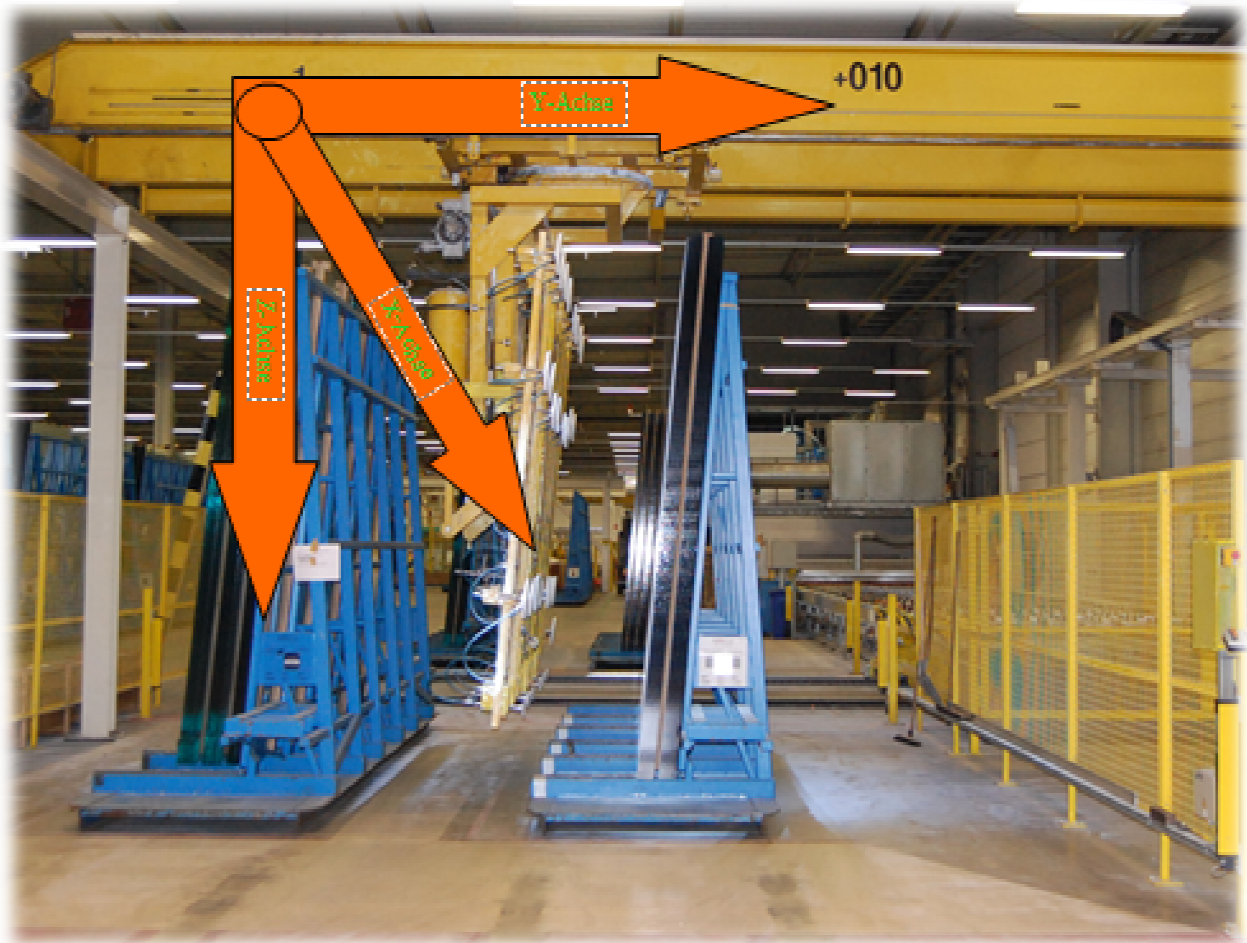


Abbildung 3-1 Achsenansicht Portalroboter (X-/Y-/Z-Achse)

¹siehe Anhang_04_Anlagenbeschreibung für weitere Bilder welcher der Anlagen darstelle



Abbildung 3-2 Achsenansicht Kipptisch (K-/L-Achse)

3.1 Achsenbeschreibung

Folgende Achsen können in hiesigen Anlagen vorkommen:

- X-Achse (Fahren Querrichtung Portalroboter)
- Y-Achse (Fahren Längsrichtung Portalroboter)
- Z-Achse (Heben/Senken Portalroboter)
- M-Achse (Drehen Saugerrahmen Portalroboter)
- L-Achse (Fahrachse Kipptisch)
- K-Achse (Kippachse Kipptisch)
- W-Achse (Schwenken Saugerrahmen – nicht in VSG1)

Dabei werden die X-Achse, die K-Achse und die L-Achse über Initiatoren und Endlagenschalter gesteuert. Die Anzahl der Initiatoren¹ ist abhängig von deren Funktionalität und kann von Anlage zu Anlage unterschiedlich sein. Die Y-Achse, die Z-Achse und die M-Achse hingegen besitzen Inkrementalgeber und werden mithilfe von Siemens FM351 Positionierbaugruppe geregelt.

Hinweis: Insgesamt werden zwei FM351 Module für dieses Teilprojekt benötigt, weil ein FM351 Modul nur zwei Inkrementalgeber unterstützt.

¹ Initiator Beispieldarstellung siehe Anhang_05_Achsenbeschreibung_X-Achse

3.2 Anlagenübersicht - Istzustand

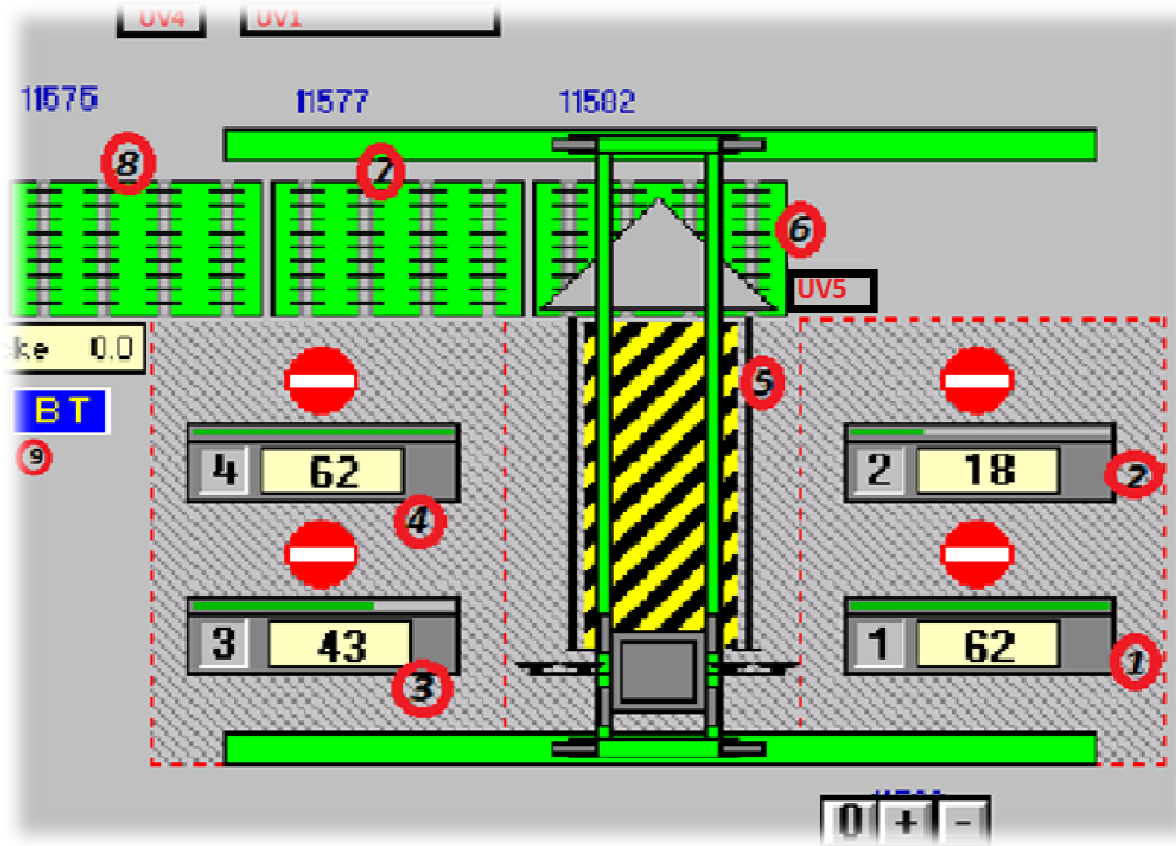


Abbildung 3-3 Übersicht VSG1 Anlage-Istzustand

- | | |
|------------------------|-------------------------------------|
| 1) Glasblock 1 | UV1 – Schaltschrank Steuerung VSG1 |
| 2) Glasblock 2 | UV4 – Hauptschaltschrank LISEC |
| 3) Glasblock 3 | UV5 – Schaltschrank LISEC Kipptisch |
| 4) Glasblock 4 | |
| 5) Portallader +010 | |
| 6) Kipptisch +020 | |
| 7) Rollenförderer +021 | |
| 8) Rollenförderer +022 | |
| 9) Bedienpult | |

3.3 Anlageübersicht - Sollzustand

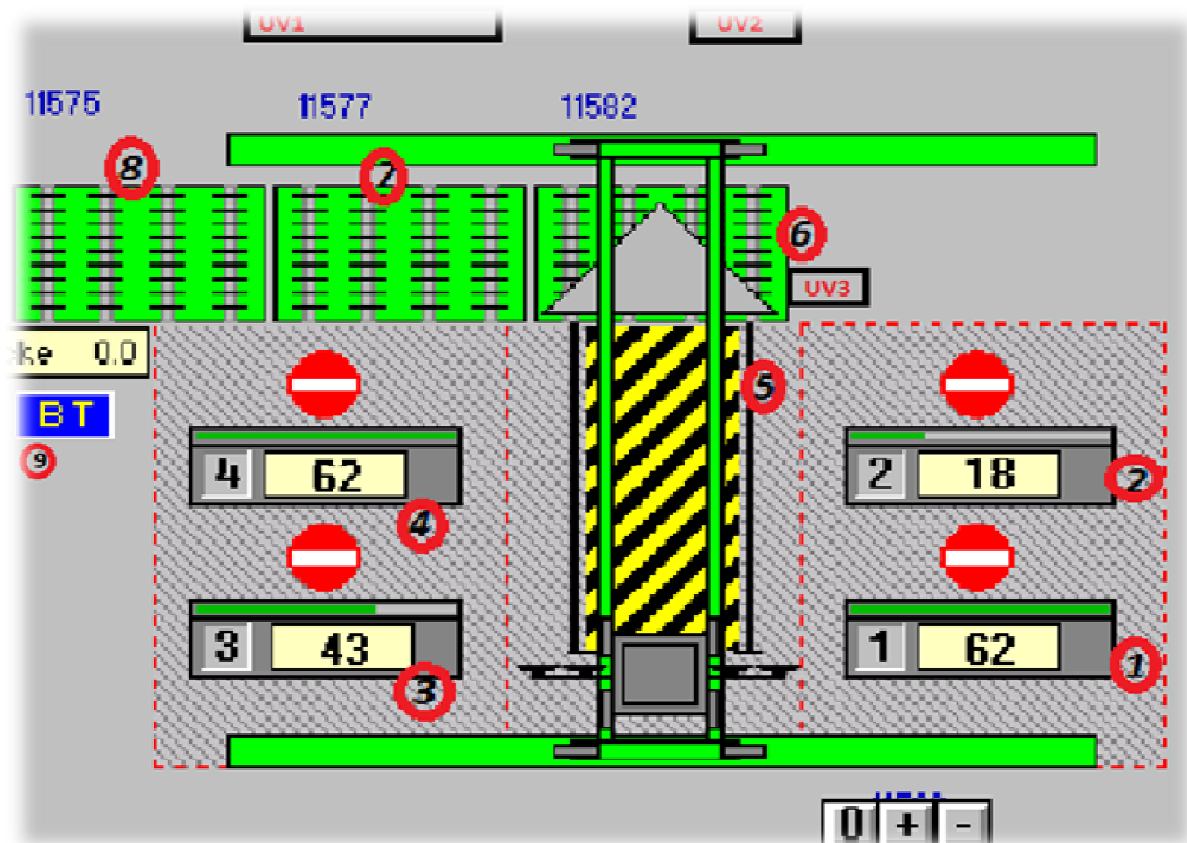


Abbildung 3-4 Übersicht VSG1 Anlage - Sollzustand

- | | |
|------------------------|--|
| 1) Glasblock 1 | UV1 – Schaltschrank Steuerung VSG1 |
| 2) Glasblock 2 | UV2 – Leistungsschaltschrank Kipptisch |
| 3) Glasblock 3 | UV3 – Verteilerkasten Kipptisch |
| 4) Glasblock 4 | |
| 5) Portallader +010 | |
| 6) Kipptisch +020 | |
| 7) Rollenförderer +021 | |
| 8) Rollenförderer +022 | |

3.4 Baugruppen Istzustand



Abbildung 3-5 genutzte S5 Baugruppen (Schaltschrank Steuerung VSG1)

Verwendete Baugruppen:

Bestell-Nummer	Bezeichnung
6ES5 240 - 1AA21	IP240 Positioning Decoder Module
6ES5 420 - 7LA11	32 Digital Input
6ES5 441 - 7LA12	32 Digital Output
6ES5 945 - 7UA13	CPU 945
6ES5 951 - 7LD21	Power Supply - PS951 7/15A
6GK1 243 - 0SA10	CP2430 AS-Interface Module
6GK1 543 - 1AA01	CP5431 Profibus Module
6ES5 308 - 3UC11	IM308C PB Interface Modul
6ES5 305 - 7LB11	IM305 Interface Module

Tabelle 3 S5-Teileliste

3.5 Baugruppen Sollzustand nach Modernisierung

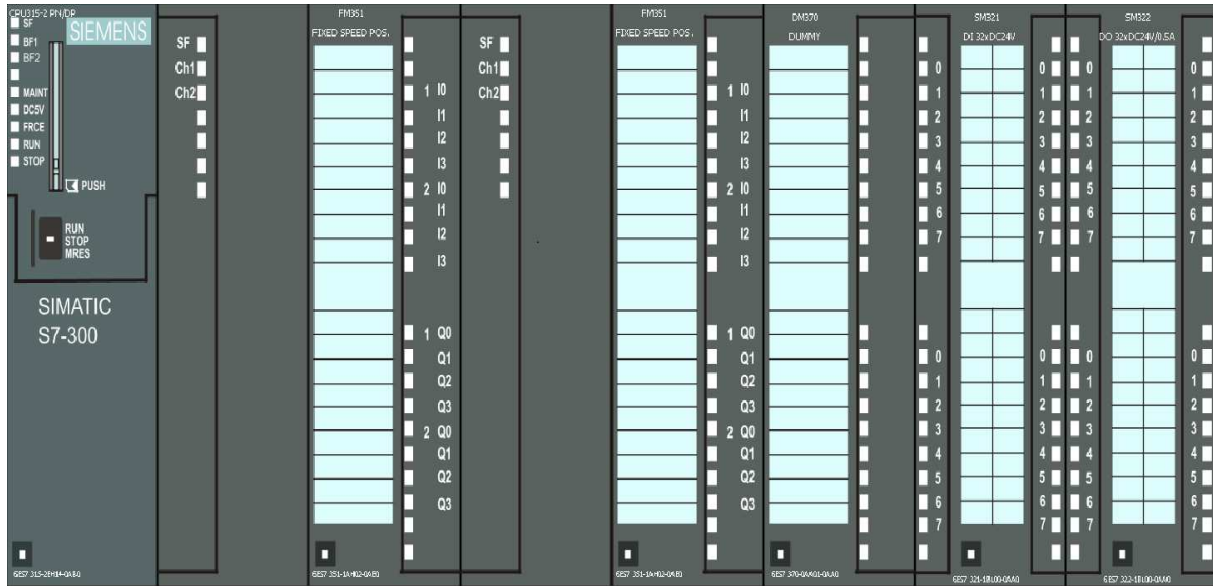


Abbildung 3-6 S7-Baugruppen

Verwendete Baugruppen:

Bestell-Nummer	Bezeichnung
6ES7315-2EH14-0AB0	Zentralbaugruppe CPU315-2 PN/DP
6ES7321-1BL00-0AA0	Digitaleingabe 32DE, DC 24V potentialgetr.
6ES7322-1BL00-0AA0	Digitalausgabe 32DA, DC 24V, 0,5A potentialgetr.
6ES7351-1AH02-0AE0	Positionierbaugruppe FM 351 inkl. Proj.-paket
6ES7351-1AH01-0AE0	Positionierbaugruppe FM 351 inkl. Proj.-paket
6ES7370-0AA01-0AA0	DM370 Dummy Module
6ES7390-1AE80-0AA0	Profilschiene 480 mm
6ES7392-1AJ00-0AA0	Frontstecker 20-polig mit Schraubkontakten
6ES7392-1AM00-0AA0	Frontstecker 40-polig mit Schraubkontakten
6ES7953-8LJ30-0AA0	Micro Memory Card 512 KByte

Tabelle 4 S7-Teileliste

3.6 Hardwarestruktur

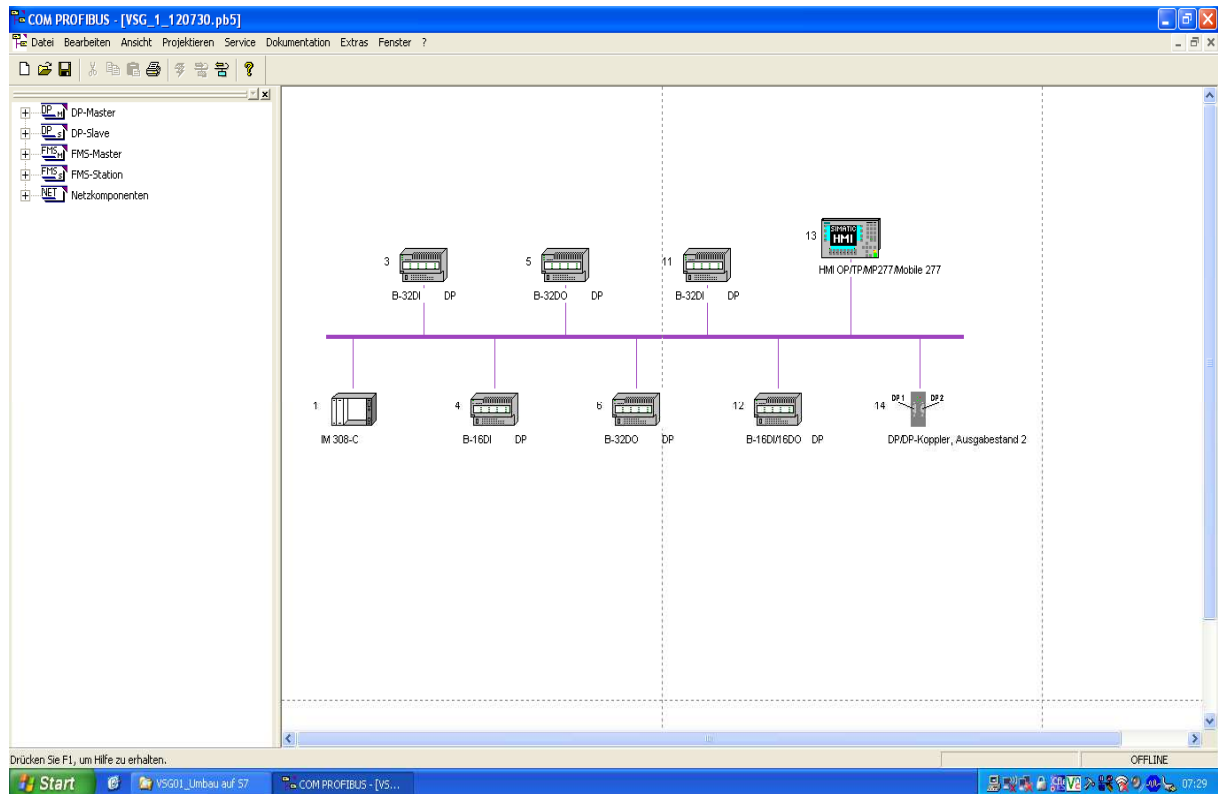


Abbildung 3-7 S5 Hardwarestruktur

Die derzeitige S5 Hardware besteht aus den obengenannten Steuerungskomponenten (siehe 3.4), sechs weiteren ET-Stationen und einem DP/DP Koppler, der zum Kipptisch führt. Ein diverser Slave wird über die ET-Station und den DP/DP Koppler gesteuert, der die benötigten Sensor und Aktor steuert.

Im Rahmen des Umbaus bleibt die HW Struktur weitestgehend unverändert. Sowohl die ET-Stationen, als auch der DP/DP Koppler bleibt erhalten. Es werden nur die Steuerungskomponenten der S5 durch Komponenten der S7 ersetzt, auch kommt es zum Austausch des HMI Moduls durch eine neueres Modell. Zusätzlich werden vier weitere Slaves an der UV3 angebracht, die die Steuerung des Kipptisches übernimmt. Außerdem wird die Anzahl der DP/DP Koppler auf drei erhöht. Die beiden neuen DP/DP Koppler dienen zur Datenkommunikation mit der Waschmaschine sowie zur Bottero Linie. Sowohl die Anzahl, als auch die Position der Aktoren und Sensoren bleiben im Rahmen der Modernisierung unverändert. Hierbei erfolgt der Versuch der Prozessoptimierung zunächst nur auf Softwareebene, sprich, die Änderung der Schrittketten bzw. die Änderung des Prozessablaufs.

4 Projektmanagement

Das Projektmanagement¹ wird durch die DIN Normenreihe **DIN 69901** geregelt. Es beschreibt die Grundlagen, Prozesse, Prozessmodell, Methode, Daten, Datenmodell und Begriffe im Projektmanagement. Dabei wird es in fünf Unterpunkte geteilt:

- DIN 69901-1 „Grundlagen“
- DIN 69901-2 „Prozesse, Prozessmodell“
- DIN 69901-3 „Methoden“
- DIN 69901-4 „Daten, Datenmodell“
- DIN 69901-5 „Begriffe“

Die DIN 69901-1 und DIN 69901-2 beschreiben verschiedene Grundbegriffe, wie Projekt, Projektmanagement oder Projektwirtschaft.

Die Nomenklatur zur Projektgliederung, Projektstrukturplan, Arbeitspaket und Netzplantechnik hingegen werden durch DIN 69901-3 geregelt.

Der DIN 69901-4 definiert Projektorganisation, Projektleitung und Projektleiter.

Der DIN 69901-5 beschreibt zum Schluss die Projektphase bzw. die Berichte zum Projektabschluss.

Diese DIN Normenreihe wurde zuletzt im Januar 2009 ausgegeben. Weitere Ergänzungen zu dieser Thematik finden sich auf **DIN 69900:2009-01**.

Weitere wichtige Normen, die zu beachten sind, wären die **ISO 10006:2003-06** für Qualitätsmanagementnorm, sowie der **ISO 21500:2012-09** „Leitfaden zum Projektmanagement“.

¹<http://www.verwaltung-innovativ.de> Suchbegriff Organisation; Prozessmanagement

4.1 Einführung

Die Projektplanung¹ ist der erste und äußerst wichtige Schritt nach dem Start des Projekts. Eine fundierte Planung ist die Basis für eine wirksame Kontrolle und Steuerung des Projektes.

Generell ist zu sagen, dass die Projektplanung eine Aufgabe ist, die über die gesamte Projektlaufzeit wahrgenommen werden muss. Der Projektplan ist lebendig und muss ständig analysiert und angepasst werden, ohne dabei den bestehenden Auftrag aus den Augen zu verlieren.

Die Planung eines Projekts orientiert sich am Projektauftrag. Dabei werden bereits vorhandene Planungsdaten (z. B. vorgegebene Ressourcen) berücksichtigt und um weitere Aspekte ergänzt.

Der Detaillierungsgrad der Planung ist abhängig von der Größe und der Komplexität des Projektes. Wird ein Projekt nicht detailliert genug geplant, besteht die Gefahr, dass die Übersicht über wichtige Einzelheiten verloren geht und die Projektleitung die Koordinations-, Führungs- und Kontrollfunktion nicht adäquat wahrnehmen kann. Eine zu detaillierte Planung erhöht den Aufwand sowohl zu Beginn, als auch während der Durchführung des Projektes, weil aufgrund des hohen Detaillierungsgrades auch andere Aufgaben der Projektleitung (z. B. Kontrolle) an Umfang zunehmen können.

Für eine effiziente und effektive Projektplanung empfiehlt sich eine „**rollierende**“ **Vorgehensweise**. Das bedeutet, dass zu Beginn des Projektes eine grobe Planung für das gesamte Projekt vorgenommen wird, die jeweils für die einzelne anstehende Projektphase weiter detailliert wird. Der Vorteil dieser Vorgehensweise ist, dass im Laufe der Untersuchung gewonnene Erkenntnisse in die Planung einfließen können und der Planungsaufwand vertretbar bleibt.

¹<http://www.wirtschaftslexikon24.net/d/planungsarten/planungsarten.htm>

4.2 Projektstrukturplan

Der Projektstrukturplan (PSP) illustriert in einer Übersicht alle Arbeitspakete, die zur Erreichung der Projektziele notwendig sind, hierarchisch strukturiert. In dieser frühen Projektphase kann der PSP noch nicht die endgültige Detailtiefe erreichen, daher muss er im Laufe der Untersuchung an den jeweils aktuellen Wissensstand angepasst werden. Gewisse Vorstellungen über die methodische Vorgehensweise während der Untersuchung müssen jedoch bereits vorhanden sein.

Formale Bestandteile eines PSP sind:

- die Hauptaufgabe (Bezeichnung des Projekts),
- die Teilaufgaben (Aufgabenblöcke resultierend aus der Hauptaufgabe),
- die Arbeitspakete (detailliert beschriebene, abgrenzbare, bewertbare Tätigkeiten, die für die Teammitglieder während der Untersuchung anfallen).

Das Ziel des PSP ist die Erfassung aller Arbeitspakete und Verantwortlichkeiten zur Erhöhung der Projekttransparenz und eine Grundlage für die folgenden Planungsschritte. Weiterhin dient der PSP als Grundlage zur Risikoanalyse, zum Projektcontrolling und als Gliederungsschema für Projektdokumente und deren systematische Ablage.

Die Erstellung des PSP erfolgt üblicherweise „Top to Down“. Dabei beginnt man mit der Hauptaufgabe, die Schritt für Schritt in die Teilaufgaben (Projektphasen und Hauptmeilensteine) und Arbeitspakete aufgegliedert wird. Ein Arbeitspaket ist also eine Teilaufgabe, die nicht weiter sinnvoll zu unterteilen und eindeutig einem Teammitglied zuzuordnen ist. Die Gliederung der einzelnen Teilaufgaben kann, je nach Zweckmäßigkeit, sowohl funktionsorientiert, als auch objektorientiert vorgenommen werden. Dabei muss allerdings darauf geachtet werden, dass es nicht zu Überschneidungen zwischen Arbeitspaketen oder Aufgaben kommt.

Die Darstellung des PSP kann sowohl in Form eines Baumdiagramms oder als Gliederungsliste erfolgen:



Abbildung 4-1 Beispielmodell eines PSP als Baumdiagramm bezogen auf durchgeführte Projekt

Nach der Erstellung des PSP sollte dieser noch einmal auf Vollständigkeit, Eindeutigkeit und Überschneidungsfreiheit geprüft werden. Anschließend werden Identifikationsnummern für Aufgaben und Arbeitspakete vergeben. Diesbezüglich richtet sich die Anzahl der Stellen der zu vergebenden Nummer nach der Gliederungsebene des PSP, in der sich die Aufgabe befindet.

Gegebenenfalls kann es sinnvoll sein, einzelne Arbeitspakete detailliert zu beschreiben. Dies ist vor allem dann empfehlenswert, wenn kein einheitliches Verständnis über deren Inhalt und den damit verbundenen Verantwortlichkeiten herrscht oder ein Wechsel der Teammitglieder möglich ist.

4.3 Projektcontrolling

Unter dem Begriff Projektcontrolling wird sowohl die Projektkontrolle, als auch die Projektsteuerung zusammengefasst. Dabei ist die Projektkontrolle die Ermittlung von aktuellen Kennzahlen und Projektdaten sowie der anschließende Vergleich mit den Planungsdaten. Gegebenenfalls kann hierfür auch auf bereits vorhandene Controlling-Daten zurückgegriffen werden.

Darauf aufbauend umfasst die Projektsteuerung koordinierte Maßnahmen zur Sicherstellung der geplanten Ergebnisse, also die Übereinstimmung des tatsächlichen mit dem geplanten Projektzustand.

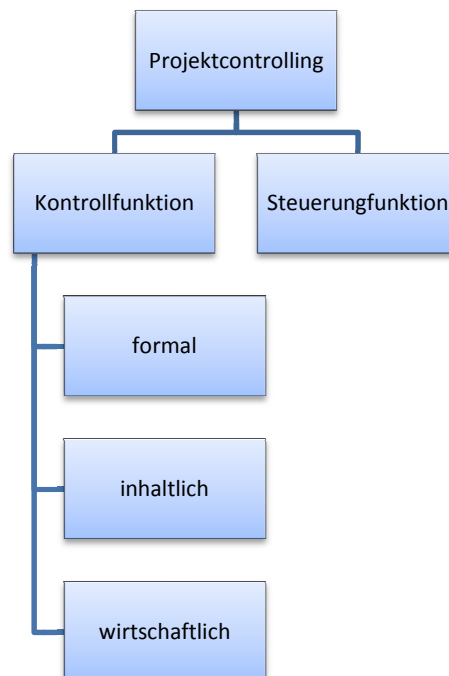


Abbildung 4-2 Kernbereich des Projektcontrolling

Formale Aspekte der Kontrollfunktion betreffen beispielsweise die Einhaltung gesetzlicher, vertraglicher und behördeninterner Vorschriften und Vereinbarungen. Inhaltliche Aspekte sind z. B. die Kontrolle der Zielerreichung und die Qualitätskontrolle. Bei wirtschaftlichen Kontrollaspekten handelt es sich z. B. um den Ressourcenverbrauch, die Termineinhaltung, das Budget und die Risikosituation.

Projektcontrolling hat das Ziel, mögliche Fehlentwicklungen (formaler, inhaltlicher und wirtschaftlicher Art) des Projektes frühzeitig zu erkennen, damit geeignete Steuerungsmaßnahmen rechtzeitig eingesetzt werden können.

In der Praxis kommt es häufig vor, dass entsprechende Kontrollmechanismen und Steuerungsmechanismen erst zu spät etabliert werden, wenn auffallende Fehlentwicklungen schon offensichtlich und steuernde Maßnahmen kaum noch in der Lage sind, das Projekt wieder auf den richtigen Weg zu bringen. Aus diesem Grund ist es äußerst wichtig, Projektcontrolling bereits in der Vorbereitungsphase zu installieren und fortwährend zu betreiben.

Um ein leistungsfähiges Projektcontrolling einführen zu können, muss sich die Projektleitung zunächst überlegen, welche Informationen den Zustand des Projektes so wiedergeben können, dass daraus gültige Schlussfolgerungen getroffen werden können. Die zu erhebenden Projektdaten müssen unbedingt mit geplanten Daten vergleichbar sein.

Zusätzlich zu den Kontrollobjekten aus dem ökonomischen Bereich (Termine, Kosten) sind auch inhaltliche Aspekte zu prüfen. Hier ist primär die Qualität der Arbeitsergebnisse bedeutend, aber auch die Aktualität der Dokumentation kann ein Indiz für die Entwicklung des Projektes sein. Zusätzlich kann die Einhaltung von formalen Vorgaben Hinweise geben und sollte regelmäßig überprüft werden. Dazu zählen unter anderem die Einhaltung von Stellvertreterregelungen und vertraglichen Vereinbarungen.

Sind die relevanten Controlling Inhalte festgelegt, muss entschieden werden, wie die Erfassung der Controlling Daten erfolgen soll. Hier bieten sich verschiedene Methoden an:

Formale / Schriftliche Abfragen	Teamorientierte Datengewinnung
<ul style="list-style-type: none"> ➤ Termin-/Ablaufmeldungen ➤ Kostenerfassungsbelege ➤ Stundensaufschreibung bzw. <li style="padding-left: 20px;">Arbeitszeiterfassung ➤ Rückmeldelisten für Arbeitspakete 	<ul style="list-style-type: none"> ➤ Statusbesprechungen ➤ informelle Rückmeldungen aus dem Team
<ul style="list-style-type: none"> ➤ Vorteil: durch schriftliche Dokumentation gute Nachvollziehbarkeit ➤ Nachteil: mögliche emotionale Ablehnung bei Projektteammitgliedern, hoher Zeitaufwand 	<ul style="list-style-type: none"> ➤ Vorteil: Informationen können während der Gewinnung direkt abgesichert und verteilt werden ➤ Nachteil: Besprechung muss diszipliniert ablaufen, um wirklich alle relevanten Daten sammeln zu können

Beobachtung	Qualitätsbewertung
➤ Rundgänge	<ul style="list-style-type: none"> ➤ Checklisten ➤ Reviews
Bemerkung: besonders zur Erhebung „weicher“ Daten geeignet (Stimmung, Motivation)	Bemerkung: detaillierte Analyse der Arbeitsergebnisse anhand von Vorgaben

Tabelle 5 Möglichkeiten der Datengewinnung für das Projektcontrolling

Aus den gewonnenen Controlling Daten werden aussagefähige Kennzahlen gebildet. Im Anschluss werden diese mit den Plandaten verglichen.

Identifizierte Abweichungen stellen Indizien für Fehlentwicklungen im Projekt dar. Kennzahlenbasiert handelt es sich dabei um messbare Abweichungen (Termine, Kosten). Darüber hinaus können auch sonstige „schwache“ Signale, wie geringe Mitarbeitermotivation oder Kommunikationsprobleme, Aufschluss über Fehlentwicklungen im Projekt geben.

Ursachen für die Abweichung der Ist-Daten von den Soll-Daten sind in den meisten Projekten ähnlich:

- unrealistische Zielformulierung und/oder Planung,
- personelle Ursachen (Überlastung, mangelnde Qualifikation, fehlende Motivation),
- organisatorische Ursachen (unklare Aufgaben und Kompetenzen, fehlende Unterstützung durch den Auftraggeber oder die Behördenleitung).

Sobald Indikatoren für Abweichungen vom Plan auftreten, sind entsprechende Steuerungsmaßnahmen einzuleiten, um das Projekt wieder auf einen erfolgreichen Weg zu führen. Anwendbare Steuerungsmaßnahmen können sein:

- die rechtzeitige Beeinflussung der Kosten- und Leistungsparameter (Terminverschiebung, Budget und Ressourceneinsatz modifizieren),
- im Projekt durch:
 - Zielreduktion, Konzentration auf die Muss-Ziele,
 - Akzeptanz von Risiken (Untersuchungsgenauigkeit senken),
- Reorganisation des Projektteams (Änderung der Aufgabenverteilung oder der Kompetenzen, Austausch von Beschäftigten, Einsatz Externer).

Hinsichtlich der geeigneten Kontrollzeitpunkte gibt es keine feste Regel. Die Kontrollzeitpunkte sind von der Dauer und Komplexität des Projektes, vom Informationsbedarf der Projektleitung und des Auftraggebers und natürlich auch vom aktuellen Status des Projektes abhängig. Üblicherweise finden in bereits gefährdeten Projekten häufiger Kontrollen statt als in Projekten, die sich immer auf dem richtigen Weg befinden. Die Häufigkeit der Kontrollen und der Kontrollzeitpunkt sind generell so zu wählen, dass bei identifizierten Fehlentwicklungen noch die Möglichkeit der Gegensteuerung besteht.

Hinzuweisen ist noch darauf, dass beim Projektcontrolling auch psychologische Faktoren eine Rolle spielen. Einerseits fühlen sich die Projektteammitglieder möglicherweise überwacht, was sich negativ auf die Motivation auswirken kann. Andererseits kann auch eine bewusste oder unbewusste Manipulation der Statusmeldungen durch die Beschäftigten stattfinden (z. B. 90%-Syndrom: Arbeitspakete werden schnell als „fast fertig“ gemeldet, aber dann nicht abgeschlossen). Dem wirkt nur eine kooperative, vertrauensvolle Projektkultur entgegen, in der Probleme offen kommuniziert werden können und diese anschließend gemeinsam beseitigt werden, ohne dass Schuldzuweisungen erfolgen. Dann werden die Controlling-Maßnahmen als gängige Methoden des Projektmanagements anerkannt.

Ein sehr hilfreiches Werkzeug für ein erfolgreiches Projektcontrolling wäre die Erstellung und Benutzung von Checklisten bzw. die Festlegung von Meilensteinen. Es ist zu empfehlen, gleich zum Projektbeginn eine grobe Meilensteinaufteilung für den einzelnen Aufgabenbereich anzufertigen, was man in der späteren Planung weiter konkretisiert. Nach der Konkretisierung werden die einzelnen Punkte mit einer Zeitplanung versehen. Durch das Vergleichen zwischen dem Ist-Zeitplan und dem Soll-Zeitplan kann der Projektfortschritt so effizient überprüft werden bzw. können Maßnahmen entwickelt und durchgeführt werden, um einer Abweichung im Projektfortschritt entgegenzutreten.

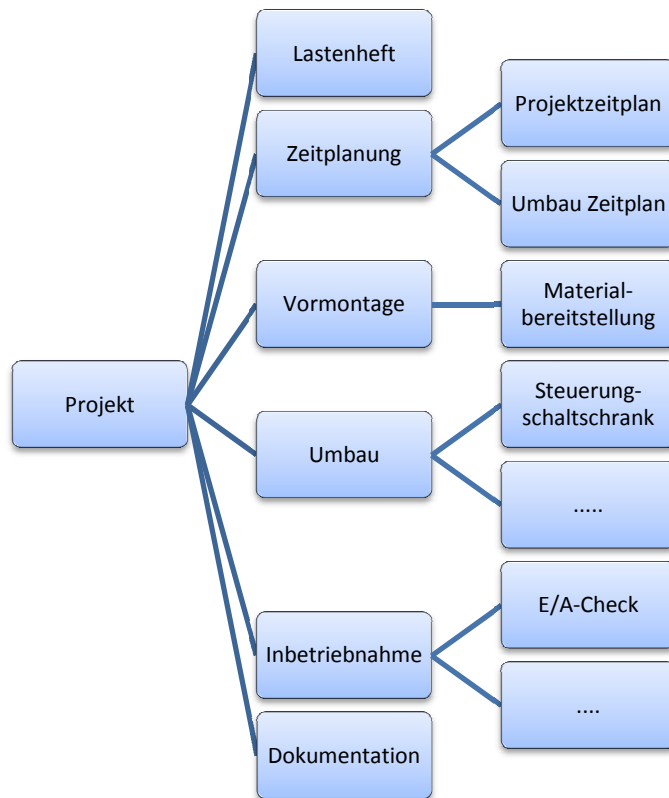


Abbildung 4-3 Beispiel Checklisteunterteilung/Meilensteine

4.4 Risikomanagement

Risikomanagement¹ ist ein Instrument des Managementwerkzeugkastens, das sich mit dem systematischen, planvollen Umgang mit Risiken verschiedenster Art beschäftigt. Unter einem Risiko versteht man dabei ein Problem, das noch nicht eingetreten ist. Ziel des Risikomanagements ist, entweder zu verhindern, dass Risiken zu Problemen werden, oder aber den Schaden beim Eintritt der Risiken zu minimieren. Es dient dadurch auch der Schaffung von Transparenz, der Beseitigung von Unsicherheiten und ermöglicht somit die Nutzung von Chancen. Durch die konsequente Suche und Analyse möglicher Risiken wird zudem auch eine möglicherweise vorhandene Betriebsblindheit beseitigt.

Mögliche Projektrisiken können folgendermaßen klassifiziert werden:

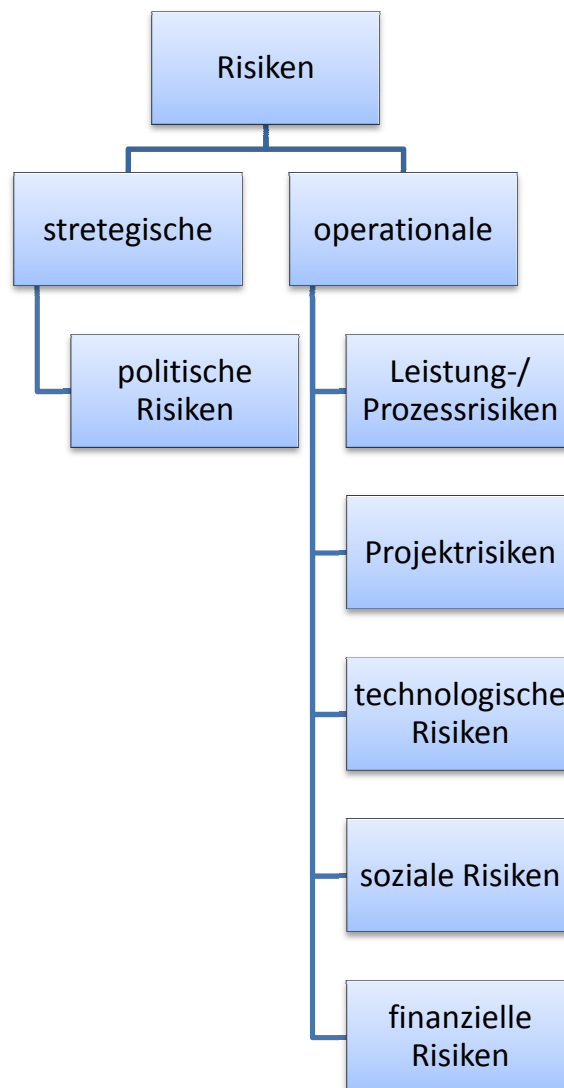


Abbildung 4-4 Risikoklassifizierung

¹ Beispiel siehe Anhang_02_Inbetriebnahmedokument, Tabelle 1 „Risikoanalyse“

Das **Strategische Risikomanagement** bildet die integrative Klammer und das Fundament des gesamten Risikomanagements. Beim strategischen Risikomanagement geht es vor allem um die Formulierung von Zielen für eine Risikopolitik sowie die Grundlagen der Unternehmen des Risikomanagements. Diesbezüglich wird etwa festgelegt, wie ein Unternehmen generell mit Risiken umgeht, z. B. risikofreudig, risikoneutral oder risikovermeidend.

Hierbei ist die Risikostrategie im Wesentlichen vom Leitbild und den Zielen der Organisation abhängig. Beispielsweise kann eine Institution nicht risikovermeidend agieren, wenn eines ihrer Ziele Innovationen beinhaltet, weil mit Innovation einerseits große Chancen, andererseits aber auch große Risiken verbunden sind.

Der **operative** Teil des Risikomanagements setzt auf die Risikopolitik und Risikostrategie der Organisation und wendet die festgelegten Grundsätze auf Risiken an, die an verschiedenen Stellen auftreten kann.

- Prozessrisiken betreffen den Leistungserstellungsprozess, damit auch das Produkt, und haben somit eine direkte Außenwirkung.
- Finanzielle Risiken betreffen die wirtschaftliche Situation einer Organisation.
- Projektrisiken gefährden die inhaltliche Zielerreichung, Kosten- und Terminsituation von Projekten. Beispiele für häufige Projektrisiken sind zu enge Zeitpläne, unvollständig definierte Rollen usw.
- Soziale Risiken betreffen das soziale Gefüge einer Organisation, also maßgeblich den Beschäftigten (Fluktuation, Motivationsmangel).
- Technologische Risiken entstehen durch den Einsatz neuer Technologien. (z. B. Teilumbau von S5 auf S7.)

Effektives Risikomanagement erfordert ein gezieltes, systematisches Vorgehen, das die folgenden Schritte beinhalten sollte:

➤ *Risiken identifizieren*

Die Quellen zur Ermittlung von Risiken einer Organisation können unterschiedlichen Ursprungs sein. Einerseits ist die Risikoidentifikation in Form eines Brainstormings möglich („Katastrophen–Brainstorming“). Die Beteiligten sammeln dabei alle möglichen Katastrophenszenarien, die ihr Projekt, einen Prozess oder die Organisation (je nachdem in welchem Kontext das Risikomanagement durchgeführt wird) betreffen könnten. Im Anschluss an die Ideensammlung werden durch die Erstellung eines Ursache–Wirkung–Diagramms die Ursachen für das Eintreten der Szenarien analysiert und somit die Risiken ermittelt.

Eine andere Möglichkeit zur Ermittlung von Risiken ist die Nutzung der Erfahrungen der Vergangenheit, denn die Probleme der Vergangenheit stellen potentielle Risiken in der Gegenwart dar. Eine Fundstelle sind Berichte aus Projektbewertungen „Lessons Learned“, die mittels Dokumentenanalyse ausgewertet werden können und demonstrieren den individuellen Erfahrungsschatz der Beschäftigten.



Abbildung 4-5 Mögliche Risiken des Projekts

➤ *Risiken analysieren und bewerten*

Alle gesammelten Risiken werden eindeutig benannt, beschrieben und in eine Risikoliste aufgenommen. Teil der Beschreibung sollten unbedingt Eintrittsindikatoren sein, die den Eintritt des Risikos frühzeitig ankündigen. Sie dienen der späteren kontinuierlichen Überwachung der Risiken und sind Grundvoraussetzung für eine gezielte Steuerung durch Maßnahmen.

Die erfassten Risiken werden hinsichtlich ihrer Eintrittswahrscheinlichkeit bewertet und priorisiert. Die Priorisierung erfolgt in Abhängigkeit zur qualitativen Bewertung anhand des folgenden Bewertungsschemas:

		Eintrittswahrscheinlichkeit		
		Low <25%	Medium 27%-75%	High >75%
		1	2	3
Schadenausmaß	Gering (wenig relevante Ergebnisse gefährdet)	1	1	1
	Mittel (Teilergebnisse gefährdet)	2	2	3
	Schwer (Erfolg des gesamte Projektes gefährdet)	3	3	3

Tabelle 6 Bewertungsschema zwischen Schadenausmaß und Eintrittswahrscheinlichkeit

Zusätzlich zur qualitativen Bewertung sollten Risiken, deren Schadenshöhe bezifferbar ist, auch quantitativ bewertet werden. Dabei könnte es sich um Schäden in Form von Terminverzögerungen (ausgedrückt in Tagen) oder zusätzlichen Kosten handeln.

➤ *Mögliche Maßnahmen und Risiken beobachten*

Folgenden Ansätzen sind grundsätzlich möglich, um diese Thematik anzugehen:

- **Risikovermeidung:** Wird ein Risiko vermieden, bedeutet das meistens, dass eine Aktion nicht durchgeführt wird. Das bedeutet aber gleichzeitig, dass Chancen, die mit dieser Aktion verbunden sind, nicht wahrgenommen werden. Als Beispiel kann die Einführung einer neuen IT-Anwendung genannt werden, die einerseits Risiken birgt, andererseits aber auch Chancen mit sich bringt.
- **Eingrenzung der Risiken:** Risiken können einerseits durch die Reduktion der Eintrittswahrscheinlichkeit, andererseits durch die Reduktion der Schadenshöhe bei Eintritt eingegrenzt werden. Zu diesem Zweck sind Maßnahmen zu entwickeln und mit dem Termin und dem Verantwortlichen in der Risikoliste zu dokumentieren. Es kann sich dabei um Maßnahmen handeln, die direkt durchgeführt werden müssen, aber auch um Eventualfallmaßnahmen, die durchzuführen sind, wenn das Risiko eintritt und zum Problem wird. Sind Maßnahmen zur Risikoeingrenzung getroffen, wird das Risiko erneut bewertet und gegebenenfalls akzeptiert.
- **Risikoakzeptanz:** Eine Möglichkeit, Risiken zu begegnen, ist deren Akzeptanz. Wenn mit einer Aktion in großem Maße positive Effekte verbunden werden, kann die Bereitschaft vorliegen, ein bekanntes Risiko zu akzeptieren, wenn dessen Schadenshöhe die positiven Effekte der Aktion nicht kompensiert. Ein weiterer Grund für die Akzeptanz eines Risikos kann sein, dass die Schadenshöhe die Durchführung von geeigneten Maßnahmen zur Eingrenzung des Risikos wirtschaftlich nicht rechtfertigt.
- **Risikobeseitigung:** Das Abstellen eines organisatorischen Mangels kann ein bestehendes Risiko ablösen.

Das Ergebnis der vorangegangenen Schritte ist eine Risikoliste, die als Grundlage für die kontinuierliche Überprüfung und Steuerung der vorhandenen Risiken (Risikocontrolling als Teil des Controllings) dient. Im Laufe der Zeit können neue Risiken hinzukommen, die dann erneut analysiert, bewertet und mit Maßnahmen belegt werden müssen. Jedoch können aufgrund von Veränderungen der Rahmenbedingungen auch Risiken wegfallen oder sich deren Eigenschaften verändern.

Die verbleibenden, nicht vermeidbaren Risiken, haben eine direkte und nicht zu unterschätzende Auswirkung auf die Projektplanung. Einerseits sind die Maßnahmen zur Risikoeingrenzung finanziell und terminlich einzuplanen, andererseits ist ebenfalls zu berücksichtigen, dass der Eintritt der verbleibenden Risiken noch immer (wenn auch einen akzeptierbaren) Schaden mit sich bringt. Eine Möglichkeit zu dessen Berücksichtigung ist die Bildung von Risikorückstellungen für quantitativ bewertbare Risiken.

Die einzuplanende Risikorückstellung für ein Risiko ergibt sich aus der Multiplikation seiner Eintrittswahrscheinlichkeit mit der Schadenshöhe:

$$\text{Höhe der Risikorückstellung} = \text{Eintrittswahrscheinlichkeit} \times \text{Schadenshöhe}$$

Beispiel: Eine Risiko tritt mit einer Wahrscheinlichkeit von 30% ein und verursacht im Falle des Eintritts einen Schaden von 10.000 €. Des Weiteren wird mit einer Terminverschiebung von fünf Tagen gerechnet.

$$\text{Höhe der finanziellen Risikorückstellung} = 0,3 \times 10.000 \text{ €} = 3.000 \text{ €}$$

$$\text{Höhe der terminlichen Risikorückstellung:} = 0,3 \times 5 \text{ Tage} = 1,5 \text{ Tage}$$

Tritt ein Risiko, für das eine Rückstellung in der Termin- und Kostenplanung berücksichtigt wurde, ein, ist die Rückstellung natürlich nicht ausreichend, um den gesamten Schaden abzudecken. Die Gesamthöhe der eingeplanten Rückstellungen in Form von Budget oder Zeit werden allerdings ausreichend sein, weil nicht alle Risiken eintreten.

Risikomanagement setzt Offenheit und Ehrlichkeit in der Kommunikation¹ voraus. Es ist darauf angewiesen, dass erwartete Risiken durch die Beschäftigten und andere Interessengruppen auch wirklich frühzeitig offen gelegt werden. Fehlt die entsprechende offene Kommunikation, kann es sein, dass Risiken so lange wie möglich verschwiegen werden und als Probleme zu Tage treten, die kaum mehr zu beeinflussen sind.

¹Beispielsiehe Karikatur_01_Projektstatus

4.5 Dokumentation

Die systematische und einheitliche Dokumentation projektrelevanter Überlegungen, Ereignisse und Ergebnisse ist Voraussetzung für die Planung, Steuerung, Durchführung und Evaluierung des Projekts sowie für die Weiterverwendung der Ergebnisse. Beispielweise wird es zur Einführung von Dokumenten-Management-Systemen bzw. Workflow-Management-Systemen oder zur Fortschreibung des Personalbedarfs benutzt. Sie ist unerlässlich für die Nachvollziehbarkeit der getroffenen Entscheidungen, des Projektvorgehens sowie im Fall von möglichem Personalzuwachs oder Personalwechsel innerhalb des Projektes.

Bezüglich der zielgerichteten und systematischen Dokumentation von Projekten sollte periodisch überprüft werden, ob die erstellten Dokumente bei der Planung, Steuerung und für die Umsetzung und Evaluierung der Ergebnisse tatsächlich nützlich sind oder angepasst werden müssen.

Dokumente sollten:

- lesbar und verständlich sein,
- eine klare Gliederung aufweisen,
- leicht identifizierbar sein (Angaben zu Inhalt, Zweck, Verfasser),
- datiert sein (inklusive Revisionsdaten),
- dem jeweiligen Nutzerkreis leicht zugänglich sein.

Durch den Einsatz von Informationstechnik entfällt inzwischen die Auswahl wesentlicher Daten zugunsten einer kompletten Archivierung aller Dokumente (inklusive Korrespondenz) auf geeigneten Massenspeichern. Ohne die entsprechende Strukturierung und Priorisierung ist ein solcher „Datenfriedhof“ jedoch häufig nutzlos.

Daher ist es sinnvoll, ein einheitliches Dokumentationssystem für alle Beteiligten einzuführen, das sich an den Phasen der Organisationsuntersuchung ausrichten kann. Am besten eignet sich dafür eine elektronische Dokumentenverwaltung auf einem zentralen Server im Intranet oder im Internet.

Vorteile sind:

- Die Dokumente sind für das Team von überall her zugänglich.
- Die Dokumente können bei Bedarf jederzeit ausgedruckt werden.
- Die Suchfunktionen erleichtern das Auffinden bestimmter Dokumente und Inhalte.
- Regelmäßige Sicherungen erhöhen die Verfügbarkeit und die Sicherheit.

Die systematische Dokumentation von Informationen und Ergebnissen wird in Organisationsuntersuchungen häufig vernachlässigt. Gründe dafür können sein, dass:

- der Aufwand zu hoch erscheint.
- viele Dokumente scheinbar unbrauchbar oder ohne erkennbare Funktion sind.
- andere Aufgaben im Projekt wichtiger oder dringender erscheinen.
- nicht klar ist, welches System sich für die Dokumentenverwaltung eignet.
- viel Disziplin notwendig ist, um ein Dokumentationssystem zu pflegen.

Einheitliche Vorlagen für die verschiedenen Typen von Dokumenten erleichtern dem Projektteam und den Adressaten die Übersicht. Wenn von Anfang an klare Regeln über die Dokumentenverwaltung eingeführt werden, kann ein großes Maß an Aufwand für die Suche von Informationen gespart werden. Dadurch werden die Berichterstattung sowie die Kommunikation nach innen und außen erleichtert.

Konkrete Aktionen zur Einrichtung eines leistungsfähigen Dokumentensystems können sein:

- Bestimmung der Art der Protokolle (Gesprächsprotokolle, Sitzungsprotokolle, Beschlussprotokolle) und einheitliche Darstellung der Dokumente mittels Seitenlayout- und Druckformatvorlagen sowie Kennzeichnung (Datum der letzten Änderung, Autor/-in, Versionsnummer etc.),
- Festlegung von Regeln für das Kennzeichnen, Sammeln, Aktualisieren, Archivieren und Vernichten projektrelevanter Dokumente.

Die Dokumentation dieses Projekts sollte die folgenden Dokumente enthalten:

- Projektauftrag (Ausgangslage, Zielsetzung),
- Wirtschaftlichkeitsbetrachtung (Kostenüberwachung),
- Projektplan (Projektstrukturplan, Aufwandsschätzung, Zeit- und Kostenplanung),
- Projektergebnisse (gegliedert anhand des Projektstrukturplans),
- Projektkommunikation,
- Protokolle, Zwischenberichte und den Abschlussbericht.

4.6 Zusammenfassung Projektdurchführung

Unter Projektdurchführung versteht man die Durchführung der einzelnen Vorbereitungsschritte sowie den eigentlichen Umbau der Anlagen bzw. die Erstinbetriebnahme der VSG1 Anlage nach dem Umbau. Zu den Vorbereitungsschritten gehören sowohl die Projektplanung, wie Zeitplanung oder Materialbereitstellung für den Umbau, als auch bestimmte Umbauschritte, die im laufenden Betrieb bereits durchgeführt werden können, wie z. B. die Bereitstellung eines neuen Schaltschranks für die S7 Steuerungskomponente etc. Eine ausführliche Projektplanung sowie ausreichendes Projektcontrolling ist für eine reibungsfreie Projektdurchführung zwingend notwendig.

Die Erstellung von Zeitplänen, einer Risikoanalyse des Projekts sowie die Erstellung des Lastenhefts und der Projektdokumentation gehören zu den wesentlichen Aufgaben einer Projektdurchführung. Im Rahmen des Praktikums gehörten die Erstellung von Lastenheft und Risikoanalyse des Projekts sowie die Projektdokumentation zu meinem Aufgabenbereich. Für die Erstellung des Lastenhefts habe ich mich auf folgende Normen und Standards gestützt:

- DIN 69901-5: Projektmanagement - Projektmanagementsysteme - Teil 5: Begriffe
- VDI 2519 Blatt 1: Vorgehensweise bei der Erstellung von Lasten-/Pflichtenheft
- VDI 2519 Blatt 2: Lastenheft/Pflichtenheft für den Einsatz von Förder- und Lagersystemen
- VDI 3694: Lastenheft/Pflichtenheft für den Einsatz von Automatisierungssystemen
- VDI 4403 Blatt 1 Entwurf: Modernisierung und Erweiterung fördertechnischer Anlagen und logistische Systeme bei laufendem Betrieb
- VDI 4403 Blatt 2 Entwurf: Modernisierung und Erweiterung fördertechnischer Anlagen und logistische Systeme bei laufendem Betrieb, Umbau und Erweiterung eines großen Zentrallagers
- IEEE 830-1998: Software Requirements Specification, Lastenheft für Softwaresysteme

Mit Hilfe meines Betreuers, Herrn Matthias Dettmer, habe ich versucht, alle benötigten Arbeitspakete vollständig zu identifizieren und strukturiert in einer Übersicht darzustellen. Dabei verwendete ich das sogenannte „Top to Down“ Prinzip¹. Das bedeutet, dass man zuerst von einer globalen Betrachtung ausgeht und diese dann von „oben“ nach „unten“ zunehmend

¹ „Top to Down“ Prinzip: <http://wirtschaftslexikon.gabler.de/Archiv/16275/top-down-prinzip-v6.html>

konkretisiert. Das heißt im Klartext, dass das Gesamtprojekt in Teilprojekte unterteilt wird, diese wiederum evtl. in weitere Teilprojekte etc. Dies ist ein besonderer Schritt, um den Projektfortschritt kontrollieren zu können, außerdem erleichtert er die Projektdurchführung bzw. die eigentliche Inbetriebnahme später und trägt zusätzlich etwas dazu bei, unnötiges Risiko zu vermeiden, wie z. B. fehlende Umbaumaterial usw. Ein Beispiel für die Unterteilung können Sie aus Abbildung aus dem Unterpunkt 4.2 entnehmen.¹

Dabei ist zu beachten, dass der Umbau VSG1 ein Testprojekt ist und als Modell für die spätere Modernisierung anderer Anlagenteile dienen sollte. Hinsichtlich dieses Schwerpunkts wurde die Dokumentation bzw. die Ausführung der einzelnen Arbeitsschritte so gehandhabt, dass die Arbeitsmuster im Sinne der Einheitlichkeit und Vereinfachung als Vorlage verwendet werden können.

¹ Weitere Beispiel zur Unterteilung siehe Anhang_02_Inbetriebnahmedokument, Tabelle 2 „Zeitplan“

4.6.1 Analyse der Projektdurchführung

Durch die Analyse der Projektdurchführung stellt sich heraus, dass wir eine „unklare“ Projektstruktur besitzen. Eine klare durchstrukturierte Struktur kann ein entscheidender Grund für den Erfolg eines Projekts sein.

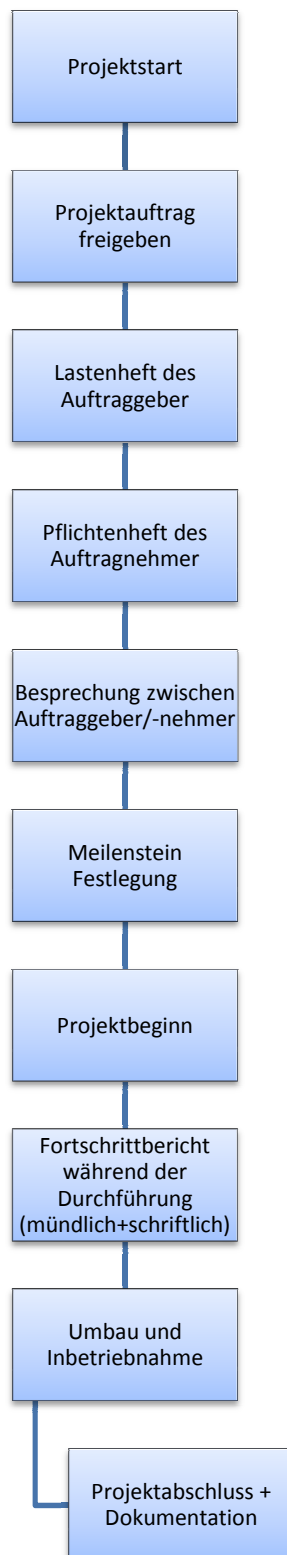


Abbildung 4-6 Projektstruktur "Soll" Form

Das Lastenheft wird vom Auftraggeber des Projektes formuliert. Es präzisiert und ergänzt den Projektauftrag. Im Lastenheft werden alle Anforderungen beschrieben, die der Auftraggeber an die Erreichung des Projektzieles stellt. Darüber hinaus werden die Rahmenbedingungen beschrieben, unter denen das Projekt arbeiten soll. Die Form und der Inhalt des Lastenheftes sind projektspezifisch individuell zu gestalten. Trotzdem bietet sich folgende allgemeine Gliederung an, die sicherstellt, dass wesentliche Punkte nicht vergessen werden:

1. Ausgangssituationsbeschreibung
Warum überhaupt ein Projekt?
Beschreibung des Ist-Zustands!
2. Zielsetzung
Was genau soll am Ende des Projektes entstanden sein?
3. Produkteinsatz
Unter welchen Rahmenbedingungen soll das Produkt zum Einsatz kommen?
4. Funktionale Anforderung
Welche Funktion muss realisiert werden?
5. Nichtfunktionale Anforderungen
Produkterweiterbarkeit sowie Zuverlässigkeitsbestimmung
6. Lieferumfang
Was genau soll in welcher Form geliefert werden?
7. Phasenplanung
Meilensteinfestlegung
8. Offene Punkte, die noch zu klären sind
Organisatorische Themen wie Lastenheftänderung sowie vertragliche Bedingungen
9. Abnahmekriterien
Qualitätsanforderung

Nach dem Lastenheft erfolgt die Erstellung des Pflichtenhefts vom Auftraggeber. Es dient der schriftlichen Dokumentation der Ergebnisse der Problemanalyse und ist die Grundlage für das weitere Vorgehen auf beiden Seiten. Es sagt dem Entwicklungsteam möglichst genau, welche Anforderungen gestellt werden, und dem Auftraggeber, welche Erwartung er haben darf. Das Pflichtenheft beinhaltet normalerweise folgende Punkte:

1. Beschreibung des zu entwickelnden Systems
(Was soll gemacht werden?)
2. Das zu entwickelnde System muss genau beschrieben werden.
3. Das gewünschte Verhalten und die Schnittstellen müssen genau spezifiziert werden.
4. An der Entwicklung beteiligte Gruppen (Wer macht es für wen?)
5. Auftraggeber und Auftragnehmer sollten nochmal explizit aufgeführt werden.
6. Welche Voraussetzungen müssen berücksichtigt werden?
7. Welche Vorleistungen liefert der Auftraggeber?
8. Welche Hardware soll verwendet werden?
9. Wie viel Entwicklungszeit steht zur Verfügung?

Nachdem auch das Pflichtenheft erstellt und akzeptiert wird, sollte für gewöhnlich das erste Meeting zwischen Auftraggeber und Auftragnehmer stattfinden, um die letzten offenen Punkte zu klären sowie die Meilensteine des Projektes endgültig festzulegen. Die Festlegung

der Meilensteine ist ein hilfreiches Mittel für das Fortschrittcontrolling während der Projektdurchführung. Erst nachdem alle Punkte abgeklärt sind, kann mit der Projektdurchführung begonnen werden.

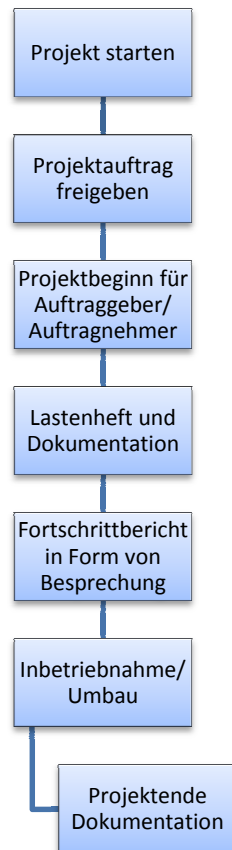


Abbildung 4-7 Projektstruktur "IST" Form

Nach der Information, die ich erhalte, existiert bis zu Beginn meines Praktikums weder ein vollständiges Lastenheft, noch ein Pflichtenheft. Aber es gibt ein spärlich formuliertes Lastenheft vom Auftragnehmer, der die Steuerungsmodernisierung für mehrere Anlagen beschreibt, inklusive des Umbaus der VSG1 Anlage. Dadurch ergibt sich, dass das Entwicklerteam nur teilweise weiß, was zu realisieren ist. Dadurch wurde wertvolle Zeit vergeudet, um über fehlende Ziele zu diskutieren, zu ändern bzw. realisieren zu lassen. Diese Unterbrechungen hindern den Projektfortschritt ungemein. Zudem existiert weder eine Zeitplanung, noch eine Meilensteinfestlegung, was das Projektcontrolling erschwert, vor allem weil der Großteil unsere Projektteilnehmer wenig bzw. keine Erfahrung bezüglich Zusammenarbeit mit extern Firmen im Rahmen eines Projektes besitzen.

Dieses Projekt findet seit April 2012 statt. Es war mehr als genug Zeit vorhanden, um das Projekt zum Erfolg zu führen. Das Projekt hätte von Anfang an richtig geplant und konsequent Schritt für Schritt durchgeführt werden müssen. Aufgrund mangelnder Erfahrung unsererseits haben wir zu sehr auf unsere Partnerfirmen vertraut und ungenaue, kritische Hinweise zur Projektdurchführung wurden nicht wahrgenommen, was uns zum Verhängnis wurde.

5 Software Überprüfung

5.1 Einführung

Ein Softwaretest prüft und bewertet die Software auf Erfüllung der für ihren Einsatz definierten Anforderungen und misst ihre Qualität. Die gewonnenen Erkenntnisse werden zur Erkennung und Behebung von Softwarefehlern genutzt. Tests während der Softwareentwicklung dienen dazu, die Software möglichst fehlerfrei in Betrieb zu nehmen.

Bei der Software-Überprüfung stehen wir vor drei grundlegenden Fragen:

- Welche Funktion muss das Programm erfüllen?
- Wie soll die Funktion dargestellt werden?
- Und wie soll es geprüft werden?

Jede einzelne Frage sollte vor der eigentlichen Überprüfung getrennt voneinander beantwortet und dokumentiert werden. Wobei die Antworten für die ersten zwei Fragen bereits im Lastenheft enthalten sein sollten. Sodass man sich nur noch die Frage stellt, wie man es optimal prüfen kann. Der Prüfer soll von der Entwicklungsmannschaft abgesondert werden. Seine Daseinsberechtigung besteht darin, die entwickelte Software zu verifizieren¹ und validieren². Die Nähe zum Entwicklerteam kann zur Folge haben, dass der Prüfer zwecks Konfliktvermeidung nicht mehr so genau testet, wie er es sollte.

Aber was versteht man eigentlich unter dem Begriff Software? Der Begriff Software ist ein Sammelbegriff für ausführbare Programme und die zugehörigen Daten. Die Software erledigt Aufgaben, indem sie von einem Prozessor ausgewertet wird und so softwaregesteuerte Geräte in ihrer Arbeit beeinflusst. Im unserem Fall ist Software also ein Maschinencode, die der SPS braucht, um die Anlagen steuern zu können.

Im Rahmen des Projektes wird das neue S7 Programm von der Firma Symacon fertiggestellt. Unsere Aufgabe besteht darin, die gelieferte Software auf ihre gewünschte und notwendige Spezifikation zu überprüfen. Eine Analyse des Ist-Programms ist dabei zwingend notwendig, um die derzeitig genutzten Spezifikationen zu identifizieren und dokumentieren. Bei Produktionsanlagen ist es empfehlenswert, eine Ablaufschrittkette zu erstellen, die man mit der neuen zu realisierenden Schrittkette vergleichen kann.

¹Verifikation: "Professionelle Softwaretests" von Marcel Alper; Seite 37

²Validierung: "Professionelle Softwaretests" von Marcel Alper; Seite 37

5.2 Allgemeine Softwaretestverfahren

In kaum einer Disziplin der Softwareentwicklung hat sich, der Komplexität der Aufgabe 'Testen' entsprechend, eine derart große Vielfalt an Begriffen für Verfahrensansätze gebildet wie beim Softwaretest. Dies beginnt bereits bei den Typ-Ausprägungen für Testvarianten, die mit Begriffen, wie Teststufe, Testzyklus, Testphase, Testmethode, Testverfahren bezeichnet werden.

Die Bezeichnung **konkreter Testarten** leitet sich meistens aus ihren individuellen Zielen und Charaktermerkmalen ab, wodurch sich eine Vielzahl an Bezeichnungen ergibt. Dieser Violdimensionalität entsprechend können für einen konkreten Test die Bezeichnungen mehrerer Testarten zutreffen. Z. B.: Entwicklertest, dynamischer Test, Blackbox-Test, Fehlertest, Integrationstest, Äquivalenzklassentest, Batchtest, Regressionstest. Es ist durchaus im Sinn effizienter Testprozesse, mehrere Testfälle mit nur einem konkreten Test abzudecken, z. B. eine technische Datenschnittstelle, die Prüfung korrekter Wertebereiche und eine Rechenformel.

5.3 Nomenklatur

5.3.1 IEEE Standard

Es ist zu vermerken, dass es bisher keine internationalen Normen gibt, die die Software-Überprüfung beschreiben. Die meisten Software-Tests basieren auf dem Standard, die vom IEEE¹ (Institute of Electrical and Electronic Engineers) veröffentlicht wurden.

Zum Beispiel beschreibt der IEEE 829 „Standard for Software Test Documentation“ die Form und den Inhalt der acht Basis-Dokumente zur Dokumentation von Software-Tests. Er schreibt jedoch nicht vor, welche der jeweiligen Dokumente zwingend verwendet werden müssen. Die acht Dokumente lassen sich wie folgt in drei Kategorien unterteilen:

1. Übersicht

- Testkonzept (test plan): Das Testkonzept bestimmt Abgrenzung, Vorgehensweise, Mittel und Ablaufplan der Testaktivitäten. Es bestimmt die Elemente und Produktfunktionen, die getestet werden sollen, die Testaufgaben, die durchgeführt werden müssen, das verantwortliche Personal für jede Aufgabe und das Risiko, das mit dem Konzept verbunden ist.

2. Test-Spezifikation (test specification)

- Testentwurfsspezifikation (test design specification): Die Testentwurfsspezifikation verfeinert die Beschreibung der Vorgehensweise für das Testen der Software. Sie identifiziert die Produktfunktionen, die von den Tests abgedeckt werden müssen. Weiteres beschreibt sie die Testfälle und Testabläufe, die benötigt werden, um Tests zu bestehen und spezifiziert die Bestehens- oder Verfehlenskriterien der einzelnen Produktfunktionen.
- Testfallspezifikation (test case specification): Die Testfallspezifikation dokumentiert die zu benutzenden Eingabewerte und erwarteten Ausgabewerte. Testfälle sind vom Test-Design getrennt. Dies erlaubt die Verwendung der Testfälle in mehreren Designs sowie die Wiederverwendung in anderen Situationen.
- Testablaufspezifikation (test procedure specification): Beschreibung aller Schritte zur Durchführung der spezifizierten Testfälle und Implementierung des zugehörigen Test-Designs.

¹IEEE-SA Standards Board: IEEE Standard for Software Test Documentation. Piscataway NJ, 1998: The Institute of Electrical and Electronic Engineers Inc;

3. Testbericht (test reporting)

- Testobjektübergabebericht (test item transmittal report): Der Bericht beschreibt die Übergabe der Testfälle für den Fall, dass getrennte Entwicklungs- und Testteams eingebunden sind, oder für den Fall, dass ein offizieller Zeitpunkt für den Beginn einer Testausführung erwünscht ist.
- Testprotokoll (test log): Das Testprotokoll dient zur Aufzeichnung der Ereignisse während einer Testausführung.
- Testabweichungsbericht (test incident report): Es beschreibt alle Ereignisse, die während einer Testausführung auftreten und weitere Nachprüfungen erfordern.
- Testabschlussbericht (test summary report): Fasst die Testaktivitäten zusammen, die mit einer oder mehreren Testentwurfsspezifikationen zusammenhängen.

Weitere IEEE Standards, die der Software-Test beschreibt, wären:

- | | |
|--|-----------|
| • SQAP Software Quality Assurance Plan | IEEE 730 |
| • SCMP Software Configuration Management Plan | IEEE 828 |
| • STD Software Test Documentation | IEEE 829 |
| • SRS Software Requirements Specification | IEEE 830 |
| • SUT Standard for Software Unit Testing | IEEE 1008 |
| • SVVP Software Validation & Verification Plan | IEEE 1012 |
| • SDD Software Design Description | IEEE 1016 |
| • SPMP Software Project Management Plan | IEEE 1058 |

5.3.2 ISO IEC 29119

Für das Jahr 2012/2013 ist die Veröffentlichung des neuen internationalen Standards **ISO/IEC 29119**¹ „Software and Systems Engineering-Software Testing“ geplant. Ziel der ISO IEC 29119 ist, bestehende etablierte Standards aus dem Bereich des Tests abzulösen:

- **BS 7925-1** (Software Testing – Vocabulary of terms in software testing)
Definiert grundlegende Testbegriffe.
- **BS 7925-2** (Software Testing – Software component testing)
Definiert Methoden zur Ableitung von Komponententestfällen.
- **IEEE 829** (Standard for Software Test Documentation)
Definiert die Dokumentation von Softwaretests und liefert Templates für die Erstellung von Dokumentationen.
- **IEEE 1008** (Standard for Software Unit Testing)
Definiert grundlegende Methoden des Komponententests.

¹<http://softwaretestingstandard.org/> Suchbegriff: IEC 29119

Diese sowie weitere Standards im Umfeld des Softwaretestens und der Testprozessverbesserung haben sich im Laufe der letzten Jahre teilweise parallel und unabhängig voneinander entwickelt und sind daher nicht immer miteinander kompatibel.

Die ISO / IEC 29119 Software Testing hat das Ziel, einen definitiven Standard der Testbegriffe, des Testprozesses, der Testdokumentation und Testtechniken für den gesamten Software-Test-Lebenszyklus gebündelt zu definieren und aufeinander abzustimmen. Bewährtes soll verwendet werden und in der neuen Norm aufgehen. Der Standard will einen Rahmen bieten, um das Testen in der Softwareentwicklung und Wartung zu unterstützen. Dazu gehören die Anwendung von Testrichtlinien und Teststrategien für die Planung, die Testanalyse und den Entwurf, die Realisierung und Testausführung, die Auswertung und Berichterstattung sowie die Abschlussaktivitäten im Rahmen des Testprozesses.

Die Struktur der ISO/IEC 29119 wird in vier Hauptteile gegliedert:

1. Konzepte und Definitionen (Concepts and Definitions)

- Ziel von Teil 1 der Norm ist, einen Überblick über den Standard und die grundsätzlichen Softwaretestkonzepte sowie ein Glossar zu Softwaretest-Begriffen des gesamten Software-Test-Lebenszyklus zu geben.

Geplante Inhalte:

- Einführung in das Softwaretesten
- Testen von Software im organisatorischen Kontext
- Testprozess
- Beziehung zwischen Test und Entwicklung
- Auswirkungen von Softwarelebenszyklusmodellen auf das Testen
- Testarten, Testtechniken und Teststufen
- Risikobasiertes Testen
- Testobjekte
- Verifikation von Anforderungen
- Validierung von Anforderungen

2. Testprozess (Test Process)

- Teil 2 der Norm beschreibt ein generisches Testprozessmodell, das im Rahmen jedes Softwareentwicklungs- und Testlebenszyklus angewendet werden kann. Dieses Prozessmodell gliedert sich in vier weitere Teilprozesse:
 - Organisatorische Testprozesse (z.B. Testrichtlinie und Teststrategie)
 - Testmanagement-Prozesse
 - Fundamentale statische Testprozesse
 - Fundamentale dynamische Testprozesse

3. Testdokumentation (Test Documentation)

- Teil 3 der Norm deckt die Testdokumentation über den gesamten Software-Test-Lebenszyklus ab. Dazu gehören Vorlagen für alle Bereiche Testprozesses nach ISO/IEC 29119:
 - Organisatorische Testprozesse – Testrichtlinie
 - Organisatorische Testprozesse – Teststrategie
 - Testmanagement-Prozesse
 - Fundamentaler Testprozess
- Die International Organization for Standardization (ISO) hat vom Institute of Electrical and Electronics Engineers (IEEE) die Erlaubnis erhalten, den bekannten IEEE 829 Test Documentation Standard als Grundlage für diesen Teil der neuen Norm zu verwenden.

4. Testtechniken (Test Techniques)

- Teil 4 der Norm wird die Softwaretestverfahren für alle Arten von Tests abdecken, einschließlich der statischen (Reviews, Inspektionen, Walkthroughs), funktionalen (Black-Box, White-Box), nicht-funktionalen (Effizienz, Sicherheit, Benutzerfreundlichkeit) und erfahrungsbasierten Tests (Intuitive Testfallermittlung, Exploratives Testen).
- Die British Computer Society hat der Organization for Standardization (ISO) die Erlaubnis erteilt, die BS-7925-1/2 Component Testing Standard als Grundlage für diesen Teil des Standards zu verwenden.

5.4 Teststrategie

Es gibt zwei Integrationsstrategien, die vom Modul über das Programm zur Anwendung führen können, die Top-down Methode und die Bottom-up Methode. Das Top-down Prinzip geht vom Abstrakten, Allgemeinen, Übergeordneten schrittweise hin zum Konkreten, Speziellen, Untergeordneten „von oben nach unten“. Das Bottom-up Prinzip bezeichnet die umgekehrte Richtung „von unten nach oben“. Es sind zwei grundsätzlich verschiedene Denkrichtungen, um komplexes Sachverhalten zu verstehen, zu beschreiben und darzustellen.

Zu dieser Thematik möchte ich noch einige Argumente nennen, was vom Verfechter der jeweiligen Theorie benutzt wird, um ihre Standpunkte zu verfestigen, die jedoch in gewisser Hinsicht sehr fragwürdig sind.

Bottom-Up Argumente¹:

Wenn der einzelne Teil funktioniert, dann funktioniert auch das Ganze.

Aus eigener Erfahrung heraus kann ich bestätigen, dass dies nur ein frommer Wunsch ist. Die Interaktionen von Modulen untereinander bergen viele Gefahren, selbst bei einer gut strukturierten Programmierung. Es entsteht ein unvermeidbarer Fehler beim Interagieren der Module miteinander. Die Fehler unterscheiden sich in ihrem Ausmaß und der Beeinträchtigung auf das System als Ganzes, was wiederum von der Art und Herangehensweise der Programmierung abhängt.

Die Komplexität wächst von unten nach oben. Daher löst die Methode die einfachsten Probleme zuerst.

Auch aus Erfahrung heraus muss ich leider sagen, dass Fehler eher zufällig als systematisch bedingt auftreten. Noch dazu neigen entstandene Fehler dazu, sich über das ganze System zu ziehen. Was steigende Komplexität angeht, kann ich da auch nicht zustimmen. Es ist richtig, dass die Komplexität zunächst zunimmt, jedoch bis zu gewissen Punkt sinkt sie wieder, bevor sie die oberste Ebene erreicht.

Es gibt einen einheitlichen Bottom!?

Unter Betrachtung des dargestellten Beispiels (Abbildung 5-3) ergibt sich die Frage: Haben wir einen gemeinsamen Bottom? Ich kann mir nicht vorstellen, dass es Programm gibt, das einen einheitlichen „Bottom“ besitzt.

¹ Siehe „Professionelle Softwaretests“ von Marcel Alper, Seite 114 „Bottom up“

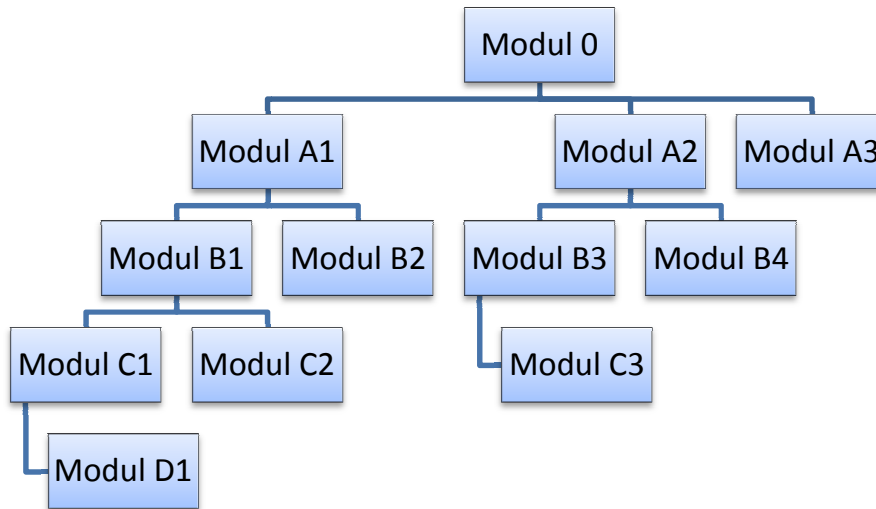


Abbildung 5-1 Programmstruktur-Beispiel

Top-Down Argumente¹:

Die meisten schweren Fehler betreffen Ablaufprobleme.

Auch aus Erfahrung kann ich behaupten, dass das Ablaufproblem eher an einer unklaren Strukturierung liegt. Und solange die Struktur nicht klar wird, wird man das Ablaufproblem nicht lösen können, egal ob man Bottom-up oder Top-down verwendet.

Die Komplexität verringert sich von oben nach unten. Das Top-Level ist das komplexeste.

Wie bereits vorhin gesagt, ist das eher ein Trugschluss. Zur Mitte hin wird die Komplexität wachsen, sinkt jedoch wieder je tiefer man kommt.

Meiner Meinung nach muss man beide Methoden mindestens einmal verwenden, um ein anständiges Ergebnis für den Softwaretest zu erlangen. Dies ist jedoch bei steigender Komplexität von Programmen und dem daraus resultierenden zeitlichen Bedarf sowie den Kostengründen für einen vollständigen Test fast unmöglich.

Hinsichtlich der noch „relativ geringen“ Komplexität unseres Programms im Rahmen des Projektes wäre zu empfehlen, beide Methoden für einen sicheren Softwaretest zu benutzen. Doch aufgrund der Unvollständigkeit des erhaltenen S7-Programms ist eine Top-down Analyse eher schwierig durchzuführen. Mit Bottom-up hingegen kann man die einzelnen Module gut verifizieren und validieren.

¹ Siehe „Professionelle Softwaretests“ von Marcel Alper, Seite 115 „Top Down“

5.5 Testverfahren: Black-Box-Test und White-Box-Test

Im Groben und Ganzen unterteilt man den Softwaretest unter **Black-Box-Test** und **White-Box-Test** bzw. **Greys-Box-Test**.¹

Der **Black-Box-Test** bezeichnet eine Methode des Softwaretests, bei der die Tests ohne Kenntnisse über die innere Funktionsweise des zu testenden Systems entwickelt werden. Er beschränkt sich auf funktionsorientiertes Testen, d. h. für die Ermittlung der Testfälle werden nur die Anforderungen, aber nicht die Implementierung des Testobjekts, herangezogen. Die genaue Beschaffenheit des Programms wird nicht betrachtet, sondern vielmehr als Black Box behandelt. Ziel ist, die Übereinstimmung eines Softwaresystems mit seiner Spezifikation zu überprüfen. Ausgehend von formalen oder informalen Spezifikationen werden Testfälle erarbeitet, die sicherstellen, dass der geforderte Funktionsumfang eingehalten wird. Das zu testende System wird dabei als Ganzes betrachtet, nur sein Außenverhalten wird bei der Bewertung der Testergebnisse herangezogen.

Der Begriff **White-Box-Test** (auch **Glass-Box-Test** genannt) bezeichnet eine Methode des Software-Tests, bei der die Tests mit Kenntnissen über die innere Funktionsweise des zu testenden Systems entwickelt werden. Im Gegensatz zum Black-Box-Test ist für diesen Test also ein Blick in den Quellcode gestattet, das heißt, es wird am Code geprüft. Es wird eingesetzt, um Fehler in den Teilkomponenten aufzudecken und zu lokalisieren, aber aufgrund ihrer Methodik ist dies kein geeignetes Werkzeug, Fehler gegenüber der Spezifikation aufzudecken. Für letzteres benötigt man Black-Box-Tests. Zu bedenken ist auch, dass zwei Komponenten, die für sich genommen korrekt gemäß ihrer jeweiligen Teilspezifikation arbeiten, zusammen nicht zwangsläufig eine korrekte Einheit gemäß der Gesamtspezifikation bilden. Dies kann durch Black-Box-Tests leichter festgestellt werden als durch White-Box-Tests.

¹ Siehe "Professionelle Softwaretests" von Marcel Alper; Seite 40 „9.2-Betrachtungsbezogen“

	Black-Box-Test	White-Box-Test
Vorteil	<ul style="list-style-type: none"> ➤ bessere Verifikation des Gesamtsystems ➤ Testen von semantischen Eigenschaften bei geeigneter Spezifikation ➤ Übertragung von systematisch erstellten Testsequenzen auf plattformunabhängigen Implementierungen 	<ul style="list-style-type: none"> ➤ Testen von Teilkomponenten und der internen Funktionsweise ➤ Geringerer organisatorischer Aufwand ➤ Automatisierung durch gute Tool-Unterstützung
Nachteil	<ul style="list-style-type: none"> ➤ größerer organisatorischer Aufwand ➤ zusätzlich eingefügte Funktionen bei der Implementierung werden nur durch Zufall getestet ➤ Testsequenzen einer unzureichenden Spezifikation sind unbrauchbar 	<ul style="list-style-type: none"> ➤ Erfüllung der Spezifikation nicht überprüft ➤ Eventuell Testen „um Fehler herum“

Tabelle 7 Vergleich Black-Box-Test und White Box-Test

Zudem sei gesagt, dass die Unterscheidung zwischen Black-Box-Test und White-Box-Test teilweise von der Perspektive abhängt. Das Testen einer Teilkomponente ist aus Sicht des Gesamtsystems ein White-Box-Test, weil für das Gesamtsystem aus der Außenperspektive keine Kenntnisse über den Systemaufbau und damit die vorhandenen Teilkomponenten vorliegen. Aus Sicht der Teilkomponente wiederum kann derselbe Test unter Umständen als Black-Box-Test betrachtet werden, wenn er ohne Kenntnisse über die Interna der Teilkomponente entwickelt und durchgeführt wird.

Grey-Box-Tests sind ein Ansatz aus dem Extreme Programming, mithilfe testgetriebener Entwicklung die gewünschten Vorteile von Black-Box-Tests und White-Box-Tests weitgehend miteinander zu verbinden und gleichzeitig die unerwünschten Nachteile möglichst zu eliminieren.

5.6 Teststufen bzw. Testzyklen

Unabhängig von den Benutzen von den oben genannten Testverfahren, lässt sich im Rahmen einer Projektdurchführung der Softwaretest in vier unterschiedliche Teststufen unterteilen. Dabei wird das sogenannte V-Modell gern benutzt, um den Zusammenhang zwischen Projektmeilenstein und Teststufen¹ darzustellen.

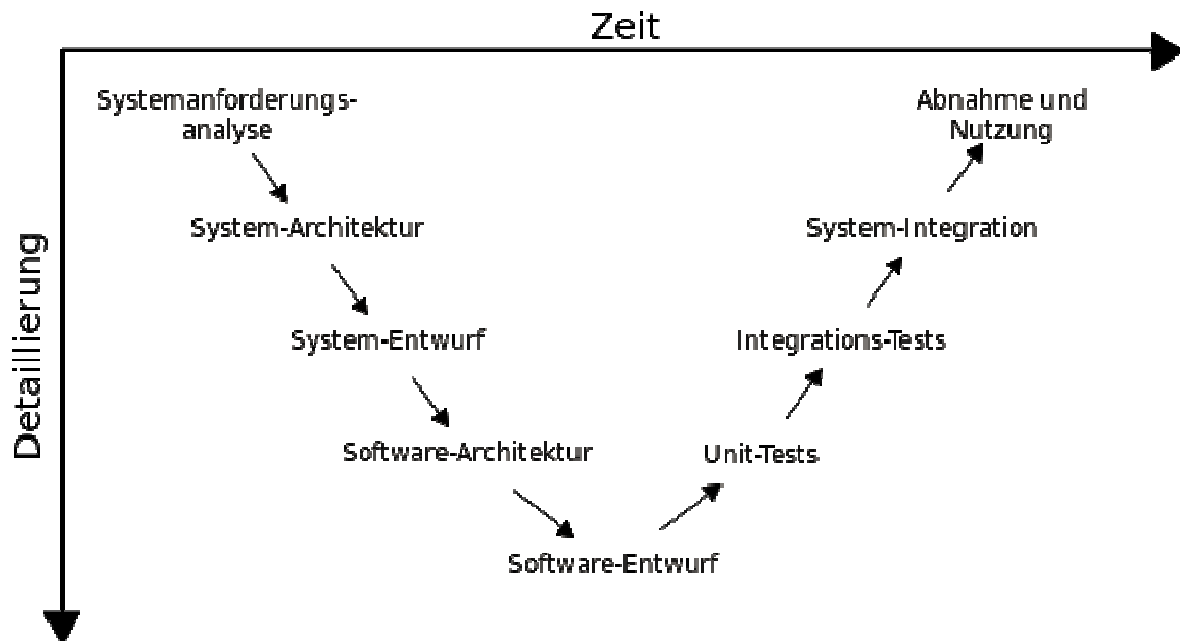


Abbildung 5-2 V-Modell

Komponententest: Der Modultest, auch Komponententest oder Unittest genannt, ist ein Test auf der Ebene der einzelnen Module der Software. Testgegenstand ist die Funktionalität innerhalb einzelner abgrenzbarer Teile der Software (Module, Programme oder Unterprogramme, Units oder Klassen). Ziel dieser häufig durch den Softwareentwickler selbst durchgeführten Tests ist der Nachweis der technischen Lauffähigkeit und korrekter fachlicher (Teil-) Ergebnisse.

Integrationstest: Der Integrationstest bzw. Interaktionstest testet die Zusammenarbeit voneinander abhängiger Komponenten. Der Testschwerpunkt liegt auf den Schnittstellen der beteiligten Komponenten und soll korrekte Ergebnisse über komplette Abläufe hinweg nachweisen.

¹ Teststufen: "Professionelle Softwaretests" von Marcel Alper; Seite 101 „11. Testphasen“

Systemtest: Der Systemtest ist die Teststufe, bei der das gesamte System gegen die gesamten Anforderungen (funktionale und nicht funktionale Anforderungen) getestet wird. Gewöhnlich findet der Test auf einer Testumgebung statt und wird mit Testdaten durchgeführt. Die Testumgebung soll die Produktivumgebung des Kunden simulieren. In der Regel wird der Systemtest durch die realisierende Organisation durchgeführt.

Abnahmetest: Ein Abnahmetest, Verfahrenstest, Akzeptanztest oder auch User Acceptance Test (UAT) ist ein Test der gelieferten Software durch den Kunden bzw. Auftraggeber. Oft sind Akzeptanztests Voraussetzung für die Rechnungsstellung. Dieser Test kann bereits auf der Produktivumgebung mit Kopien aus Echtdaten durchgeführt werden.

Die vorgenannten Teststufen sind in der Praxis oft nicht scharf voneinander abgegrenzt, sondern können, abhängig von der Projektsituation, fließend oder über zusätzliche Zwischenstufen verlaufen. So könnte z. B. die Abnahme des Systems auf der Grundlage von Testergebnissen (Reviews, Testprotokolle) von Systemtests erfolgen.

Besonders für System- und Abnahmetests wird das Blackbox-Verfahren angewendet, das heißt, der Test orientiert sich nicht am Code der Software, sondern nur am Verhalten der Software bei spezifizierten Handlungen (Eingaben des Benutzers, Grenzwerte bei der Datenerfassung etc.).¹

¹ Für weitere möglichen Teststufen siehe: "Professionelle Softwaretests" von Marcel Alper; Seite 102-110

5.7 Testverlauf

Folgende Herangehensweise wurde von mir entwickelt und im Rahmen des Softwaretests durchgeführt:

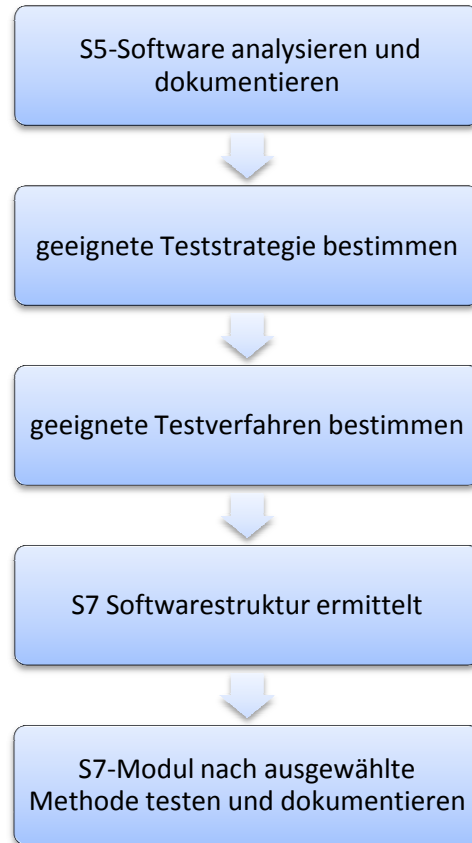


Abbildung 5-3 durchgeführte Testablauf

Zu vermerken ist, dass je nach Funktionalität der Module ein anderes Testverfahren ausgewählt werden muss, um es optimal prüfen zu können. Als Beispiel wäre die Überprüfung der Schrittkette in S7-Graph zu nennen. Bei dem wäre ein Black-Box-Test nicht zu empfehlen, weil der einzelne Schritt genauestens untersucht werden sollte. Ein weiteres Beispiel hierzu ist der Überprüfung der WinCC Oberfläche. Hier wiederum ist der Black-Box-Test dem White-Box-Test vorzuziehen. Denn hier spielt die Programmierung an sich eine untergeordnete Rolle, die Funktion des Moduls als Ganzes muss einfach funktionieren und seine Spezifikation erfüllen (Benutzerfreundlichkeit, Benutzerebene etc.). Die ausgewählte Teststrategie hingegen bleibt weitestgehend unverändert.

5.7.1 Software Spezifikation des Projekts

Unter Spezifikation versteht man die Eigenschaft der Programmmodule hinsichtlich ihrer Aufbauweise und Funktionalität.

Das Programm sollte nach modularem Aufbau programmiert und dabei hauptsächlich Multiinstanz bzw. indirekte Adressierung verwendet werden. Dabei ist zu beachten, dass FUP die bevorzugte Programmierungssprache ist. Wenn sich ein Programmteil aufgrund des komplexen Aufbaus nicht in FUP realisieren lässt, sollte es in AWL stattfinden, z. B. für "indirekte Adressierung".

Die Schrittketten sind in S7-Graph zu programmieren und sollten mit WinCC Oberflächen zu beobachten sein, um die Fehlersuche zu erleichtern.

Für das Projekt VSG01 sind folgende Module zu programmieren:

- Modul für Rollenförderer
- Modul für den Kipptisch
- Modul für den Portalroboter, aufeinander synchronisierbare skalierbare Achsenfunktion, die auf folgende Achsen angewendet werden können (X-Achse, Y-Achse, Z-Achse, M-Achse, W-Achse, K-Achse)
- Blatzzählerfunktion
- Modul für Umrichter (Lenze / SEW)
- Schrittkette für Referenzfahrt / Grundstellungsfahrt und Ablaufsteuerung (S7 Graph)
- sowie eine entsprechende geeignete WinCC Oberfläche

Die W-Achse gehört nicht zur VSG1, wird aber aufgrund der Vollständigkeit und für spätere Benutzung in andere Anlagen mit programmiert.

5.7.2 Softwarestruktur

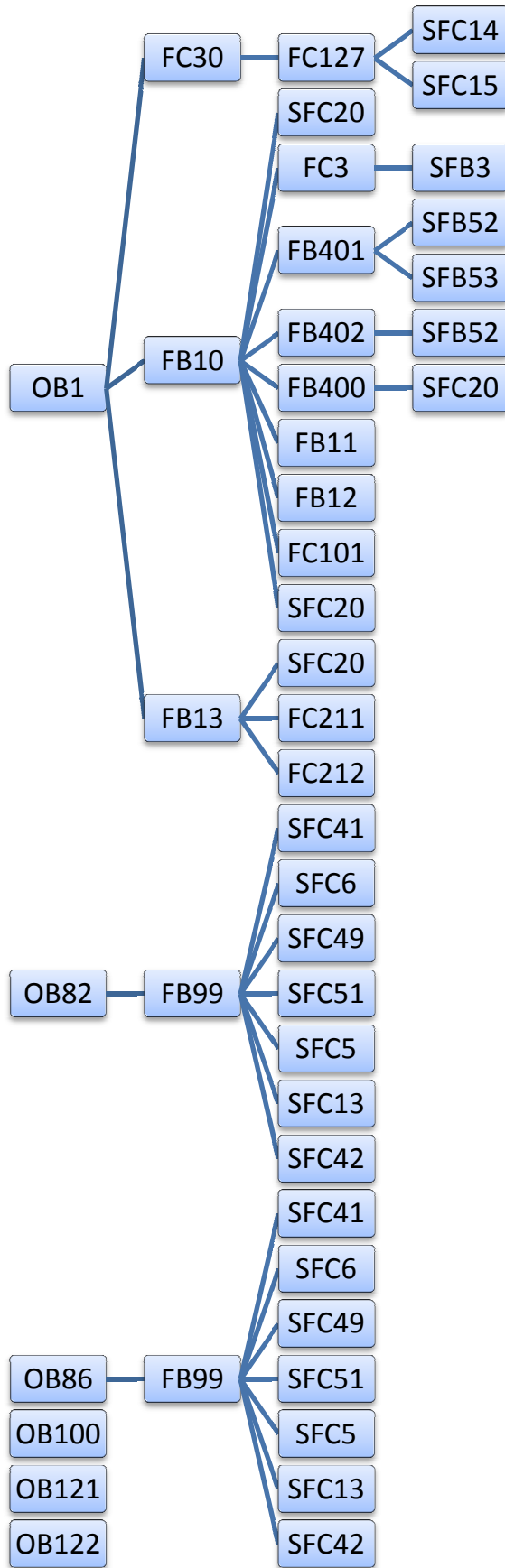


Abbildung 5-4 Programmstruktur aufgerufene Modulen Stand 12011_17

Modulname	Symbolname	Programmiersprache	Standardbaustein	Kommentar
OB1	Cycle Execution	FUP	Ja	Main Programm Sweep
OB82	OB_Diagnosealarm	FUP	Ja	
OB86	OB86-Dezentrale Periph.	FUP	Ja	
OB100	OB100-Neustart	AWL	Ja	
OB121	PROG_ERR	AWL	Ja	Programming Error
OB122	Modul Access Error	AWL	Ja	
FB10	ST10_Portal	FUP	nein	Achsensteuerung
FB11	ST10_Reset	Graph	Nein	Funktion_Reset
FB12	ST10_Kette	Graph	Nein	Schrittkette zur Ablaufsteuerung
FB13	ST10_Koordinierung	FUP	nein	Portal10_Steuerung
FB99	DP_DIAG_Detail	AWL	Nein	DP-Diagnose
FB400	FM351_BA	FUP	Ja	FM351 Modul
FB401	FM351_CTRL	SCL	Ja	FM351 Modul
FB402	FM351_DIAG	SCL	Ja	FM351 Modul
FB404	FM351_MD	AWL	Ja	FM35_Modul
FC3	FC3_Handbedienmerker	AWL	nein	Merker generieren
FC30	Lenze_BeispielFC30	AWL	ja	Antriebsreglerfunktionen
FC101	AUSG.1 Spule	FUP	nein	Arbeitsposition
FC102	AUSG.2 Spule	FUP	nein	M-Achse Positionierung
FC127	LenzebeispielFC	AWL	ja	DP-Parameterübertragung
FC211	Reihenfolge	AWL	nein	Station Reihenfolge Manipulation
FC212	BlockPruefFunk	AWL	nein	Eingabe Überprüfung

Tabelle 8 Modulbeschreibung, Stand 12011_17

SFC und SFB sind Standardfunktionen aus der Simatic Bibliothek und wurden deshalb nicht in der oben dargestellten Tabelle aufgeführt.

Zu erwähnen ist, dass es sich hier um eine unvollständige Version des S7 Programms handelt. Das heißt, die abgebildete Programmstruktur ist unvollständig. Es existieren zu diesem Zeitpunkt mehrere unbenutzte Bausteine im Programm, die, auf Abbildung 5-4, für eine bessere Übersicht nicht dargestellt wird. Sie werden für Testzwecke oder Reserve verwendet. Es steht außer Frage, dass diverse Bausteine dazugehören, UDT, Globale DB, FC und FB, die für eine bessere Strukturierung des Programms sowie eine bessere Kapselung der einzelnen Module noch zusammengefasst werden könnten. Durch diese Zusammenlegung der Bausteine wird sich die Programmstruktur zwangsweise noch verändern.

5.7.3 Programmtest

Im Rahmen der Projektdurchführung wird das Grey-Box-Test Verfahren verwendet, um das S7 Programm als Ganzes zu kontrollieren, also eine Mischung aus Black-Box-Test Verfahren und White-Box-Test Verfahren. Es war so geplant, dass das einzelne Modul entweder durch das Black-Box-Test Verfahren oder durch das White-Box-Test Verfahren getestet werden sollte, in Abhängigkeit von ihrer Funktionalität und ihrem Aufbau.

Dabei ist zu vermerken, dass das einzelne Modul nur auf ihre Funktionalität sowie die richtige Eingang/Ausgang Beschaltung überprüft werden sollte. Die Überprüfung der Source Codes, also die Fehlersucher sowie die Fehlerbehebung, gehört nicht zum Gegenstand der Überprüfung.

Entstandene Fehler beim Prüfen werden dokumentiert und je nach dem Ausmaß der Fehler wird das dem Entwickler sofort mitgeteilt. Unter Ausmaß der Fehler versteht man, ob es dabei um einfache Rechtschreibfehler oder falsche Namengebung geht bzw. ob der Fehler die Funktion des Moduls beeinflusst. Aufgrund rechtlicher Übereinkünfte, wie Garantie, sind sämtliche Rechte für Programmänderung und Korrektur dem Entwicklerteam vorenthalten.

Der Softwaretest erfolgt wie folgt:

- Trennung des Gesamtprogramms in einzelne Untermodule
- Überprüfung der einzelne Module auf notwendige Spezifikation
- Überprüfung der einzelne Module auf gewünschte zusätzliche Spezifikation
- Überprüfung der verwendete Eingänge und Ausgänge in Bezug zu bereits verwendeten Eingängen und Ausgängen (S5)
- Überprüfung der HMI Oberfläche auf ihre Funktionalität und Benutzerfreundlichkeit

Dabei wird die Überprüfung unter folgenden Kriterien durchgeführt:

- **Funktionale Tests bzw. Funktionstests:** Systemüberprüfung in Bezug auf funktionale Anforderungsmerkmale, wie Korrektheit und Vollständigkeit.
- **Schnittstellentests:** Funktionalitätsüberprüfung bei der Zusammenarbeit voneinander unabhängiger Komponenten
- **Fehlertests:** Systemüberprüfung im Fehlerfall
- **Datenkonsistenztests:** die Auswirkung der getesteten Funktion auf die Korrektheit von Datenbeständen sind zu überprüfen (Testbezeichnungen: Datenzyklustest, Wertebereichstest, Semantiktest)

- **Stresstests:** sind Tests, die das Verhalten eines Systems unter Ausnahmesituationen prüfen und analysieren.
- **Crash Test:** sind Stresstests, die versuchen, das System zum Absturz zu bringen.
- **Lasttests:** sind Tests, die das Systemverhalten, wie z. B. unter besonders hohen Speicher-, CPU Anforderungen, analysieren. Besondere Arten von Last-Tests können Multi-User-Tests und Stresstests (dabei wird das System an die Grenzen der Leistungsfähigkeit geführt) sein.

Wie bereits an mehreren Stellen genannt, handelt sich hier um eine unvollständige Version des S7 Programms. Dennoch möchte ich an einigen Beispielen erklären, wie die Module getestet werden können. Für einen einheitlichen strukturierten Softwaretest habe ich folgende Herangehensweise für den Test der einzelnen Module entwickelt. Es geht dabei um die Beantwortung folgender Fragen:

1. Wie soll/kann das S7 Programm als Ganzes geprüft werden?
Bottom-up Prinzip; Top-Down Prinzip; White-Box-Test; Black-Box-Test;
2. Welche Funktion soll das zu realisierende Modul beinhalten?
Analyse der S5 Modul;
3. Wie wurde der Modul auf S7 realisiert?
Programmiersprache; Kapselungsmöglichkeiten; Welcher Einfluss hat der Modul auf das Programm als Ganzes?
4. Wie soll/kann der Modul geprüft werden?
White-Box-Test; Black-Box-Test;
5. Dokumentation Modul Testbericht (jeder Modul einzeln)
Testnummer; gewählte Testverfahren; Programmversion; Testergebnis; Defizit;

Die Fragen werden in diese Reihenfolge beantwortet. Dabei muss die erste Frage nur einmal beantwortet werden, weil es prinzipiell für alle Module gilt. Das Programm und die dazugehörigen Module werden mehrere Male getestet und je nach erhaltener Softwareversion dokumentiert. Demzufolge existieren auch mehrere Testberichte. Die Testberichte lassen sich außerdem als eine Schätzung für den derzeitigen Projektfortschritt benutzen.

Beispiel : Überprüfung der neuen Schrittkette in Graph FB11 und FB12

1. Wie soll/kann das S7 Programm als Ganzes geprüft werden?

Da wir kein vollständiges Programm auf einmal bekommen, habe ich mich dazu entschieden die Bottom-Up Methode zu nehmen. Durch die Überprüfung der einzelnen Module von unten nach oben sowie den Vergleich mit frühen Berichten, lässt sich so eine relativ genaue Aussage über den Projektfortschritt machen.

2. Welche Funktion soll das zu realisierende Modul beinhalten?

Um den neuen Ablauf kontrollieren zu können, muss man natürlich erstmal wissen, was die alte Ablaufkette auf S5 beinhaltet. Dabei habe ich das S5-Programm analysiert und eine grobe Schrittkette¹ erstellt, die Grundlage für meine Überprüfung.

3. Wie wurde das Modul auf S7 realisiert?

Die Ablaufsteuerung wird in S7-Graph realisiert.

4. Wie soll/kann das Modul geprüft werden?

Für die Überprüfung der Schrittkette für die Ablaufsteuerung wird die White-Box-Test Methode verwendet. Es wird zwar dabei kein Source Code im eigentlichen Sinne überprüft, weil es sich um eine grafische Programmierungssprache handelt, jedoch werden die Schaltbedingungen sowie die Reihenfolge des einzelnen Schritte genauesten untersucht.

5. Dokumentation Modul Testbericht (jeder Modul einzeln)

In Version12011_17, ist die Schrittkette der Ablaufsteuerung weitestgehend vollständig. Aber die richtigen Einschaltbedingungen für die einzelnen Schritte sowie eine vollständige Netzwerkdokumentation sind noch nicht vorhanden.

Die Ablaufsteuerung bleibt im Rahmen des Umbaus weitestgehend unverändert. Die Schrittkette wird lediglich auf S7-Graph² umgesetzt, der Ablauf sollte zunächst unverändert bleiben. Erstmals ist es das Ziel, dass die Anlagen mit der neu realisierten Ablaufkette auf S7-Basis, die auf dem alten Ablauf basiert, funktionieren. Die Ablaufänderungen im Sinne des Optimierungsprozesses wird zunächst außen vor gelassen. Bei der Überprüfung ist es wichtig, das Ganze Schritt für Schritt anzugehen, vor allem weil es relativ schwierig war, das S5-Programm vollständig zu analysieren, aufgrund „herrenloser“ Bausteine und fehlender Netzwerkdokumentation. Bei vielen Stellen war ich auf die Hilfe unserer Mitarbeiter angewiesen, um die genaue Reihenfolge der Schritte bestimmen zu können.

¹ Schrittkette siehe Anhang_1_Lastenheft; Seite 23 „7 Allgemeine Ablaufsteuerung der VSG1

² Beispielhafte Darstellung der Ablaufsteuerung in S7-Graph siehe Anhang_08_Schrittkette_VSG1_Version_12011_17

Beispiel : Überprüfung Visualisierung der Anlage

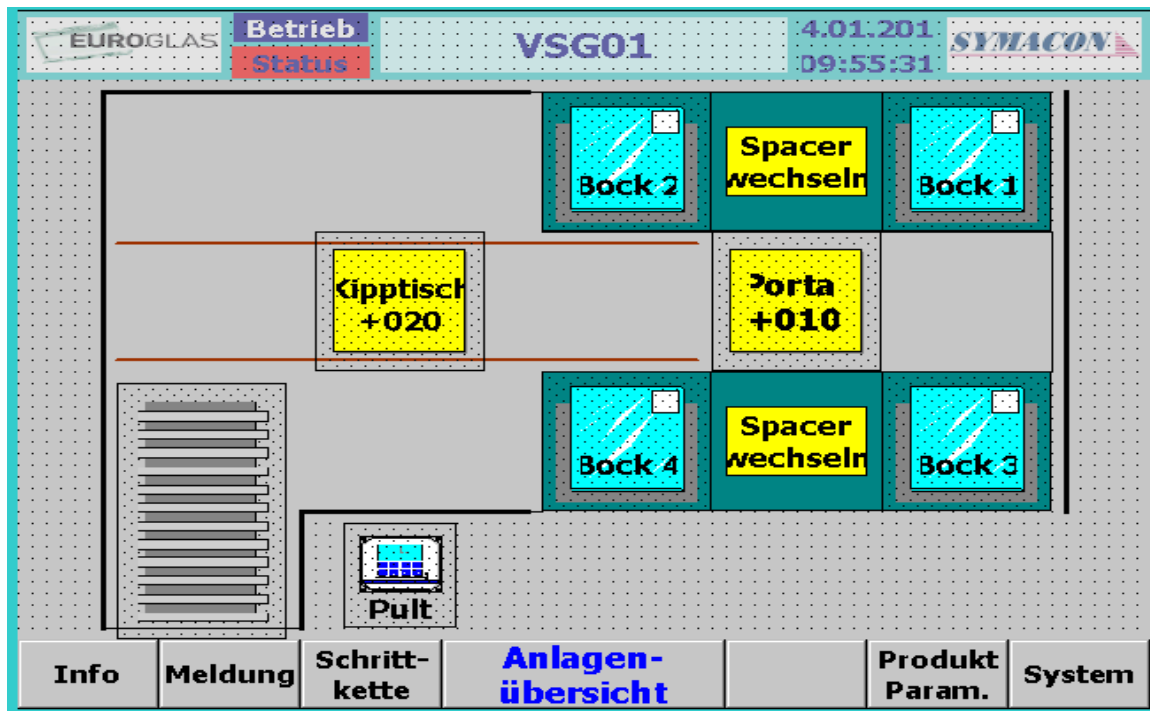


Abbildung 5-5 Beispiel WinCC Anlagenübersicht Version 12011_17

1. Wie soll/kann das S7 Programm als Ganzes geprüft werden?

Da wir kein vollständiges Programm auf einmal bekommen, habe ich mich entschieden, die Bottom-Up Methode zu nehmen. Durch die Überprüfung der einzelnen Module von unten nach oben sowie den Vergleich mit ersten Berichten, lässt sich so eine relativ genaue Aussage über den Projektfortschritt machen.

2. Welche Funktion soll das zu realisierende Modul beinhalten?

Es geht dabei um die Visualisierung der Anlagen auf einer Bedienoberfläche. Neben den üblichen Bedienelementen (Start/Stop; Automatikbetrieb/Handbetrieb) sowie eine digitale Darstellung der Anlagen, sollte es zusätzlich möglich sein, die programmierte Schrittkette auf dem Bedienpult zu beobachten.

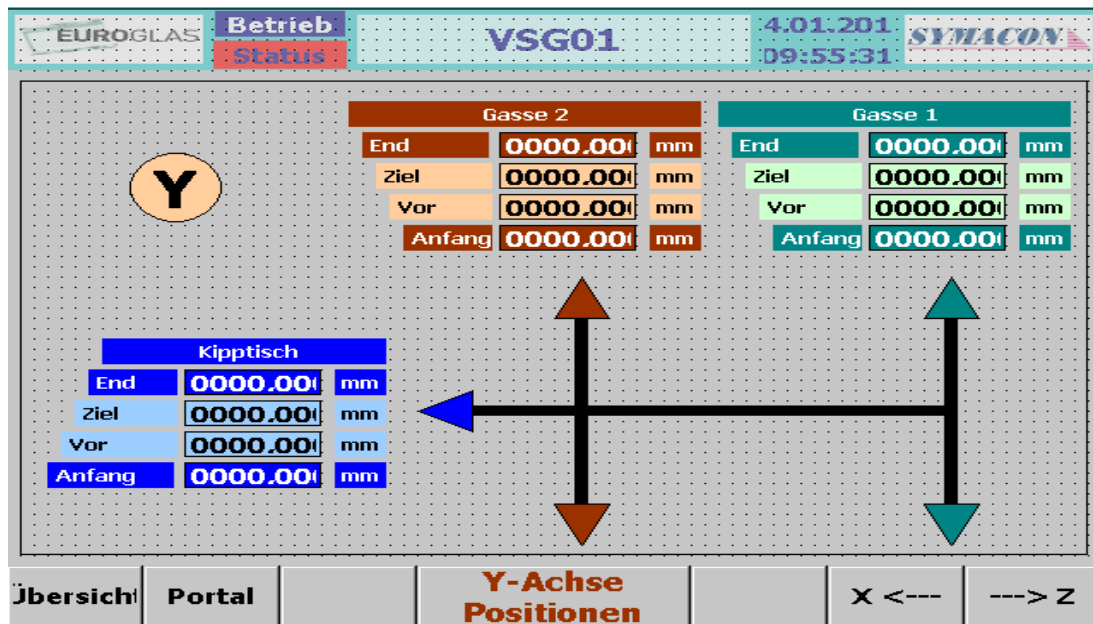


Abbildung 5-6 Beispiel WinCC Y-Achse Darstellung Version 12011_17

3. Wie wurde das Modul auf S7 realisiert?

Die Visualisierung erfolgt über WinCC Oberfläche.

4. Wie soll/kann das Modul geprüft werden?

Das Modul wird in zwei Schritten geprüft. Einerseits wird der Inhalt der einzelnen Bedienebenen überprüft, das heißt, es wird untersucht, ob alle notwendigen und gewünschten Angaben vorhanden sind. Zum Beispiel die Anzahl der Glasblöcke, die Anzahl der Glasplatten pro Gestell, die Scheibendicke etc. Andererseits wird es von einem der Schichtarbeiter hinsichtlich der Benutzerfreundlichkeit getestet. Geplant ist, einen Schichtleiter zu uns zu holen und ihn mit der neuen Oberfläche "spielen" zu lassen. Zusammengefasst bedeutet dies, dass hier der Grey-Box-Test verwendet wurde, um dieses Modul zu überprüfen.

5. Dokumentation Modul Testbericht (jeder Modul einzeln)

Die Analyse der Softwareversion 12011_17 zeigt, dass die Entwickler schon recht weit vorangekommen sind. Alle notwendigen und gewünschten Angaben sind vorhanden und die Visualisierung selbst ist bisher sehr übersichtlich. Bis auf hie und da vorhandene kleine Fehler, wie falsche Gegenstandbenennung oder falsche Maßeinheit, ist das Programm an sich fertig und nutzbar.

Zu bemerken ist, dass es sich hier eigentlich um einen Anwendertest handeln. Das bedeutet, die Benutzerfreundlichkeit und Übersichtlichkeit der Bedienebene spielt hier die primäre Rolle. Die Softwaretests hinsichtlich der Programmierungsweise spielen nur eine sekundäre Rolle.

5.8 Zusammenfassung

Aufgrund unvollständiger Netzdokumentation des S5 Programms sowie diversen „Leichenbausteinen“ war es relativ schwierig, alle benutzten Funktionen zu identifizieren. Besonders die Schrittkette des Prozessverlaufs war in manchen Schritten sehr „verwirrend“, weil sie im Laufe des Jahres immer wieder verändert und nicht vollständig dokumentiert wurde. Durch intensive Gespräche zwischen unseren Mitarbeitern und den Mitarbeitern von Symacon sowie durch die Besichtigung der Anlage im laufenden Betrieb, wurde dieses Problem, so gut es ging, minimiert.

Ein weiteres Problem stellt die Benutzung der Eingänge und Ausgänge dar. Da die VSG1 Anlagen bereits, wie oben schon erwähnt, mehrere Male sowohl in der Hardware, als auch in der Software verändert wurden, stellt sich die Frage, welcher der E/A derzeit wirklich genutzt wird bzw. nur als „Leiche“ im Programm vorhanden ist. Dazu kommt noch, dass viele E/A falsch beschriftet sind bzw. die Beschriftung im Programm stimmt mit der Beschriftung im E-Plan nicht überein. Diese Tatsache erschwert sowohl die vollständige S5 Programmanalyse, als auch den E/A Check für die Inbetriebnahme.

6 Projektauswertung

6.1 Ergebnis

Der Umbau wurde von 19.12.2012 bis 03.01.2013 durchgeführt. Von hardwaretechnischer Seite waren wir für die Inbetriebnahme bereit. Sämtliche Verkabelung wurden vollständig durchgeführt und getestet. Der E/A Test wurde fehlerfrei durchgeführt. Doch seitens der Software könnte das S7 Programm nicht rechtzeitig fertiggestellt werden. Genauer gesagt, das Entwicklerteam hat es nicht rechtzeitig geschafft, das Modul für die Steuerung der Lenze Umrichter zu schreiben.

Die Folge war, dass wir den Umbau am 02.01.2013 abbrechen und auf den alten S5 zurück bauen mussten. Der Rückbau verlief ohne große Probleme, genauso wie die Inbetriebnahme und es konnte seit 04.01.2013 wieder produziert werden. Die Umrüstung von Simatic S5 auf Simatic S7 wird auf unbekannte Zeitverschoben.

6.2 Defizite

Problem: Zeitliche Knappheit für das Personal bei der Vorbereitung.

Aufgrund von Nebentätigkeiten des zuständigen Personals für die Vorbereitung kommt es zu ständigen ungewollten Unterbrechungen bei den notwendigen Vorbereitungsschritten. Dies ist zu vermeiden, weil jede Unterbrechung die Chance auf einen Fehler sowie den Zeitaufwand für die jeweiligen Vorbereitungsschritte erhöht.

Problem: Zwangsweise Unterbrechung der laufenden Anlagen

Für einige Vorbereitungsschritte ist es notwendig, Anlagenteile für bestimmte Zeiträume anzuhalten und außer Betrieb zu nehmen. Dazu gehören z. B. die Verlegung von Leistungskabeln und Leitungen oder die Anbringung von Unterverteilungen etc. Dies ist nicht zu vermeiden, sollte so eingegrenzt und geplant werden, sodass es durch Arbeit am Wochenende abgeschlossen werden kann.

Problem: Unzureichende Erfahrung der Projektteilnehmer bezüglich Projektarbeit mit externen Firmen

In der laufenden Projektdurchführung hat sich diese mangelnde Erfahrung als fatale Folge entpuppt und es ist auch einer der Hauptgründe für das Scheitern des Projekts. Diese mangelnde Erfahrung führt zu unzureichendem Projektcontrolling. Wir haben zu sehr darauf vertraut, dass unsere Partnerfirmen uns wirklich alles sagen, sofern wirklich wichtige Probleme entstehen, die das Projekt gefährden könnten. Dadurch werden ungenaue Aussagen und Hinweise nicht vollständig wahrgenommen und nicht kontrolliert.

Problem: Kein typischer Projektverlauf

Bei einem normalen Projektverlauf wird zuerst das Lastenheft mit allen notwendigen Spezifikationen und Wünschen bezüglich der Projektziele erstellt. Das Lastenheft sollte von unseren Technikern und Projektteilnehmern angefertigt werden und ist zusätzlich dazu da, alle Teilnehmer über die Ziele und die Priorisierung des Projekts zu informieren.

Danach erfolgt die Erstellung des Pflichtenhefts von möglichen Partnerfirmen, das beschreibt, wie sie unser Interesse umsetzen wollen. Dieser Schritt ist besonders wichtig, weil es die erste Kommunikation zwischen den beiden Firmen darstellt. Ein gutes Pflichtenheft bedeutet außerdem, dass die ausgewählte Partnerfirma bereits unsere Wünsche und unser Interesse versteht und erlaubt uns erste Einblicke bezüglich ihrer Fähigkeit und Kompetenz.

Dann beginnt erst die eigentliche Projektplanung mit Terminplanung, Meilenstein-Festlegung, intensive Gespräche mit der Partnerfirma über das Projekt, um Interesse und Machbarkeit zu definieren, sowie weitere mögliche Verbesserungsmöglichkeiten.

Da wir dieser Struktur nicht gefolgt sind, kommt es immer wieder zu Gesprächen für weitere Spezifikationen und Änderungen, was zu unnötigen Verzögerungen geführt hat. Gespräche zwischen den Firmen innerhalb der Projektdauer sollten eigentlich nur dazu benutzt werden, um den Fortschritt zu betrachten und mögliche Maßnahmen dagegen zu entwickeln, sollte dieser nicht zufriedenstellend sein.

In Punkt "6.2.1 Analyse der Projektdurchführung" wird dieses Problem detailliert beschrieben.

7 Optimierungsmöglichkeiten

Wie bereits in Punkt „2.2 Vorwort Simatic“ beschreiben, handelt es sich bei der VSG1 Anlage um kein vollautomatisiertes System. Der Faktor Mensch spielt dabei eine essentielle Rolle, um die Anlage in laufendem Betrieb zu halten. Hinsichtlich des Vergleichs Nutzen und Kosten ist jedoch eine vollständige Automatisierung des VSG1 derzeit nicht profitabel. Nichtsdestotrotz sind gewisse Schritte möglich, um den Automatisierungsgrad der Anlagen zu erhöhen bzw. seine Leistung zu verbessern.

7.1 Riss -Erkennungssensor auf Rollenförderer +022

Derzeitiger Prozessverlauf:

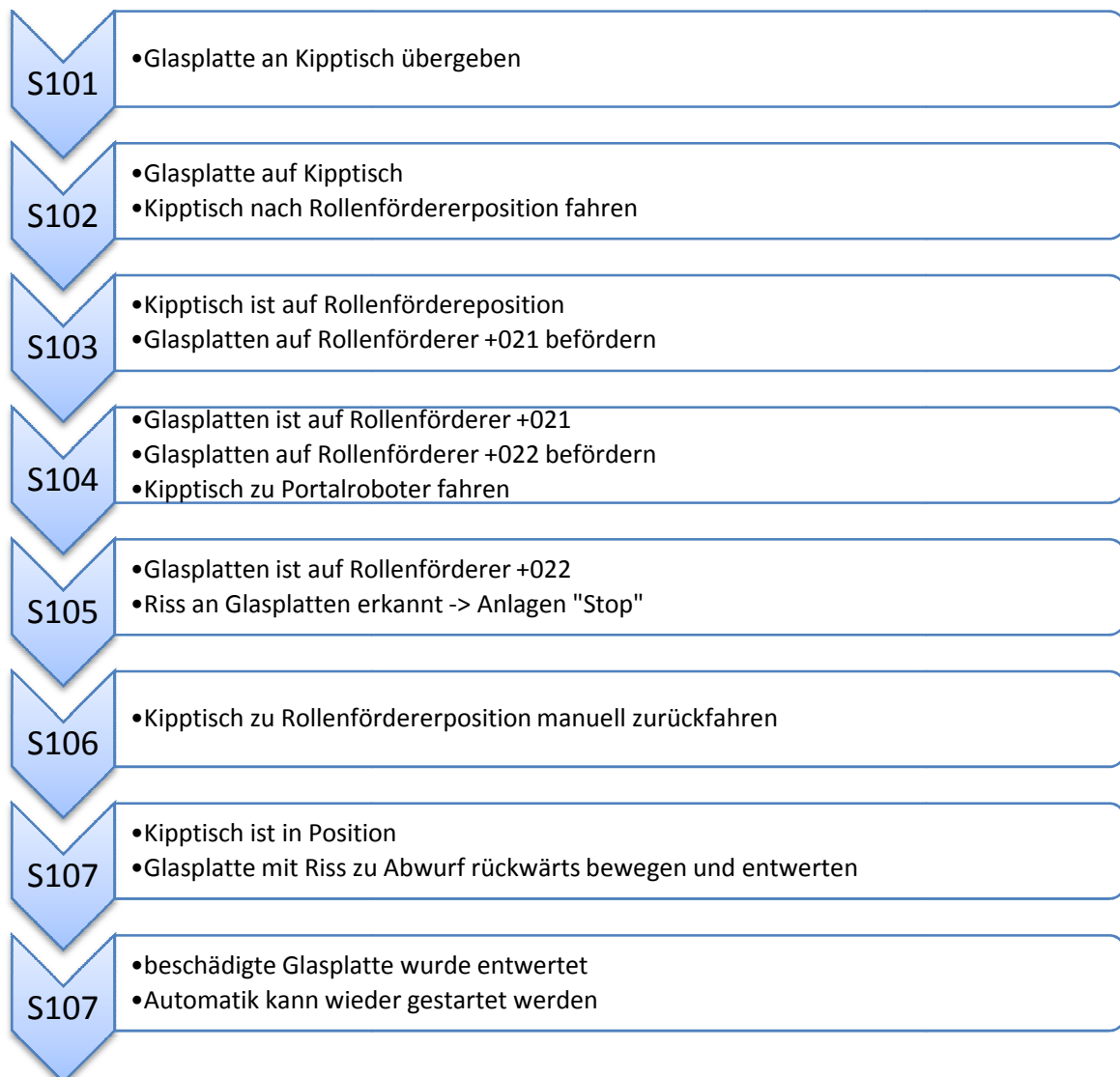


Abbildung 7-1 Schrittkette VSG1 bei Riss auf Glasplatte

Hierbei ist zu beachten, dass, wenn der Kipptisch bereits eine neue Glasplatte aufgenommen hat, diese entweder mit entwertet oder bei Hand zurück auf den Glasblock gestellt wird. Anhängig, wie es gehandhabt wird, verursacht dies einen Verlust der Arbeitszeit bzw. einen Materialverlust.

Optimierungsversuch 01: Änderung der Schrittkette sowie Automatisierung des Abwurfvorgangs

Beim Optimierungsversuch 01 wird lediglich eine Verriegelung hinzugefügt. Diese Verriegelung besagt, dass die Glasplatte nur an den Kipptisch übergeben werden kann, wenn der Rollenförderer +022 frei ist oder er ist besetzt, aber es wurde kein Riss erkannt. Die Schrittkette wurde folglich so aussehen:

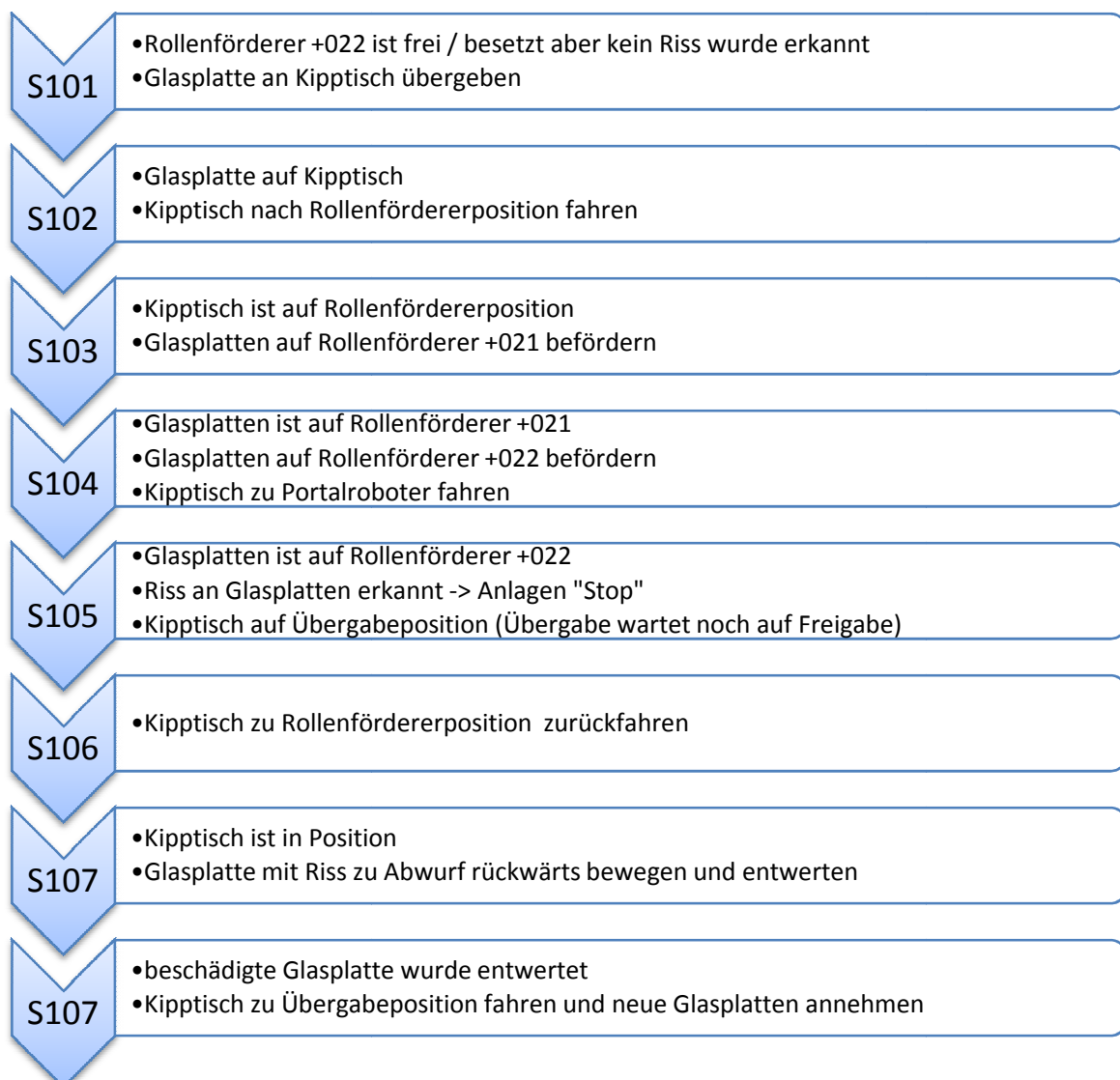


Abbildung 7-2 Schrittkette VSG1 Optimierungsversuch 01

Optimierungsversuch 02: Verlegung des Rissensors von Rollenförderer +022 auf Rollenförderer +021 sowie Automatisierung des Abwurfvorgangs.

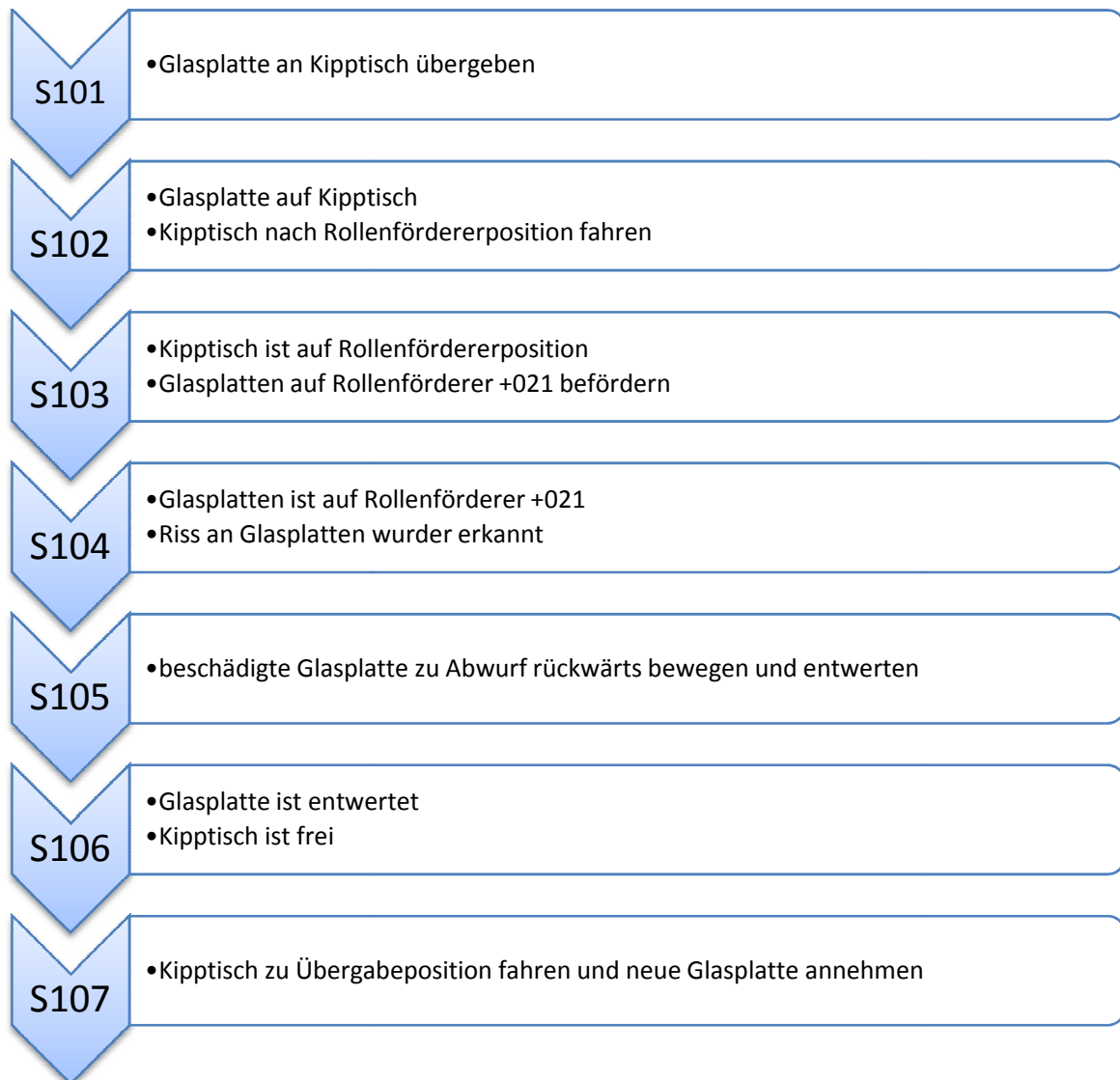


Abbildung 7-3 Schrittkette VSG1 Optimierungsversuch 02

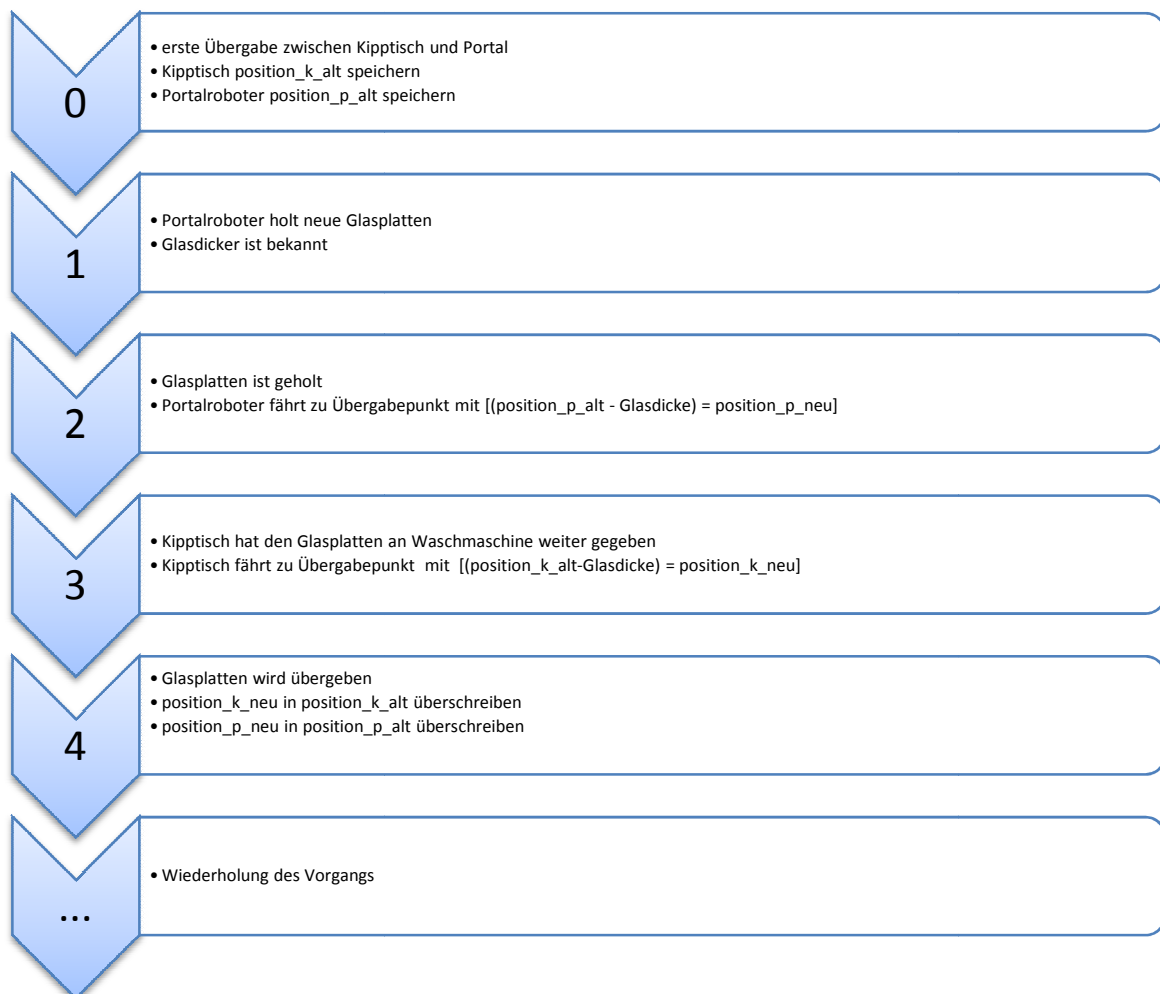
	Vorteil	Nachteil
Optimierungsversuch 01	<ul style="list-style-type: none"> ➤ Geringer Materialverlust ➤ Kein Umbau im eigentlichen Sinne ➤ Höherer Automatisierungsgrad 	<ul style="list-style-type: none"> ➤ Gleicher bzw. höherer Zeitverlust durch Hin- und Herfahren des Kipptisches und warten auf die Freigabe zur Glasübergabe
Optimierungsversuch 02	<ul style="list-style-type: none"> ➤ Bessere Zeitausnutzung ➤ Geringer Materialverlust ➤ Höherer Automatisierungsgrad 	<ul style="list-style-type: none"> ➤ Umbau der Transportstrecke
<p>Hinweis. Hinsichtlich der Automatisierung der Entwertung der beschädigten Glasplatte muss die Länge der Abfangbehälter auf das „Bandmaß“ der Glasplatte angepasst werden, weil kein Mitarbeiter mehr da steht, der die Glasplatte kleiner schneidet. Mitarbeiter werden nur noch benötigt, um Spacer und Gestell zu wechseln.</p>		

Tabelle 9 Vorteil und Nachteil der Optimierungsversuche

7.2 Intelligente Positionierung der Portalroboter und Kipptisch beim Übergabe

Derzeit ist es so, dass der Kipptisch und Portalroboter jedes Mal eine feste Position für die Übergabe von Glasplattenanfährt. Je nach der übrig gebliebenen Anzahl der Glasplatten auf dem Glasbock wird wertvolle Produktionszeit vergeudet für das Befahren der Transportstrecke. Die Idee dabei ist, dass die Ablaufsteuerung geändert werden sollte, sodass der Kipptisch und Portalroboter, je nach Anzahl der Glasplatte auf dem aktiven Glasbock, ihre Position ändern.

Prinzipiell würde die Schrittkette so ablaufen:



Ungefähr nach diesen Schritten sollte die neue Schrittkette dann ablaufen. Dadurch wird das Befahren von unnötiger Strecke verhindert und wertvolle Produktionszeit gewonnen. Die Änderung der Kette an sich ist relativ leicht in S7- Graph umzusetzen. Das Problem liegt an dem richtig konfigurierten Abstand zwischen Portalroboter und Glasbock sowie zwischen Portalroboter und Kipptisch, der so klein wie möglich sein sollte, um dennoch einen reibungsfreien Ablauf garantieren zu können. Dabei ist zu erwähnen, dass der Portalroboter schon einige Jahre hinter sich hat und nicht mehr so präzise wie gewünscht funktioniert. Das ist ein wichtiger Aspekt, der bei der Parametrierung der Abstände bzw. bei der Bestimmung der Toleranz- und Abweichungsbereiche zu beachten ist.

7.3 Vollständige Automatisierung

Für eine vollständige Automatisierung muss das Problem, wie automatische Spacer Entfernung, Riss-Erkennung der ganzen Glasblock, gelöst werden. Dieses Problem ist lösbar, aber hinsichtlich die entstehenden Kosten und zeitliche Investition, ist es derzeit noch nicht wirklich profitabel. Darum werde ich die notwendige Aufgabe und den dazugehörigen Lösungsvorschlag nennen, aber nicht weiter näher darauf eingehen.

Die **automatische Spacer Entfernung** kann durch einen Greifarmroboter automatisch entfernt werden. Durch einen angebrachten Näherungssensor kann der Roboter, beim Befehl für das Entfernen von Spacer, einfach an der oberen Kante des Glasblocks entlang fahren und die erkannten Spacer entnehmen.

Es gibt verschiedene Methoden zur **Riss-Überprüfung ganzer Glasblöcke**. Dabei ist zu beachten, dass dieser Vorgang nicht unbedingt in den VSG1 Prozessablauf eingebunden werden muss, sondern auch bereits davor durchgeführt werden kann. Zur Methode der Riss-Überprüfung gehören das Thermobild von thermischen Verfahren, das Ultraschallverfahren bzw. Durchschallungsverfahren, aber auch das optische Verfahren sowie das Durchleuchtungsverfahren. Je nach Applikation und erwarteter Prüfqualität können die Verfahren einiger Glasplatten bis hin zu einem ganzen Glasblock gleichzeitig prüfen.

Der automatische Wechsel von Gestellen ist nur im Rahmen einer vollständigen Automatisierung der gesamten Produktionsstätte von Wareneingang bis hin zu Warenausgang sowie Lagerhaltung möglich. Es ist möglich, die Gestelle mit „aktivem“ Datenträger auszustatten und so zu programmieren, sodass mithilfe von RFID Verfahren die notwendige Information aufgerufen werden kann. Durch die enthaltene Information kann man den Inhalt auf das Gestell bestimmen und je nach Auftrag mit Transportbändern zu seinem Bestimmungsort bringen und abarbeiten. *Glasblock Position siehe 3.1.2 oder 3.1.3 (Übersicht der Anlage)*

Es ist jedoch fragwürdig, selbst bei hohem Automatisierungsgrad der Produktionsstätte, ob es sich wirklich lohnt, diesen Schritt durchzuführen. Es ist jedenfalls machbar, ein Beispiel dafür wäre das BMW Werk in Leipzig mit seinem M-förmigen Transportband, wo die einzelnen Schritte der Automontage durchgeführt werden.

Aufgrund betrieblicher Geheimhaltung war es mir nicht möglich Bilder der BMW Produktionsstätte zu bekommen. Auch hier wird ein Mitarbeiter für den Prozessablauf benötigt, also kein vollautomatisiertes System

als Ganzes, ich wurde dennoch jedem Student und Neueinsteiger empfehlen, an einer Werkführung teilzunehmen. Es bietet einen guten Einblick für die Automationsmöglichkeit in diesen Branchen.

7.4 Mit „Blick in die Zukunft“ – vollständige Vernetzung der Anlage

7.4.1 Allgemein Busstruktur/Bustopologie

Die Topologie bezeichnet bei einem Computernetz die Struktur der Verbindungen mehrerer Geräte untereinander, um einen gemeinsamen Datenaustausch zu gewährleisten.

Die Topologie eines Netzes ist entscheidend für seine Ausfallsicherheit: Nur wenn alternative Wege zwischen den Knoten existieren, bleibt bei Ausfällen einzelner Verbindungen die Funktionsfähigkeit erhalten. Dann gibt es neben dem Arbeitsweg einen oder mehrere Ersatzwege (oder auch Umleitungen).

Die Kenntnis der Topologie eines Netzes ist außerdem nützlich zur Bewertung seiner Performance sowie der Investitionen und für die Auswahl geeigneter Hardware.

Es wird zwischen physikalischer und logischer Topologie unterschieden. Die physikalische Topologie beschreibt den Aufbau der Netzverkabelung; die logische Topologie den Datenfluss zwischen den Endgeräten.

Topologie wird grafisch mit Knoten und Kanten dargestellt. In großen Netzen findet man oftmals eine Struktur, die sich aus mehreren verschiedenen Topologien zusammensetzt.

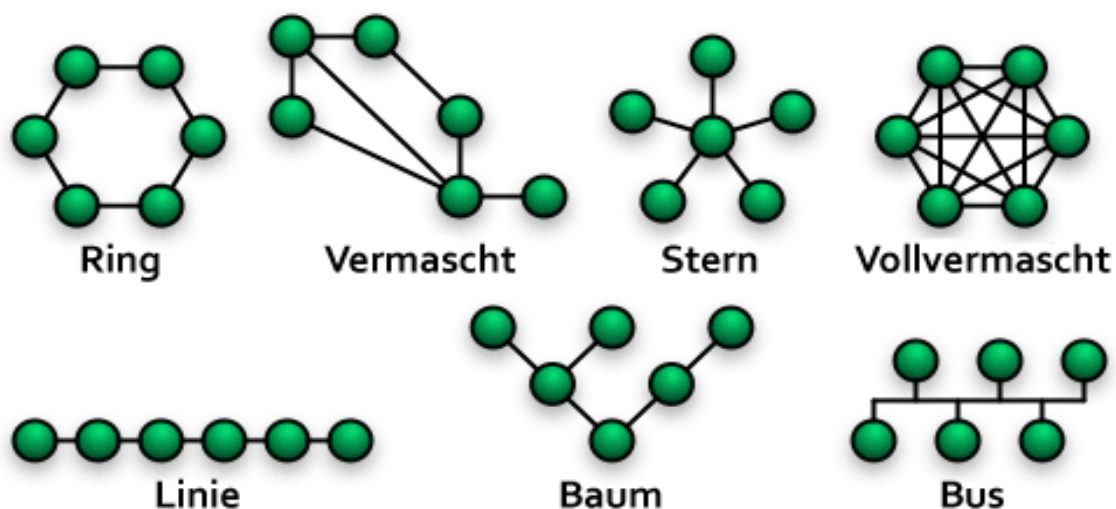


Abbildung 7-4 Netzwerktopologie

7.4.2 Anlagenbusstruktur

Bei der VSG1 Anlage wird derzeit das Linien- oder Strangverfahren benutzt, das heißt, alle Geräte sind direkt mit demselben Übertragungsmedium verbunden. Linientopologie bedeutet, dass es keine aktiven Komponenten zwischen den Geräten und dem Medium gibt. Dabei ist das Übertragungsmedium bei Systemen mit einer kleineren physikalischen Ausdehnung oft direkt auf einer Leiterplatte realisiert und sonst als Kabel oder Kabelbündel. Um eine mögliche Kollision zu vermeiden, ist bei dieser Bus-Topologie sicherzustellen, dass immer nur ein Gerät zum selben Zeitpunkt Signale sendet. Dies kann durch eine zentrale Einheit, den sogenannten Bus-Arbeiter geregelt werden. Sprich, um senden zu dürfen, muss der Teilnehmer über eine separate Leitung eine entsprechende Anfrage an den Bus-Arbeiter stellen. Eine zentrale Regelung ist aber gerade bei dynamischen Netzwerken, wie Computernetzwerken, oft nicht praktikabel und ineffizient. Daher werden bei diesen Netzwerken gleichzeitige Zugriffe erkannt und die entstehenden Probleme aufgelöst. Ein oft benutztes Verfahren ist z. B. CSMA/CD (Carrier Sense Multiple Access / Collision Detection).

Vorteile:

- Der Ausfall eines Gerätes hat für die Funktionalität des Netzwerkes keine Konsequenzen
- Nur geringe Kosten, weil nur geringe Kabelmengen erforderlich sind
- Einfache Verkabelung und Netzerweiterung
- Es werden keine aktiven Netzwerkkomponenten benötigt

Nachteile:

- Eine Störung des Übertragungsmediums an einer einzigen Stelle im Bus (defektes Kabel) blockiert den gesamten Netzstrang
- Es kann zu jedem Zeitpunkt immer nur eine Station Daten senden. Währenddessen sind alle anderen Sender blockiert (interne Datenzwischenpuffer)
- Bei Bussen, die Kollisionen zulassen und auf eine nachträgliche Behebung setzen, kann das Medium nur zu einem kleinen Teil ausgelastet werden, weil bei höherem Datenverkehr überproportional viele Kollisionen auftreten

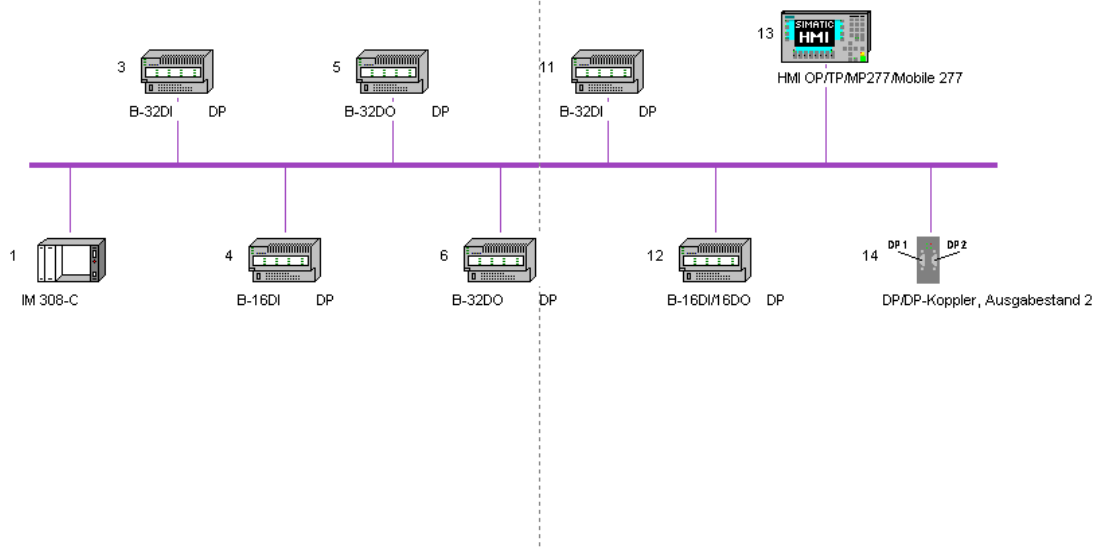


Abbildung 7-5 derzeitige Busstruktur

Hinsichtlich der Kosten-Nutzen Frage, ist die Verwendung der Linientopologie derzeit vollkommen ausreichend und deckt alle nötigen Spezifikationen für eine störungsfreie Produktion.

Wie bereits mehrfach genannt, handelt es sich bei der Modernisierung des Steuerungssystems der Teilanlage VSG1 um einen Testlauf und sollte als Modell für eine später folgende Modernisierung dienen. Das heißt, dass schon feststeht, dass die gesamten Steuerungselemente der Glasveredelungsanlage mit der Zeit modernisiert werden sollen. Unter diesen Aspekt wäre es ratsam, bereits zu diesem Zeitpunkt ein Konzept für die vollständige Vernetzung der Anlage zu entwickeln.

Es gibt mehrere Vorteile, die durch eine Vernetzung der Anlage zustande kommen können:

- Verbessertes Qualitätsmanagement
- Verbesserte Fernwartungsmöglichkeit
- Einführung einer zentralen „Warte“ (zentralen Überwachung)
- Verbesserte Datenverfolgungsmöglichkeit
- Erhöhung des Automatisierungsgrads sowie verbesserte Datenkommunikation mit anderen Unternehmensebenen
- Vereinfachte Datenarchivierung
- Zentrale Verwaltung von Rezepten für die einzelnen Prozessabläufe
- Vereinfachung für die Einführung eines auftragsbasierten Produktionsverlaufs

Selbst ohne die obengenannten Vorteile ist eine Vernetzung der einzelnen Teilanlagen miteinander in ein automatisiertes System prinzipiell sinnvoll. Es sind verschiedene Schritte nötig, um so ein Konzept zu entwickeln. Im Prinzip entscheidet man sich zunächst für eine der Übertragungsarten, die man verwenden möchte. Unter Übertragungsarten versteht man entweder kabelgebundene, mit Profibus oder Ethernet, oder aber mit kabelloser Datenübertragung, wie WLAN. Je nach verwendeten Übertragungsarten gibt es besondere Schwerpunkte, die zu berücksichtigen sind (z. B. Übertragungstrecke, Übertragungsgeschwindigkeit, Zugriffsverfahren etc.).

In Anhängigkeit der benutzten Übertragungsarten wird dann eine passende Netzwerktopologie dazu ausgesucht. Es versteht sich von selbst, dass es z. B. keinen Sinn macht, eine Bustopologie für WLAN zu entwerfen bzw. ist eine „Vollvermaschte“ Busstruktur über Profibus oder Ethernet allein aus Kostengründen im Verhältnis zu ihrem Nutzen nicht wirklich optimal. Wie oben bereits gesagt, reicht die derzeitige Linienstruktur für einen störungsfreien Betrieb aus. Jedoch kann es nicht alle notwendigen Voraussetzungen hinsichtlich einer vollständigen Vernetzung aller Teilanlagen erfüllen. Da wäre eine Ringstruktur oder Sternstruktur bzw. eine Mischung aus beiden Strukturen mehr angebracht, um die erhöhten Zuverlässigkeits- und Sicherheitsansprüche abdecken zu können.¹

¹ Für möglichen Topologie siehe_Anhang_03_Netz_V2

8 Literaturverzeichnis

- Folien S5 Migration von Ralf Stegmann Siemens AG
- Vgl. DeMarco, Tom/Lister, Timothy (2003): Bärenango – Mit Risikomanagement Projekte zum Erfolg führen, Hanser;
- Andreas Spillner, Tilo Linz: Basiswissen Softwaretest. Aus- und Weiterbildung zum Certified Tester - Foundation Level nach ISTQB-Standard. 4. Auflage. dpunkt.verlag, Heidelberg 2010
- Organisation Theorie und Praxis aus ökonomischer Sicht. Gebunden Schäffer-Poeschel ISBN 978-3-7910-3137-8
- IEEE-SA Standards Board: IEEE Standard for Software Test Documentation. Piscataway NJ, 1998: The Institute of Electrical and Electronic Engineers Inc. ISBN 0-7381-1443-X
- Eric J. Braude: Software Engineering. An Object-Oriented Perspective. Hoboken NJ, 2000: Wiley Inc. ISBN 0-471-32208-3
- Projektmanagement Fallstudie von Harold Kerzner, ISBN 3-8266-1447-X
- Professionelle Softwaretests von Marcel Alper, ISBN 3-528-05454-9
- IT-Projekte strukturiert realisieren von Ralp Brugger, vieweg Verlag, ISBN 3-528-05847-1

9 Internetverzeichnis

- <http://www.verwaltung-innovativ.de/>
 - Suchbegriff: Organisation/Prozessmanagement
- <http://de.wikipedia.org>
 - Suchbegriff: Projektmanagement; SW Testverfahren
- <http://www.beuth.de/de/norm/din-69901-1/>
- <http://wirtschaftslexikon.gabler.de/Archiv/16275/top-down-prinzip-v6.html>
- <http://www.wirtschaftslexikon24.net/d/planungsarten/planungsarten.htm>
- <http://softwaretestingstandard.org/>
- <http://www.euroglas.com/unternehmen/uebersicht-unternehmen-euroglas.html>

10 Anhang

Die Anhänge befindet sich auf den beigelegte CD.

Anhang 01: Lastenheft

Anhang 02: Inbetriebnahmedokument

Anhang 03: Netz V2

Anhang 04: Anlagenbeschreibung

Anhang 05: Achsenbeschreibung X-Achse

Anhang 06: Sollzeitplan VSG1

Anhang 07: Euroglas Imagebroschüre

Anhang 08: Schrittkette VSG1 Version 12011_17